

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

**NOVEL TECHNIQUES FOR SPARSE REPRESENTATION
PROBLEMS**

CHAI WOON HUEI

**Interdisciplinary Graduate School
Energy Research Institute @ NTU**

2021

NOVEL TECHNIQUES FOR SPARSE REPRESENTATION PROBLEMS

CHAI WOON HUEI

INTERDISCIPLINARY GRADUATE SCHOOL

A Report submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Doctor of Philosophy

2021

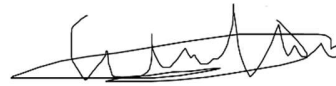
Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

11/02/2021

.....

Date



.....

Chai Woon Huei

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accordance with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

11/02/2021

.....
Date



.....
Assoc Prof Quek Hiok Chai

Authorship Attribution Statement

This thesis contains material from papers published in the following peer-reviewed conferences and publication drafts under preparation for the publications in the following peer-reviewed journals where I was the first and corresponding author.

Chapter 3 is prepared as W.H. Chai, and C. Quek. A novel Quasi-Newton Technique for Composite Convex Minimization. *Pattern Recognition* (Under revision)

The contributions of the co-authors are as follows:

- I defined the initial project direction
- Assoc Prof Quek edited the manuscript drafts.
- The manuscript was revised by Assoc Prof Quek.
- I generated the idea, derived the theory, designed the study and the algorithm, performed all the experimental works, and analyzed the results at the School of Computer Science and Rolls-Royce@NTU Corporate Lab.

Chapter 4 is prepared as W.H. Chai, and C. Quek. Representation Recovery via L_1 -norm Minimization with Corrupted Data. *Information Sciences* (Submitted)

The contributions of the co-authors are as follows:

- Assoc Prof Quek provided the initial project direction and edited the manuscript drafts.
- The manuscript was revised by Assoc Prof Quek.
- I generated the idea, derived the theory, designed the study and the algorithm, performed all the experimental works, and analyzed the results at the School of Computer Science and Rolls-Royce@NTU Corporate Lab.

Chapter 5 is prepared as W.H. Chai, and C. Quek. Feature Selection for Dictionary Transfer. *Pattern Recognition Letters* (Under preparation for publication)

The contributions of the co-authors are as follows:

- I defined the initial project direction
- Assoc Prof Quek edited the manuscript drafts.
- The manuscript was revised by Assoc Prof Quek.
- I generated the idea, derived the theory, designed the study and the algorithm, performed all the experimental works, and analyzed the results at the School of Computer Science and Rolls-Royce@NTU Corporate Lab.

Chapter 6 is published as:

- W. H. Chai, S.-S. Ho, and C.-K. Goh, “Exploiting sparsity for image-based object surface anomaly detection,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 1986–1990.

The contributions of the co-authors are as follows:

- I defined the initial project direction
 - Asst Prof Ho edited the manuscript drafts.
 - The manuscript was revised by Asst Prof Ho and Dr Goh.
 - I generated the idea, designed the study and the algorithm, performed all the experimental works, and analyzed the results at the School of Computer Science and Rolls-Royce@NTU Corporate Lab.
- W. H. Chai, S.-S. Ho, C.-K. Goh, L.-T. Chia, and H. C. Quek, “A fast sparse reconstruction approach for high-resolution image-based object surface anomaly detection,” in *2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA)*, 2017, pp. 13–16.

The contributions of the co-authors are as follows:

- I defined the initial project direction
- Asst Prof Ho edited the manuscript drafts.
- The manuscript was revised by Asst Prof Ho, Dr Goh Assoc Prof Chia, and Assoc Prof Quek.

- I generated the idea, designed the study and the algorithm, performed all the experimental works, and analyzed the results at the School of Computer Science and Rolls-Royce@NTU Corporate Lab.

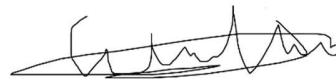
Appendix F is prepared as W.H. Chai, and C. Quek. A Novel Ensemble Method. *Pattern Recognition Letters* (Under preparation for publication)

The contributions of the co-authors are as follows:

- I defined the initial project direction
- Assoc Prof Quek edited the manuscript drafts.
- The manuscript was revised by Assoc Prof Quek.
- I generated the idea, designed the study and the algorithm, performed all the experimental works, and analyzed the results at the School of Computer Science and Rolls-Royce@NTU Corporate Lab.

11/02/2021

.....
Date



.....
Chai Woon Huei

Abstract

Sparse representations have been used in solving many problems in computer science. Two issues that need to be addressed in formulating such a representation are:

- the problem design; and
- the optimization technique.

Many optimization problems contain one/multiple non-smooth terms in the objective function. Besides, the feasibility of an optimization problem depends on the availability of adequate computational resources. In this thesis, a new parallelizable optimization technique that uses more information and has better convergence than state-of-the-art counterparts is presented. Theoretical derivation of the bound of the recovery probability of using sparse representation based on a L_1 -minimization is also shown. A clustering-based technique for dictionary and signal dimension reduction to replace the traditional naïve downsampling technique is introduced to address computational resource constraints. Finally, an anomaly detection and localization technique using a sparse representation problem and used in a case study for an important and challenging field; namely automated visual inspection (AVI) is presented. The experimental results are encouraging.

Lay Summary

Sparse representations have been used in solving many problems in computer science. Two issues need to be addressed in formulating such a representation are:

- the problem design, and
- the optimization technique.

Many optimization problems contain one/multiple non-smooth terms in the objective function. Besides, the feasibility of an optimization problem depends on the availability of adequate computational resources. In this thesis, two contributions in problem designs, two contributions in optimization techniques, and an application of sparse representation are presented.

The two contributions in problem design are:

- theoretical derivation of the bound of the recovery probability of using sparse representation based on a L_1 -minimization, and
- sparse representation-based anomaly detection and localization,

The two contributions in optimization technique are:

- a parallelizable optimization technique which uses more information and has better convergence than the state-of-the-art counterparts, and
- a clustering-based technique for dictionary and signal dimension reduction to replace the traditional naïve downsampling technique (to make the problems feasible under computational resources constraints),

An application case study, surface defect detection, and localization, using the proposed sparse representation-based anomaly detection and localization technique is investigated. It is an important and challenging in the field; namely: automated visual inspection (AVI) in industry. The experimental results are encouraging.

Acknowledgements

First of all, I would like to thank my family for their unwavering support and encouragement. Their support has been invaluable. Equally, I would like to take this opportunity to express my utmost gratitude to my supervisor Assoc Prof Quek Hiok Chai, who has supported me endlessly with his guidance, patience, and time during the past 3 years of my PhD candidature. His expertise and knowledge have helped me immensely. I would also like to extend my gratitude to my previous supervisor Asst Ho Shen-Shyang. He was responsible for developing the foundation of my research skills and supported me to become a qualified PhD candidate in computer science during my first 2.5 years. Moreover, I would like to thank my co-supervisors and mentor, Dr Goh Chi Keong, Dr Albert Phang, Assoc Prof Chia Liang Tien, Asst Prof Fan Zheng, and Dr Terence Hung. Without their supervision, advice, and guidance, I would never have been able to reach my full potential as a better researcher, one who is highly disciplined in conducting high-quality research.

Furthermore, I would like to thank the Rolls-Royce@NTU Corporate Lab with, support from the National Research Foundation (NRF) Singapore under the Corp Lab@University Scheme, Computational Intelligence Lab (CIL), and Parallel & Distributed Computing Laboratory (PDCL). A special thank you to the lab technicians, Kesavan Asaithambi and Irene Goh. They provided the facilities and technical support for my research. I would also like to thank my colleagues, Dr Hung Tzu-Yi, Dr Mohd Tanveer, Low Cheng Seng, Patrick Chen Pei-Hung, That Mon Htwe, Natarajan Vidhya, Vaikundam Sriram, Peng Haiyun whom I have had a lot of fruitful discussions with. Also, a big thank you to my friends, Ma Yukun, Gao Sa Wei Pengfei, Song Wen, Hu Zehong, and Pan Siyi for their encouragement and fun moments. These moments certainly helped me relieve some stress during those difficult days.

Lastly, I would like to thank the School of Computer Science and Engineering (SCSE) and Energy Research Institute@NTU (ERIAN) under Interdisciplinary Graduate School (IGS)

Acknowledgements

at Nanyang Technological University (NTU) and the ERIAN executive Huang Minying. I am really grateful for the various support rendered to me from executive Huang Minying. I am eternally grateful for the precious opportunities, and the scholarship provided by the respective schools during my Doctor of Philosophy journey.

Table of Contents

Abstract	xii
Lay Summary	xiv
Acknowledgements	xvi
Table Captions.....	xxvi
Figure Captions	xxix
Abbreviations	xxxiii
Notations	xxxix
Chapter 1 Introduction	1
1.1. Research Statement.....	2
1.2. Problem design.....	3
1.3. Optimization Technique.....	5
1.4. Organization of Thesis.....	7
Chapter 2 Literature Review	10
2.1. Sparse Representation.....	10
2.1.1. Originality	11
2.1.2. Regularizations	11
2.1.3. $L1$ - $L1$ Regularization.....	16
2.1.4. Theory Contributions.....	18
2.1.5. Curse of Dimensionality	26
2.2. Automated Visual Inspection (AVI) and Its Techniques.....	27
2.2.1. Surface Defects	28
2.2.2. AVI Techniques	29
2.2.3. Challenges and Issues	33
2.3. Summary	34
Chapter 3 A Novel Quasi-Newton Technique for Composite Convex Minimization.....	36
3.1. Integrated Techniques and the Benefits.....	37
3.2. Proximal Average Function and Its Proximal Mapping and Gradient	39
3.3. Quasi-Newton Technique with Proximal Mappings.....	42

3.4.	Proposed Method and Algorithms with Combination of Gradient Descent and Quasi-Newton Techniques	43
3.5.	Convergence Analysis	46
3.5.1.	Quasi-Newton Method.....	53
3.5.2.	Steepest Gradient Descent Method.....	55
3.5.3.	Nesterov's Accelerated Gradient Descent Method.....	56
3.5.4.	Discussion of Convergence Complexity of the Proposed Method and several States of the Art Methods.....	58
3.6.	Experiments	60
3.6.1.	Problem of Sum of Multiple Sparsities.....	61
3.6.2.	Experiments with $\alpha_1 = 0.5$ and $\alpha_2 = 0.5$	66
3.6.3.	Experiments with $\alpha_1 = \frac{1}{1+10^5}$ and $\alpha_2 = \frac{10^5}{1+10^5}$	72
3.7.	Summary	77
Chapter 4	Representation Recovery via $L1$ -norm Minimization with Corrupted Data	79
4.1.	Notations	82
4.2.	Multi-bouquet Models and Main Assumptions	84
4.3.	The Main Result.....	87
4.4.	Roadmap of the Proof.....	91
4.5.	Optimality Conditions.....	92
4.6.	Iterative Construction of Separator	94
4.7.	Probabilistic Analysis and Final Proof	95
4.7.1.	Projection of Sparse Vectors.....	96
4.7.2.	Initial Certificate	98
4.8.	Proof of Theorem 1	99
4.9.	Experiments	107
4.9.1.	Simulations using Artificial Data fulfilling the Bouquet Model	111
4.9.2.	Simulations using Artificial Data fulfilling the Multi-Bouquet Model	116
4.9.3.	Simulations using Practical Data used in Face Recognition.....	120
4.9.4.	Simulations using Practical Data used in Viewpoint Classification.....	122
4.10.	Summary	124
Chapter 5	Feature Selection for Dictionary Transfer	126

5.1.	Related Works.....	128
5.2.	Motivation.....	130
5.3.	Main Notations and Assumptions.....	130
5.4.	Feature Selection for Dictionary Transfer	131
5.4.1.	Feature Grouping with Unsupervised Clustering	133
5.4.2.	Feature Selection with Nearest Neighbor Approach	134
5.4.3.	Numbers of Cluster and Nearest Neighbors to be selected.....	134
5.5.	Experiments	135
5.5.1.	Simulations using Artificial Data fulfilling the Multi Bouquet Model.....	139
5.5.2.	Simulations using Practical Data used in Face Recognition.....	141
5.5.3.	Simulations using Practical Data used in Viewpoint Classification.....	145
5.6.	Summary	147
Chapter 6	Anomaly Detection and Localization.....	149
6.1.	Sparse Reconstruction with Noise	151
6.2.	Sparse Reconstruction Approach for Face Recognition.....	152
6.3.	Proposed Anomaly Detection and Localization Approach.....	153
6.4.	Sparse Reconstruction Approach (SRA) for Defect Detection and Localization	155
6.4.1.	Sparse Reconstruction.....	155
6.4.2.	Defect Detection and Localization.....	157
6.4.3.	Finding Best λ , η and T	158
6.4.4.	Parameter Analysis	159
6.5.	Extended SRA for High-Resolution Images.....	165
6.5.1.	Fast Sparse Reconstruction.....	166
6.5.2.	Defect Detection and Localization.....	168
6.5.3.	Finding Best λ and η	170
6.6.	Illumination Reconstruction for Lambertian Surfaces.....	171
6.7.	Computation Time and Memory Demand Analysis.....	174
6.7.1.	Computation Time Analysis	174
6.7.2.	Memory Demand Analysis	175
6.8.	Experiments	176

6.8.1. Experimental Results for Original SRA	176
6.8.2. Experimental Results for Extended SRA.....	183
6.8.3. Performance Comparisons of Original and Extended SRAs	192
6.9. Summary	194
Chapter 7 Conclusions and Future Works	195
7.1. Conclusions.....	195
7.2. Future Works	197
7.2.1. Composite Convex Minimization Technique	197
7.2.2. Representation recovery via $L1$ -norm minimization	197
7.2.3. Feature Selection Technique for Dictionary Transfer	197
7.2.4. Anomaly detection and localization.....	198
Appendix A (Appendix for Chapter 2)	199
A.1. Characteristics of Surface Defects	199
A.1.1. Fabric Surface Defects.....	199
A.1.2. Random Texture Surface Defects	199
A.1.3. Electronic Device Surface Defects	200
A.1.4. Complex Surface Defects	200
A.2. Details of published AVI techniques	201
A.2.1. Techniques for Uniform Surface	218
A.2.2. Techniques for Uniform Random Texture Surface	225
A.2.3. Techniques for Patterned Textured Surface.....	228
A.2.4. Techniques for Non-uniform and Non-complex Surface	231
A.2.5. Techniques for Complex Surface.....	232
A.2.6. General Approaches and Supporting Techniques.....	235
Appendix B (Appendix for Chapter 3).....	237
Appendix C (Appendix for Chapter 4).....	245
C.1. Tools for Probability Analyses	245
C.1.1. Preliminaries	245
C.1.2. Observations on Submatrix Singular Values	246
C.1.3. Two Concentration Results.....	247
C.1.4. Concentration of Measure.....	249

C.1.5. Concentration for Functions of i.i.d. Rademacher Sequences	256
C.2. Technical Lemmas	257
C.3. Proofs of Lemmas	257
C.3.1. Proof of Lemma 1	257
C.3.2. Proof of Lemma 3	260
A. Lower Bounding $1 - \xi$	260
C.3.3. Proof of Lemma 4	265
C.3.4. Proof of Lemma 5	274
C.3.5. Proof of Lemma 9	275
C.3.6. Proof of Lemma 10	277
C.3.7. Proof of Lemma 12	279
C.3.8. Proof of Lemma 13	279
Appendix D (Appendix for Chapter 5)	281
Appendix E (Appendix for Chapter 6).....	285
E.1. Algorithm for Finding Best λ , η and T for SRA.....	285
E.2. Algorithms for Finding Best λ and η for Extended SRA	286
E.2.1. Main Algorithm	288
E.2.2. Algorithm for Optimizing a Parameter	290
E.2.3. Algorithm for Determining Detection Accuracy	290
Appendix F A Novel Ensemble Method.....	293
F.1. Related Works.....	295
F.1.1. Boosting	295
F.1.2. Bagging.....	295
F.1.3. Combination Methods.....	296
F.1.3.1. Stacking and Meta-learning	296
F.2. Notations	296
F.3. Methodology	299
F.3.1. Base Classifier (first-level learner) Training	299
F.3.2. Estimation of Relevant Metric with Stacked Regression-based Meta-learning	301
F.3.3. Test Set Labels Estimation.....	304
F.3.3.1. Binary Classification.....	304

F.3.3.2. Multi-Class Classification.....	305
F.4. Boosting-based and Averaging Ensemble Approaches	307
F.4.1. Boosting-based Ensemble Approach	307
F.4.1.1. Binary Classification.....	308
F.4.1.2. Multi-class Classification.....	309
F.4.2. Averaging Ensemble Approach	310
F.4.2.1. Binary Classification.....	310
F.4.2.2. Multi-class Classification.....	311
F.4.3. Probability of Each Class.....	312
F.4.4. Feature and Classifier Rankings based on Adequateness	312
F.4.5. Error Bound Analysis	313
F.5. Experimental Results	314
F.5.1. Metallic Surface Defect Dataset	314
F.5.2. Experiment Design.....	315
F.5.3. Model Comparisons With and Without Mislabelling.....	316
F.5.4. Feature Analysis.....	318
F.5.5. Nemenyi Post-hoc Test.....	320
F.6. Summary	320
References	324

Table Captions

Table 2.1 Summary of the characteristics of different norms.....	13
Table 2.2 Problem design theory categorization.....	20
Table 2.3 Main events of theoretical concepts.....	22
Table 2.4 Pros and cons of Jacobi iteration techniques and Gauss-Seidel techniques	24
Table 2.5 Published AVI Techniques	31
Table 3.1 Characteristics of experimental datasets	63
Table 3.2 Parameters to be optimized using particle swarm algorithm and their ranges..	66
Table 3.3 Objective function value $F(x_k)$ versus iteration number k on sum of multiple sparsities with $\alpha_1 = 0.5$ and $\alpha_2 = 0.5$	72
Table 3.4 Precision ϵ_p versus iteration number k on sum of multiple sparsities with $\alpha_1 =$ 0.5 and $\alpha_2 = 0.5$	72
Table 3.5 Objective function value $F(x_k)$ versus iteration number k on sum of multiple sparsities with $\alpha_1 = \frac{1}{1+10^5}$ and $\alpha_2 = \frac{10^5}{1+10^5}$	77
Table 3.6 Precision ϵ_p versus iteration number k on sum of multiple sparsities with $\alpha_1 =$ $\frac{1}{1+10^5}$ and $\alpha_2 = \frac{10^5}{1+10^5}$	77
Table 6.1 Comparison of (balanced) accuracies with different dictionaries.....	179
Table 6.2 Comparison of (balanced) accuracies with different resolutions.....	179
Table 6.3 Comparison of (balanced) accuracies with different resolutions and algorithms	183
Table 6.4 Instance details.....	186
Table 6.5 Instance pair details.....	186
Table 6.6 Instance pair details.....	192

Figure Captions

Figure 1.1 Issues of sparse representation problems.....	3
Figure 1.2 Issues being solved in each chapter	8
Figure 2.1 Geometric interpretations of different norms [96] - adapted from [2]	13
Figure 2.2 The geometry of the solutions of minimizations with different regularizers in 2-dimensional space - adapted from [2].....	15
Figure 2.3 Factors of visual inspection problem.....	29
Figure 2.4 Examples of surface defect: (left to right) metallic defect [350], textural defect [347], PCB defect [348] and fabric defect [349].....	30
Figure 2.5 General flow of the automated visual inspection	31
Figure 3.1 Dataset generation procedure	63
Figure 3.2 Objective function value and error vs iteration on the sum of multiple sparsities with $\alpha_1 = 0.5$ and $\alpha_2 = 0.5$	71
Figure 3.3 Objective function value Fxk and error vs iteration k on sum of multiple sparsities with $\alpha_1 = \frac{1}{1+10^5}$ and $\alpha_2 = \frac{10^5}{1+10^5}$	76
Figure 4.1 Example of a multi bouquet model.....	86
Figure 4.2 Dictionary generation procedure for the artificial dataset	109
Figure 4.3 Dictionary generation procedure for the practical dataset.....	109
Figure 4.4 Data generation procedure for the artificial dataset.....	110
Figure 4.5 ECs under changing dimension m and model parameter ρ	112
Figure 4.6 ECs under changing model parameters δ and ρ	113
Figure 4.7 ECs under changing model parameter k_1 and ρ	114
Figure 4.8 ECs under changing model parameters ν and ρ	115
Figure 4.9 ECs under changing dimension m and model parameter ρ	117
Figure 4.10 ECs under changing model parameters δ and ρ	118
Figure 4.11 ECs under changing model parameter k_1 and ρ	119
Figure 4.12 ECs under changing model parameters ν and ρ	120
Figure 4.13 Exemplar Images from Extended Yale B face database [425]	121

Figure 4.14 ECs under changing dimension m and model parameters N and ρ	122
Figure 4.15 Exemplar Images from Rolls-Royce database [5], [6]	123
Figure 4.16 ECs under changing dimension m and model parameters N and ρ	124
Figure 5.1 Proposed Method Pipeline	133
Figure 5.2 Dictionary generation procedure for artificial dataset	137
Figure 5.3 Dictionary generation procedure for practical dataset	138
Figure 5.4 Data generation procedure for both artificial and practical datasets	138
Figure 5.5 ECs under changing dimension m' and model parameter ρ	140
Figure 5.6 ECs under changing dimension m' and model parameter ρ	141
Figure 5.7 ECs under changing dimension m' and model parameter ρ when $m = 30 \times 24$ and $N = 4$	143
Figure 5.8 ECs under changing dimension m' and model parameter ρ when $m = 42 \times 48$ and $N = 9$	144
Figure 5.9 ECs under changing dimension m' and model parameter ρ when $m = 30 \times 24$ and $N = 4$	146
Figure 5.10 ECs under changing dimension m' and model parameter ρ when $m =$ 42×48 and $N = 9$	147
Figure 6.1 Image-based sparse decomposition for face recognition	153
Figure 6.2 Demonstration of anomaly information in the residuals which can be determined using sparse reconstruction approach	155
Figure 6.3 Defect detection and localization using sparse reconstruction approach	157
Figure 6.4 The examples of training images for constructing a dictionary and the input/observed image	160
Figure 6.5 Reconstruction results using very small λ	161
Figure 6.6 Reconstruction results using very large λ	162
Figure 6.7 The estimated sparse coefficient vectors x with different λ	162
Figure 6.8 Accuracies and the optimal threshold T versus λ at a fixed η	164
Figure 6.9 Accuracies and the optimal threshold T versus η at a fixed λ	164
Figure 6.10 The flow chart of the extended SRA	166
Figure 6.11 ROC curves of normal, ORDL and SADL dictionaries with different λ	178
Figure 6.12 ROC curves of different models with different λ	182

Figure 6.13 The overview of the experimental design.....	185
Figure 6.14 Examples of defect detection and localization results.....	186
Figure 6.15 Effects of low image resolution used in the first sparse reconstruction	188
Figure 6.16 Effects of trade-off coefficient β	190
Figure 6.17 Effects of kernel size S_k	191
Figure 6.18 Performance comparisons of original and extended SRAs	193
Figure 7.1 Sequential Optimization	198

Abbreviations

ADMM	Alternating Direction Method of Multipliers
AP	Average Precision
APA-APG	Adaptive Proximal Average based Accelerated Proximal Gradient method
APG	Accelerated Proximal Gradient method
AUC	Area Under the Curve
AVI	Automated Visual Inspection
Bagging	Bootstrap AGGREGatING
BFGS	Broyden-Fletcher-Goldfarb-Shanno algorithm
BGA	Ball Grid Array
BODI	Background-Oriented Defect Inspector
BP	Basis Pursuit
BRIEF	Binary Robust Independent Elementary Features
BeSEP	Binary ϵ -Stable Embedding Property
CC	Compatibility Condition
CCTV	Closed-Circuit TeleVision
CDSST	Concentric Discrete Square Sampling Template
CM	Concentration Measure
CNN	Convolutional Neural Network
CR	Coherence Rank
CS	Compressed Sensing
DB2	Daubechies 2
DB4	Daubechies 4
DCDR	Double-Center-Double-Radius
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DLSL	Dual-Light-Switching Lighting
DS	D-Skeleton

ELM	Extreme Learning Machine
EOBMM	Enhancement Operator Based on Mathematical Morphology
ERC	Exact Recovery Condition
ERP	Exact Reconstruction Principle
FAA	Federal Aviation Administration
FAR	False Acceptance Rate
FCM	Fuzzy C-Means
FCN	Full Convolutional Networks
FDCT	Fast Discrete Curvelet Transform
FFT	Fast Fourier Transform
FHRCNN	Fuzzy-based Hyper Rectangular Composite Neural Networks
FIR	Finite Impulse-Response filter
FISTA	Fast Iterative Shrinkage-Thresholding Algorithm
FN	False Negative
FP	False Positive
FRR	False Rejection Rate
GB	GigaByte
GIST	“GIST of scene”
GLCM	Gray-Level Co-occurrence Matrix
GOCM	Gradient-Only Co-occurrence Matrices
HC	$H_{s,q}$ condition
H-image	Homogeneity image
HL	Hoffman’s Lemma
HKBTN	Half K -Balancedness, k -Thickness and k -Neighborliness
HMM	Hidden Markov Model
HMMPP	Hierarchical Multi-Marked Point Process
HOG	Histogram of Oriented Gradients
HIS	Hue-Saturation-Intensity
HRCNN	Hyper-Rectangular Composite Neural Networks
IC	Integrated Circuit
i.i.d.	Independent Identically Distributed

k-NN	K-nearest neighbours
LADM	Linearized Alternating Direction Method
LBP	Local Binary Pattern
LCV	Local Common Vector
LqLA	L_1 -regularized Least Absolute
LRBM	Linear Representation Methods
L-BFGS	Limited-memory version of BFGS
mAP	mean Average Precision
MBD	Multiple Birth and Death
MC	Mutual Coherence
MFS	Multi-Fractal Spectrum
MPCNN	Max-Pooling Convolutional Neural Networks
MR	graph-based Manifold Ranking
MS-SSIM	MultiScale Structural Similarity Index Metric
NCC	Normalized Cross-Correlation
NDF	Neighbouring Different Filter
NP	Non-Parametric
NP-hard	Non-deterministic Polynomial-time Hard
NS	Null Space
NSP	Null Space Property
N.A.	Not Available
OKNP	Outwardly K -Neighborliness Property
OMP	Orthogonal Matching Pursuit
OOI	Object Of Interest
ORDL	Online Robust Dictionary Learning
PA-APG	Proximal Average based Proximal Gradient method
PA-QN	Proximal Average Quasi-Newton method
PCA	Principal Component Analysis
PCB	Printed Circuit Board
PHOG	Pyramid of Histograms of Orientation Gradients
PHOT	PHase Only Transform

PICC	Partial Information Correlation Coefficient
PPV	Positive Predictive Values
PSO	Particle Swarm Algorithm
PSVM	Polar Support Vector Machine
RAM	Random-Access Memory
RCCP	Ratio of Correctly Classified Pixels
REC	Restricted Eigenvalue Condition
RF	Random Forest
RGB	Red-Green-Blue
RIP	Restricted Isometry Property
ROC	Receiver Operating Characteristic
ROI	Region Of Interest
RRSP	Restricted Range Space Property
RSP	Range Space Property
RSSREA	Relevant Sensitive Stacked Regression Ensemble Approach
rp	Recovery Probability
R-CNN	Region-based Convolutional Neural Network
SADL	Scale Adaptive Dictionary Learning
SC	Sub-Coherence
SF	Spatial Filtering
SLBP	Smoothed Local Binary Pattern
SMC1	Sub-Mutual Coherence
SMC2	Scaled Mutual Coherence
SO	Scoop Object
SOMO	Self Organizing Map Optimization
SOAF	Scoop Object Area F-rate
SOOF	Scoop Object Object-level F-rate
SP	Solder Paste
SPAF	Solder Paste Area F-rate
SPEC	Spectral Feature
SPOE	Solder Paste Average Orientation error

SPOF	Solder Paste Object-level F-rate
SR	Sparse Reconstruction Approach (in figures)
SRA	Sparse Reconstruction Approach
SRC	Sparse Representation-based Classification approaches
SS	Sparsest Solution
SST	Shannon's Sampling Theorem
SVM	Support Vector Machine
s.t.	Such That
S-APG	Smoothed Accelerated Proximal Gradient
TN	True Negative
TNCCC	Traditional Normalized Cross-Correlation Coefficient
TP	True Positive
TWSVM	TWin Support Vector Machine
YALL1	Yang proposed alternating direction algorithm (Your Algorithms for L1) for L1-problems
UFF	Union Free Families
UP	Uncertainty Principle
URP	Unique Representation Property
VC	Verifiable Conditions
VI	Visual Inspection
VVRKFA	Vector-Valued Regularized Kernel Function Approximation

Notations

Symbol	Descriptions
$\mathbb{R}$	Real coordinate space (real numbers)
$\mathbb{R}_+$	Positive real numbers
$\mathbb{Z}$	Integers
$\mathbb{Z}_+$	Positive integers
$\mathbb{X} \in \mathbb{R}$	Finite-dimensional linear space
$x \in \mathbb{R}^n$	Sparse coefficient vector (sparse signal in compressed sensing) of an input/observed data
$y \in \mathbb{R}^m$	Vector of observations (measurements), input/observed data, observation sample
$A \in \mathbb{R}^{m \times n}$	Transform matrix (measurement matrix or dictionary) with n atoms
$e \in \mathbb{R}^m$	Error vector, corrupting vector, residuals
$\lambda \in \mathbb{R}_+$	Tradeoff parameter
$p \in \mathbb{R}_+$	Positive real value
$p' \in \mathbb{R}_+$	Positive real value
$p\text{-norm}/L_p\text{-norm} \in \mathbb{R}_+$	Function defined in (2.4)
$L_{2,p}\text{-norm} \in \mathbb{R}_+$	Function defined in (2.10)
Y	Matrix of samples/labels
M	Matrix
$m_i \in \mathbb{R}^m$	i^{th} column of matrix M
M^*	Optimal solution (matrix) of an optimization problem
$\mu \in \mathbb{R}$	Constant
$I(a) \in \mathbb{R}$	Indicator function that obtains the value 1 if $a > 0$ and 0 otherwise
$\Psi(x) \in \mathbb{R}^n$	Non-linear mapping function of x
$\varepsilon \in \mathbb{R}$	Constant/bound energy which bound the small dense noise
$\tilde{x} \in \mathbb{R}^{m+n}$	Vector of the concatenation of x and e ($\tilde{x} = \begin{bmatrix} x \\ e \end{bmatrix}$)
$\tilde{A} \in \mathbb{R}^{m \times (m+n)}$	Matrix of the concatenation of A and I ($\tilde{A} = [A \quad I]$) in which I is an identity matrix
$\alpha_i \in \mathbb{R}$	Positive constant such that $\sum_i \alpha_i = 1$
$f_i(x) \in \mathbb{R}$	Convex function (either smooth or non-smooth)
$g_i(x) \in \mathbb{R}$	Smooth continuously differentiable convex function
$h_i(x) \in \mathbb{R}$	Nonsmooth convex function
$F(x) \in \mathbb{R}$	Convex composite which contains a linear combination of individual smooth functions such that $F(x) = \sum_i \alpha_i f_i(x)$

$g(x) \in \mathbb{R}$	Smooth convex composite which contains a linear combination of individual smooth convex functions such that $g(x) = \sum_i \alpha_i g_i(x)$
$h(x) \in \mathbb{R}$	Nonsmooth convex composite which contains a linear combination of individual nonsmooth convex functions such that $h(x) = \sum_i \alpha_i h_i(x)$
$O \in \mathbb{Z}_+$	Number of smooth convex functions
$P \in \mathbb{Z}_+$	Number of nonsmooth convex functions
$N \in \mathbb{Z}_+$	Number of convex functions in objective function $F(x)$ /Number of classes in the classification
$O(\cdot)$	Big O notation
$F^* \in \mathbb{R}$	Optimal solution (a real value) of an optimization problem which convex objective function is $F(x)$
$x^* \in \mathbb{R}^n$	Optimal argument of F^* (minimizer of $F(x)$)
$u \in \mathbb{R}^n$	n -dimensional variable
$Prox_f^\gamma(x) \in \mathbb{R}^n$	Proximal mapping of function $f(x)$ with parameter γ
$\nabla f(x) \in \mathbb{R}^n$	Gradient of $f(x)$
$\nabla^2 f(x) \in \mathbb{R}^{n \times n}$	Second derivative of $f(x)$ and gradient of $\nabla f(x)$
$F_\gamma(x) \in \mathbb{R}$	Minimization problem corresponding to the proximal mapping $Prox_F^\gamma(x)$
$\tilde{F}_\gamma(x) \in \mathbb{R}$	Proximal average function of $F(x)$
$\bar{F}_\gamma(x) \in \mathbb{R}$	Optimization problem defined in eqn. (3.7)
$x_\gamma^* \in \mathbb{R}^n$	Minimizer of $\bar{F}_\gamma(x)$
$k \in \mathbb{Z}$	Integer
$\Delta x_k \in \mathbb{R}^n$	Increment step at k^{th} iteration of the optimization algorithm
$x_k \in \mathbb{R}^n$	Estimated x after k iterations of the optimization algorithm
$B_k \in \mathbb{R}^{n \times n}$	Inverse of approximated Hessian matrix at k^{th} iteration of the optimization algorithm
$B_0 \in \mathbb{R}^{n \times n}$	Initial value of B_k
$\alpha'_k \in \mathbb{R}_+$	Step size
$\alpha'_0 \in \mathbb{R}_+$	Initial value of α'_0
$s_k \in \mathbb{R}^n$	Gradient difference defined in eqn. (3.11) at k^{th} iteration of optimization algorithm
$m_n \in \mathbb{Z}_+$	Maximum number of main for loop iteration
$b_n \in \mathbb{Z}_+$	Maximum number of main for backtracking line search iteration
$0_n \in \mathbb{R}^n$	n -dimensional zero vector
$m_1 \in \mathbb{R}_+$	Parameter for first Wolfe condition
$m_2 \in \mathbb{R}_+$	Parameter for second Wolfe condition
$W_c \in \mathbb{R}$	Variable indicates whether Wolfe conditions are fulfilled
$p_k \in \mathbb{R}^n$	Descent direction at k^{th} iteration of the optimization algorithm

$p_{1,k} \in \mathbb{R}^n$	Descent direction at k^{th} iteration using the Quasi-Newton method
$p_{2,k} \in \mathbb{R}^n$	Descent direction at k^{th} iteration using the steepest gradient descent in L_2 -norm
$p_{3,k} \in \mathbb{R}^n$	Descent direction at k^{th} iteration using the steepest gradient descent in L_1 -norm
$p_{4,k} \in \mathbb{R}^n$	Descent direction at k^{th} iteration using the Nesterov's accelerated gradient descent
$e_i \in \mathbb{R}^n$	i^{th} standard basis vector
$\dot{x}_k \in \mathbb{R}^n$	n -dimensional variable
$\dot{e}_k \in \mathbb{R}^m$	m -dimensional variable
$\dot{u}_k \in \mathbb{R}^n$	n -dimensional variable
$\bar{R} \in \mathbb{R}$	Minimum ratio
$\rho \in \mathbb{R}_+$	Parameter
$U_\gamma^*(x)$	Set of variables u fulfilling eqn. (3.18) with parameter $\gamma > 0$
$u_\gamma^*(x) \in \mathbb{R}^n$	Any variable belongs to set $U_\gamma^*(x)$
$U_0^*(x)$	$U_\gamma^*(x)$ with parameter $\gamma = 0$
$u_0^*(x) \in \mathbb{R}^n$	$u_\gamma^*(x)$ with parameter $\gamma = 0$
$\bar{\varepsilon}_\gamma \in \mathbb{R}_+$	Defined as $\bar{\varepsilon}_\gamma = \bar{F}_\gamma(x) - \bar{F}_\gamma(x_\gamma^*)$
$\epsilon_\gamma \in \mathbb{R}_+$	Upper bound of $\bar{\varepsilon}_\gamma$
$\Delta\varepsilon(\gamma) \in \mathbb{R}_+$	Term which bounds the error bound of $F(x) - F(x^*)$ with $\bar{\varepsilon}_\gamma$
$\epsilon \in \mathbb{R}_+$	Upper bound of $\Delta\varepsilon(\gamma)$
$D_\gamma \in \mathbb{R}_+$	Defined as $D_\gamma = F(x_\gamma^*) - F(x^*)$
$m_\gamma \in \mathbb{R}_+$	Constant depends on parameter γ
$I_N \in \mathbb{R}^{N \times N}$	$N \times N$ -dimensional identity matrix
$C \in \mathbb{R}^{N \times N}$	Defined in eqn. (3.18)
$D \in \mathbb{R}^{1 \times N}$	Defined in eqn. (3.18)
$E \in \mathbb{R}^{N \times N}$	Defined in eqn. (3.18)
Q	Defined in eqn. (3.18)
$Q_1, \dots, Q_N$	Defined in eqn. (3.18)
$l \in \mathbb{R}$	Constant
$L \in \mathbb{R}$	Constant
$c \in \mathbb{R}$	Constant
$\eta_k \in \mathbb{R}_+$	Recursive variable defined as $\eta_k = \frac{1 + \sqrt{1 + 4\eta_k^2}}{2}$
$\rho_k \in \mathbb{R}$	Function of recursive variable η_k such that $\rho_k = \frac{1 - \eta_k}{\eta_{k+1}}$
$\check{x}_k \in \mathbb{R}^n$	Proximal mapping defined as $\check{x}_k = \text{Prox}_{\bar{F}_\gamma}^\gamma(x_{k-1})$
$L_f \in \mathbb{R}$	Constant defined in [1]
$a \in \mathbb{R}_+$	Parameter/constant

$\tau_k \in \mathbb{R}_+$	Variable defined as $\tau_k = \frac{1}{k+a}$
$\tilde{x}^1 \in \mathbb{R}^n$	Variable defined in [1]
$Diam_{\frac{1}{C^2 Q}}$	Function defined in [1]
$\tau \in \mathbb{R}_+$	Parameter (depends on 2 parameters as defined in (4.9) when it is used in Chapter 4)
$\gamma_a \in \mathbb{R}_+$	Parameter
$\bar{\gamma} \in \mathbb{R}_+$	Parameter
$\bar{\gamma}_{ADMM} \in \mathbb{R}_+$	Parameter
$o \in \mathbb{R}^n$	n -dimensional variable
$o_i \in \mathbb{R}$	i^{th} element of o
$x_o \in \mathbb{R}^n$	Ground truth/original sparse coefficient vector
$y_o \in \mathbb{R}^m$	Ground truth noiseless observation sample (measurements)
$e_o \in \mathbb{R}^m$	Ground truth/original (corrupted) error vector
$k_1 \in \mathbb{Z}_+$	Number of non-zero elements/supports of x_o
$k_2 \in \mathbb{Z}_+$	Number of non-zero elements/supports of e_o
$x'' \in \mathbb{R}^n$	n -dimensional vector
$y'' \in \mathbb{R}^m$	m -dimensional vector
$A'' \in \mathbb{R}^{m \times n}$	$m \times n$ -dimensional matrix
$e'' \in \mathbb{R}^m$	m -dimensional vector
$\epsilon_p \in \mathbb{R}$	Precision metric defined in eqn. (3.55)
$e^* \in \mathbb{R}^m$	(x^*, e^*) are optimal arguments of the optimization problem with objective function $F(x, e)$
$[n]$	Set $\{1, \dots, n\}$
$\binom{[n]}{k}$	Set of subsets $S \subset [n]$ of size k
$I \in \mathbb{Z}^{k_1}$	Support of x_o such that $I = \text{supp}(x_o) \subset [n]$ and $k_1 = I $
$J \in \mathbb{Z}^{k_2}$	Support of e_o such that $J = \text{supp}(e_o) \subset [m]$ and $k_2 = J $
$I^c \in \mathbb{Z}^{n-k_1}$	Complement set of I such that $I^c = [n] \setminus I$
$J^c \in \mathbb{Z}^{m-k_2}$	Complement set of J such that $J^c = [m] \setminus J$
$\sigma_x \in \mathbb{Z}^{k_1}$	Signs of non-zero elements of x_o such that $\sigma_x = \text{sign}(e_o(I)) \in \{\pm 1\}^{k_1}$
$\sigma \in \mathbb{Z}^{k_2}$	Signs of non-zero elements of e_o such that $\sigma = \text{sign}(e_o(J)) \in \{\pm 1\}^{k_2}$
$N_i \in \mathbb{Z}_+$	Total number of elements belonging to class i
$\tilde{x}_i \in \mathbb{R}^{N_i}$	Vector containing all x elements which belong to class i
$x_j^i \in \mathbb{R}$	j^{th} element of $\tilde{x}_i$
$\tilde{M}_i \in \mathbb{R}^n$	i^{th} row of M
$M_J \in \mathbb{R}^{ J \times n}$	Submatrix of M containing the rows indexed by J
$M_{I,J} \in \mathbb{R}^{ I \times J }$	Submatrix of M indexed by I (in rows) and J (in columns)

$M_i \in \mathbb{R}^{n \times N_i}$	Submatrix of M which consists of all rows of M but only columns which belong to class i , where each column of M belongs to a class
$M_j^i \in \mathbb{R}^{ J \times N_i}$	Submatrix of M_i containing the rows indexed by J
$M_{J,K}^i \in \mathbb{R}^{ J \times K }$	Submatrix of M_i indexed by J (in rows) and K (in columns) in which $K \subseteq [N_i]$
I	Identity matrix
$\sigma_1, \sigma_2, \dots, \sigma_{\min}(M) \in \mathbb{R}$	Singular values of M such that $\sigma_1 \geq \sigma_2 \geq \dots, \sigma_{\min}(M)$ in which $\sigma_{\min}(M)$ denotes $\sigma_{\min\{m,n\}}(M)$, i.e., the smallest singular value
$\sigma_{\max}(M) \in \mathbb{R}$	The largest singular value of M
$\ M\ \in \mathbb{R}$	L_2 operator norm of M such that $\ M\ = \sigma_{\max}(M)$
$\ M\ _F \in \mathbb{R}$	Frobenius norm of M such that $\ M\ _F = \sqrt{\text{tr}[M * M]}$
$\ M\ _{2,\infty} \in \mathbb{R}$	$L_2 \rightarrow L_\infty$ operator norm of M such that $\ M\ _{2,\infty} = \max_i \ M_i\ $
$V \subseteq \mathbb{R}^n$	Linear subspace
$P_V \subseteq \mathbb{R}^{n \times n}$	Projection operator onto V
t	Variable or time
$\text{sign}(t) \in \mathbb{Z}$	Sign of t
$\max\{\cdot\} \in \mathbb{R}$	Maximum among the elements in the set $\{\cdot\}$
$S[t] \in \mathbb{R}$	Shrinkage operator S with threshold $\frac{1}{2}$ for scalars t such that $S[t] = \text{sign}(t) \max\{t - \frac{1}{2}, 0\}$ and extended to vectors by applying it elementwise
$A_i, \tilde{A}_i \in \mathbb{R}^m$	i^{th} column of A
$B_i \in \mathbb{R}^{m \times N_i}$	Submatrix of A which consists of all atoms, a_j^i , belong to class i
$a_j^i \in \mathbb{R}^m$	j^{th} column of A_i
$\mu \in \mathbb{R}^m$	m -dimensional mean vector
$\mu_i \in \mathbb{R}^m$	m -dimensional mean vector of i^{th} class
$\mu_j^i \in \mathbb{R}$	j^{th} element of μ_i
$\mu_J^i \in \mathbb{R}^{k_2}$	Subvector of μ_i containing the elements indexed by J
$v \in \mathbb{R}_+$	Parameter
$v_i \in \mathbb{R}_+$	Parameter of i^{th} class
$v_{\max} \in \mathbb{R}_+$	Constant such that $v_{\max} = \max_i v_i$
$v_{\min} \in \mathbb{R}_+$	Constant such that $v_{\min} = \min_i v_i$
$v' \in \mathbb{R}_+$	Constant such that $v' = \frac{v_{\max}}{v_{\min}}$
$\mathcal{N}\left(\mu_i, \frac{v_i^2}{m} I_m\right) \in \mathbb{R}^m$	Normal distribution with a mean of μ_i and variance of $\frac{v_i^2}{m} I_m$
$C_\mu \in \mathbb{R}_+$	Constant
$\delta \in \mathbb{Z}_+$	Parameter

$\rho \in \mathbb{Z}_+$	Parameter
$\alpha \in \mathbb{Z}_+$	Parameter
$Y[t] \in \mathbb{R}$	Function of t defined in eqn. (4.12)
$C_s \in \mathbb{R}_+$	Constant defined in eqn. (4.10)
$v_{\delta,\rho}^*, v^* \in \mathbb{R}_+$	Constant depends on parameter δ and ρ as defined in eqn. (4.15)
$\varepsilon_{\delta,\rho,v_{\max}}^*, \varepsilon^* \in \mathbb{R}_+$	Constant depends on parameter δ, ρ and $v_{\max}$ as defined in eqn. (4.15)
$\alpha_{\delta,\rho,v_{\max}}^*, \alpha^* \in \mathbb{R}_+$	Constant depends on parameter δ, ρ and $v_{\max}$ as defined in eqn. (4.15)
$m_{\delta,\rho,v_{\max},v_{\min}}^*, m^* \in \mathbb{R}_+$	Constant depends on parameter $\delta, \rho, v_{\max}$ and $v_{\min}$ as defined in eqn. (4.15)
$o(\cdot)$	Little o notation
$q \in \mathbb{R}^{m+n-k_1-k_2}$	Certificate of optimality
$q_{1,\dots,m-k_2} \in \mathbb{R}^{m-k_2}$	Vector containing the first $m - k_2$ elements of q
$q_0 \in \mathbb{R}^{m+n-k_1-k_2}$	Candidate of q
$Z_i \in \mathbb{R}^{m \times N_i}$	$m \times N_i$ -dimensional matrix which has columns of independent and identically distributed (i.i.d.) $N\left(0, \frac{1}{m}\right)$
$Z_j^i \in \mathbb{R}^m$	j^{th} column of Z_i
$Z_j^i \in \mathbb{R}^{k_2}$	Submatrix of Z_i containing the rows indexed by J
$1_i^* \in \mathbb{R}^{m \times N_i}$	$m \times N_i$ -dimensional matrix of ones
$\tilde{x}_i \in \mathbb{R}^{N_i}$	Subvector of x_o and it contains all elements which belong to i^{th} class in x_o
$x_j^i \in \mathbb{R}$	j^{th} element of $\tilde{x}_i$
$\tilde{x}_t$	The only $\tilde{x}_i$ containing the support of x_o
v_t	v_i corresponding to the class of $\tilde{x}_i$
Z_t	Z_i corresponding to the class of $\tilde{x}_i$
I_t	Support of $\tilde{x}_t$ such that $I_t = \text{supp}(\tilde{x}_t)$
$Z_{j,I_t}^t \in \mathbb{R}^{k_2 \times I_t }$	Submatrix of Z_t indexed by J (in rows) and I_t (in columns)
$A_{J^c,I} \in \mathbb{R}^{(m-k_2) \times k_1}$	Submatrix of A indexed by J^c (in rows) and I (in columns)
$q^* \in \mathbb{R}^{m+n-k_1-k_2}$	Any solution of q to eqn. (4.17)
Φ	Linear subspace defined in eqn. (C.59)
P_Φ	Projection operator onto $\text{range}(\Phi^*)$
$\xi_s(\Phi)$	Norm P_Φ restricted to sparse vectors as shown in eqn. (4.20)
$s \in \mathbb{Z}_+$	Upper bound of $\ x\ _0$
$\mathcal{E}_I$	Event defined in Section 4.8
$\mathcal{E}_I^c$	Complement event of $\mathcal{E}_I$
$\mathbb{P}[\mathcal{E}_I^c]$	Probability of event $\mathcal{E}_I^c$
$\text{rank}(A_{J^c,I}) \in \mathbb{Z}_+$	Rank of matrix $A_{J^c,I}$

$\# \binom{[n]}{k_1} \in \mathbb{Z}_+$	Number of k_1 -combinations of the set $[n]$
$\tilde{Z} \in \mathbb{R}^{m \times (m - \delta m)}$	$m \times (m - \delta m)$ -dimensional matrix containing elements of i.i.d. $N\left(0, \frac{1}{m}\right)$
$\tilde{1}^* \in \mathbb{R}^{m \times (m - \delta m)}$	$m \times (m - \delta m)$ -dimensional matrix with all elements are 1
$\tilde{A} \in \mathbb{R}^{m \times m}$	Matrix defined in eqn. (4.52) and is a supermatrix of A
$\tilde{x} \in \mathbb{R}^m$	Vector defined in eqn. (4.53) and is a super vector of x
$\tilde{x}_o \in \mathbb{R}^m$	Vector defined in eqn. (4.53) and is super vector of x_o
$\tilde{e}_o \in \mathbb{R}^m$	Vector defined in Section 4.8 such that $J = \text{supp}(e_o) \subseteq \text{supp}(\tilde{e}_o)$ and $\text{sign}(e_o(J)) = \text{sign}(\tilde{e}_o(J))$
$\tilde{J}$	Random subset such that $\tilde{J} \subset [m]$
$\tilde{\sigma}$	Signs of non-zero elements of $\tilde{e}_o$ such that $\tilde{\sigma}(\tilde{J}^c) = 0$ and i.i.d. Rademacher
J'	defined in Lemma 13 in Appendix C
$\hat{x} \in \mathbb{R}^n$	Estimated vector of x_o
$\hat{e} \in \mathbb{R}^m$	Estimated vector of e_o
$A_i'' \in \mathbb{R}^{m \times N_i}$	$m \times N_i$ -dimensional matrix
$\mu_i'' \in \mathbb{R}^m$	n -dimensional vector
$b \in \mathbb{R}_+$	Parameter/constant
$\beta_s(M) \in \mathbb{R}_+$	Extremal submatrix singular value of M defined in (C.4)
$\gamma_s(M) \in \mathbb{R}_+$	Extremal submatrix singular value of M defined in (C.5)
$\phi_s(M) \in \mathbb{R}_+$	Extremal submatrix singular value of M defined in (C.6)
$\binom{m}{k}$	Number of k -combinations of m
$\tilde{N}$	Matrix
$\tilde{N}_i$	Matrix
$\tilde{W} \in \mathbb{R}^{m \times n}$	$m \times n$ -dimensional matrix in which $m < n$
$P_{\tilde{W}} \in \mathbb{R}^{n \times n}$	Projection operator onto the range of $\tilde{W}^*$
$W \in \mathbb{R}^{(k_1 + N) \times (m - k_2)}$	Matrix defined in (C.15) and (C.56)
$P_W \in \mathbb{R}^{(m - k_2) \times (m - k_2)}$	Projection operator onto the range of W^*
$\hat{\mu}_{J^c} \in \mathbb{R}^{(m - J) \times N}$	$(m - J) \times N$ -dimensional matrix defined as $[\mu_{J^c}^1, \dots, \mu_{J^c}^N]$
$\vartheta \in \mathbb{R}^{(m - J) \times N}$	$(m - J) \times N$ -dimensional matrix defined as $\vartheta = \frac{\hat{\mu}_{J^c}}{N\sqrt{1 - \rho}}$
$Z_{J^c, I} \in \mathbb{R}^{(m - k_2) \times k_1}$	Submatrix of Z indexed by J^c (in rows) and I (in columns)
$U \in \mathbb{R}^{(n - k_1) \times (m - k_2)}$	$(n - k_1) \times (m - k_2)$ -dimensional matrix defined in (C.53)
$\hat{U} \in \mathbb{R}^{(n - k_1) \times (m - k_2)}$	$(n - k_1) \times (m - k_2)$ -dimensional matrix in which each column is i.i.d. $N\left(0, \frac{1}{m}\right)$
$A_{[m], I} \in \mathbb{R}^{m \times k_1}$	Submatrix of A indexed by I (in columns)
$\dot{x} \in \mathbb{R}^{k_1}$	k_1 -dimensional vector

$x_o(I) \in \mathbb{R}^{k_1}$	Subvector of x_o indexed by I
J'	Subset of J
$e'_o \in \mathbb{R}^m$	m -dimensional vector such that $e'_o(J') = e_o(J')$ and $e'_o(J'^c) = 0$
$h \in \mathbb{R}^m$	m -dimensional vector
$h_J \in \mathbb{R}^{k_2}$	k_2 -dimensional vector
$(I)_{[m],J}$	Submatrix of $m \times m$ -dimensional identity matrix indexed by J (in columns)
$\tilde{B} \in \mathbb{R}^{m \times k_1}$	$m \times k_1$ -dimensional matrix which has full rank $k_1 + k_2$ such that $\tilde{B} = [A_{[m],I}(I)_{[m],J}]$
$\mathbf{0}$	Zero vector or matrix
$R \in \mathbb{R}^{(n+N) \times k_2}$	$(n + N) \times k_2$ -dimensional matrix defined in (C.54)
$T \in \mathbb{R}^{(n+N) \times k_2}$	$(n + N) \times k_2$ -dimensional matrix defined in (C.55)
$w \in \mathbb{R}^{m+n-k_1-k_2}$	$(m + n - k_1 - k_2)$ -dimensional vector
$u \in \mathbb{R}^{n+N}$	$(m + n - k_1 - k_2)$ -dimensional vector
$w_1 \in \mathbb{R}^{m-k_2}$	$(m - k_2)$ -dimensional vector
$w_2 \in \mathbb{R}^{n-k_1}$	$(n - k_1)$ -dimensional vector
$u_1 \in \mathbb{R}^{n-k_1}$	$(n - k_1)$ -dimensional vector
$u_2 \in \mathbb{R}^{k_1+N}$	$(k_1 + N)$ -dimensional vector
$\Gamma \subset \mathbb{R}^{(1-\rho)m}$	Nullspace of W
$P_\Gamma \in \mathbb{R}^{(m-k_2)+(m-k_2)}$	Projection operator onto Γ
$\tilde{y}$	Variable
Ψ	Matrix $[P_\Gamma - \lambda P_\Gamma U^*]$
$u_1^*(w)$	Optimizer of u_1 depending on w
$u^*(w)$	Optimizer of u depending on w
$\Phi^\dagger$	Right inverse (Moore–Penrose inverse) of Φ
$\mathcal{E}_\Phi$	Event defined in eqn. (C.84)
$\mathcal{E}_1$	Event defined in eqn. (C.91)
$\mathcal{E}_2$	Event defined in eqn. (C.92)
$\mathcal{E}_3$	Event defined in eqn. (C.93)
$\mathcal{E}_4$	Event defined in eqn. (C.94)
$\mathcal{E}_i$	i^{th} event which is among $\mathcal{E}_1$ to $\mathcal{E}_4$
$\mathcal{E}_R$	Event that eqns (C.97)–(C.98) hold
Z_0, J_0	Z_i and J such that any $(Z_0, J_0) \in \mathcal{E}_\Phi \cap \mathcal{E}_R$
$\ \cdot\ _\diamond$	Either the Frobenius or operator norm
$\mathcal{E}_T$	Event that eqns (C.113) and (C.114) hold
$t_1, \dots, t_n,$	Constants
$\tilde{t}$	Constant such that $\tilde{t} = \min(1, t_1, \dots, t_n)$
π	Random permutations sampled according to the uniform measure
$\pi_1(k)$	Random permutations sampled according to the uniform measure of integer k
$\# \{\cdot\} \in \mathbb{Z}_+$	Number of elements in the set $\{\cdot\}$

$\mathbb{E}f(\pi)$	Expectation of $f(\pi)$
P_{ϑ}	Matrix that projects onto the span of ϑ
$\tilde{B}$	Matrix
$\tilde{C}$	Matrix
$\tilde{q}_1 \in \mathbb{Z}_+$	Positive integer
$\tilde{q}_2 \in \mathbb{Z}_+$	Positive integer
$\dot{x} \in \mathbb{R}^n$	n -dimensional vector
$\dot{e} \in \mathbb{R}^m$	m -dimensional vector
$\hat{h} \in \mathbb{R}^m$	m -dimensional vector
$m' \in \mathbb{Z}_+$	Positive integer such that $m \gg m'$
$A' \in \mathbb{R}^{m' \times n}$	$m' \times n$ -dimensional (low dimensional) dictionary (transform matrix or measurement matrix) with n atoms
$e' \in \mathbb{R}^{m'}$	m' -dimensional (low dimensional) corrupting vector (error vector, residuals)
$N_c \in \mathbb{Z}_+$	Number of clusters
$N_{nn} \in \mathbb{Z}_+$	Number of nearest neighbours
$F_i \in \mathbb{R}^n$	i^{th} row of A (i^{th} feature vector)
$f_{ij} \in \mathbb{R}$	j^{th} element of F_i
$m'_{max} \in \mathbb{Z}_+$	Maximum feasible row number of A'
$F' \in \mathbb{R}^{m' \times n}$	$m' \times n$ -dimensional matrix
$F'_i \in \mathbb{R}^n$	n -dimensional vector and i^{th} row of F'
$\check{C} \in \mathbb{R}^{N_c \times m}$	N_c cluster centres
$L \in \mathbb{R}^m$	Labels of each F_i
$x_j^i \in \mathbb{R}$	j^{th} element of x_i
$C(y) \in \mathbb{Z}_+$	Classification result of y
$y_i \in \mathbb{R}$	i^{th} element of y
$\hat{e}_i \in \mathbb{R}$	i^{th} element of $\hat{e}$
$\bar{T} \in \mathbb{R}$	Threshold
$\hat{e}_{i,j} \in \mathbb{R}$	j^{th} element in i^{th} image
$\xi_{i,j} \in \mathbb{R}$	Slack variable corresponding to $\hat{e}_{i,j}$
$\eta \in \mathbb{R}$	Parameter
$Acc_b \in \mathbb{R}$	Balanced accuracy defined in eqn. (6.5)
$e_i \in \mathbb{R}$	i^{th} element of e
$I_{in} \in \mathbb{R}^{w \times h}$	Input/observed image ($w \times h$ -dimensional matrix)
$\hat{R} \in \mathbb{R}^{w \times h}$	Residual image ($w \times h$ -dimensional matrix)
$y' \in \mathbb{R}^{m'}$	m' -dimensional (low dimensional) input/observed data
$\hat{e}' \in \mathbb{R}^{m'}$	m' -dimensional (low dimensional) estimated corrupting vector (error vector, residuals)
$w' \in \mathbb{Z}_+$	Positive integer such that $w' < w$
$h' \in \mathbb{Z}_+$	Positive integer such that $h' < h$
$\hat{e}_r \in \mathbb{R}^{m'}$	Rounded corrupting vector (error vector, residuals)/Rounded of $\hat{e}$
$\hat{e}_{r,i} \in \mathbb{R}$	i^{th} element of $\hat{e}_r$

$D_i(T) \in \mathbb{Z}_+$	Dimensions (width and height) of the smallest bounding box of the connected region defined in eqn. (6.9)
$D_l \in \mathbb{Z}_+$	Lower bound of the dimensions $D_i(\bar{T})$
$D_u \in \mathbb{Z}_+$	Upper bound of the dimensions $D_i(\bar{T})$
$S_k \in \mathbb{Z}_+$	Size of morphological operation square kernel
$N_c(t)$	Number of estimated defective connected regions after applying eqn. (6.9) with $\bar{T} = t$ on $\hat{R}$
$S_1 \in \mathbb{R}$	Stopping criterion of accuracy
$S_2 \in \mathbb{Z}_+$	Stopping criterion of iteration number
$\beta \in \mathbb{R}$	Weight parameter
$Acc_w(\beta) \in \mathbb{R}$	Weighted accuracy depending on β
$p_{i,j} \in \mathbb{R}$	Pixel intensity of a captured image at location coordinate i and j
$n_{i,j} \in \mathbb{R}^3$	Surface normal at i and j corresponding to $p_{i,j}$
$l \in \mathbb{R}^3$	Lighting direction
$\tilde{\rho} \in \mathbb{R}_+$	Diffuse albedo
I_k	k^{th} image
$p_{i,j k} \in \mathbb{R}$	$p_{i,j}$ of k^{th} image
$n_{i,j k} \in \mathbb{R}$	$n_{i,j}$ of k^{th} image
$l_k \in \mathbb{R}^3$	Lighting direction which I_k was acquired under
$\cos\theta_{k,a,i,j} \in \mathbb{R}$	Value of $n_{k,i,j}$ in x-axis
$\cos\theta_{k,b,i,j} \in \mathbb{R}$	Value of $n_{k,i,j}$ in y-axis
$\cos\theta_{k,c,i,j} \in \mathbb{R}$	Value of $n_{k,i,j}$ in z-axis
$\cos\alpha_{k,a} \in \mathbb{R}$	Value of l_k in x-axis
$\cos\alpha_{k,b} \in \mathbb{R}$	Value of l_k in y-axis
$\cos\alpha_{k,c} \in \mathbb{R}$	Value of l_k in z-axis
$N_v \in \mathbb{Z}_+$	Number of images for which the pixels are used to express the pixel $p_{1,i,j}$ in image I_1
$c_i \in \mathbb{R}$	Variable
$e_{k,i,j} \in \mathbb{R}$	Corrupting vector (error vector, residuals) corresponding to $p_{k,i,j}$
$\check{c}$	Vector
$\tilde{D} \in \mathbb{Z}_+$	Number of variables
$\omega \in \mathbb{R}$	Exponent of matrix multiplication
$\tilde{\theta} \in \mathbb{R}$	Dual exponent of matrix multiplication
$\tilde{\epsilon} \in \mathbb{R}$	Relative accuracy
$O^*(\cdot)$	Big O star (asterisk) operator
I_t	Set of labelled training images
S	Stopping criterion
$\tilde{N} \in \mathbb{Z}_+$	Search granularity
$\lambda_o \in \mathbb{R}_+$	Optimized λ
$\eta_o \in \mathbb{R}_+$	Optimized η
$\bar{T}_o \in \mathbb{R}_+$	Optimized $\bar{T}$

$maxAcc \in \mathbb{R}$	Maximum accuracy
$maxAccp \in \mathbb{R}$	Previous maximum accuracy
$acc1a \in \mathbb{Z}_+$	Variable
$acc1b \in \mathbb{Z}_+$	Variable
$acc2a \in \mathbb{Z}_+$	Variable
$acc2b \in \mathbb{Z}_+$	Variable
$\lambda_p \in \mathbb{R}_+$	λ corresponding to the best accuracy in the previous iteration
$\eta_p \in \mathbb{R}_+$	η corresponding to the best accuracy in the previous iteration
$\lambda_{min} \in \mathbb{R}_+$	Lower bound of λ in the parameter search
$\lambda_{max} \in \mathbb{R}_+$	Upper bound of λ in the parameter search
$\eta_{min} \in \mathbb{R}_+$	Lower bound of η in the parameter search
$\eta_{max} \in \mathbb{R}_+$	Upper bound of η in the parameter search
$\tilde{\lambda} \in \mathbb{R}_+^{\tilde{N}}$	$\tilde{N}$ -dimensional vector which stores $\tilde{N}$ different λ
$\tilde{\lambda}[i] \in \mathbb{R}_+$	i^{th} element of $\tilde{\lambda}$
$\tilde{T} \in \mathbb{R}^{\tilde{N}}$	$\tilde{N}$ -dimensional vector which stores $\tilde{N}$ different $\bar{T}$
$\tilde{T}[i] \in \mathbb{R}$	i^{th} element of $\bar{T}$ corresponding to $\tilde{\lambda}[i]$
$acc1 \in \mathbb{R}^{\tilde{N}}$	$\tilde{N}$ -dimensional vector which stores $\tilde{N}$ accuracies corresponding to each element in $\tilde{\lambda}$ and $\tilde{T}$
$acc1[i] \in \mathbb{R}$	i^{th} element of $acc1$ corresponding to $\tilde{\lambda}[i]$ and $\tilde{T}[i]$
$k1 \in \mathbb{Z}_+$	Positive integer
$k2 \in \mathbb{Z}_+$	Positive integer
$k3 \in \mathbb{Z}_+$	Positive integer
$acc1 \setminus acc1[k1] \in \mathbb{R}^{\tilde{N}-1}$	Subset of $acc1$ by removing $acc1[k1]$ from $acc1$
$N_m \in \mathbb{Z}_+$	Iteration number of the main loop
$N_s \in \mathbb{Z}_+$	Iteration number
$N_t \in \mathbb{Z}_+$	Total number of images in I_t
$P_{min} \in \mathbb{R}$	Lower bounds of the parameter search range
$P_{max} \in \mathbb{R}$	Upper bounds of the parameter search range
P_c	Parameter choice
$\tilde{P} \in \mathbb{R}^{\tilde{N}}$	$\tilde{N}$ -dimensional vector which stores $\tilde{N}$ different parameter values
$\tilde{P}[i] \in \mathbb{R}$	i^{th} element of $\tilde{P}$
$P'_o \in \mathbb{R}$	Value of the fixed-parameter
$P_o \in \mathbb{R}$	Optimized parameter
$acc_t \in \mathbb{R}^{N_t}$	N_t -dimensional vector which stores N_t accuracies corresponding to each image in I_t
$acc_t[i] \in \mathbb{R}$	i^{th} element of acc_t corresponding i^{th} image in I_t
$acc \in \mathbb{R}^{\tilde{N}}$	$\tilde{N}$ -dimensional vector which stores $\tilde{N}$ accuracies corresponding to each element in $\tilde{P}$
$acc[i] \in \mathbb{R}$	i^{th} element of acc corresponding to $\tilde{P}[i]$
$acc \setminus acc[k1] \in \mathbb{R}^{\tilde{N}-1}$	Subset of acc by removing $acc[k1]$ from acc

$Acc_o \in \mathbb{R}$	Accuracy with optimized parameter P_o
$Accuracy \in \mathbb{R}$	Accuracy

Chapter 1

Introduction

“Success is not final; failure is not fatal: It is the courage to continue that counts.”

- Winston S. Churchill

Sparse representation is a subfield of linear representation methods (LRBM) [2]. One or multiple dictionaries are generally designed such that each atom in the dictionaries corresponds to an element of the sparse representation. Data reconstruction can be undertaken using the dictionaries and sparse representations. It has been widely used in many fields such as in signal processing, image processing, computer vision, pattern recognition, machine learning, and classifications.

This thesis proposes new techniques and theoretical derivations for sparse representation problems as well as presenting a case study of the sparse representation problem.

The contributions include:

- *theoretical derivation of the bound of the recovery probability (rp) of a sparse representation using an L_1 -minimization,*
- *sparse-representation based anomaly detection technique,*
- *a novel technique for nonsmooth composite convex optimization which can be used in sparse representation,*
- *a dimension reduction technique for dictionary compression under strict computational resource constraints,*

- *an application case study of using the proposed sparse-representation techniques is studied; namely: anomaly detection for image-based visual surface inspection.*

The organization and motivation of the thesis are presented as follows. Section 1.1 introduces the research statement of this thesis. Sections 1.2 and 1.3 present the motivation of problem design and optimization technique respectively for sparse representation problem. Section 1.4 outlines the organization of the thesis.

1.1. Research Statement

Two issues (see Figure 1.1), problem design and optimization, exist in sparse representation problems. The aims of resolving these issues are to improve the performance such as:

- optimal point vs ground truth variables,
- convergence point vs optimal point, and
- convergence speed

and the feasibility such as:

- computation time, and
- memory demand

of the sparse representation problem. This thesis focuses on proposing new techniques, problem reformulation methods and, theoretical derivations to improve the sparse representation problems by enhancing different components in the problems. It is expected that using any or all the proposed techniques and proposed theoretical works would improve the overall performance and feasibility in solving sparse representation problems.

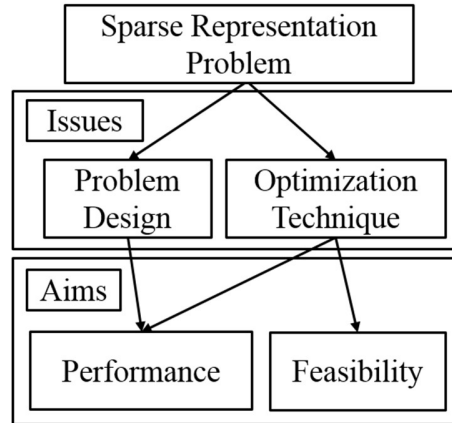


Figure 1.1 Issues of sparse representation problems

1.2. Problem design

Methods for solving the sparse representation problems include minimizations with the following norms:

- L_0 -norm,
- L_p -norm ($0 < p < 1$),
- L_1 -norm and
- L_2 -norm

of sparse representation [2]. All these methods are designed to recover the sparse representation with prior information that the representation has only a few non-zero elements. Minimization with L_0 -norm or L_p -norm ($0 < p < 1$) is generally a nonconvex optimization problem while minimizations with L_1 -norm or L_2 -norm are generally a nonsmooth convex optimization problem or a smooth convex optimization problem respectively. Among them, L_1 -norm minimization, is widely used in sparse representation problems due to the following attractive properties; namely:

- the optimization is convex,
- its analytical solution can be solved in polynomial time, and
- the sparsity of the solution can be equivalent to the one obtained by L_0 -norm minimization with full probability [3].

The theoretical derivation of the bound of recovery probability (rp) from a sparse representation recovery using an L_1 -norm minimization (eqn. (1.1)) and the assumption that the dictionary contains atoms fulfilling the bouquet model have been derived [4].

$$\begin{aligned} & \min_{x,e} \|x\|_1 + \|e\|_1 \\ & s. t. \text{ (such that) } y = Ax + e \end{aligned} \quad (1.1)$$

where

- $x \in \mathbb{R}^n$ is the sparse coefficient vector (sparse signal in compressed sensing) of a vector of observations (measurements) $y \in \mathbb{R}^m$,
- $A \in \mathbb{R}^{m \times n}$ is a transform matrix (measurement matrix or dictionary) with n atoms, and
- $e \in \mathbb{R}^m$ is an unknown vector whose non-zero elements correspond to the corrupted dimensions.

It is useful for the domain where data of each class can be represented by independent identically distributed (i.i.d) Gaussian vector and the means of data of all classes are much closed to each other; for example, face recognition. However, it is not useful for the domain where the data of each class has significantly different means such as viewpoint¹ classification for anomaly detection [5], [6].

An extension to the eqn. (1.1) is shown in eqn. (1.2),

$$\begin{aligned} & \min_{x,e} \|x\|_1 + \lambda \|e\|_1 \\ & s. t. \quad y = Ax + e \end{aligned} \quad (1.2)$$

where λ is a tradeoff parameter.

It has been used in image-based surface anomaly detection where the dictionary atoms are constructed using the images from various viewpoints of an object[5], [6]. As the vectorized images from the same viewpoint with different illumination conditions can be represented by a bouquet model as face images. The data of each viewpoint fulfils a bouquet model respectively. The atoms which belong to multiple viewpoints in the

¹ Viewpoint classification is the classification of viewpoints used in taking the input photos.

dictionary A will fulfil the multi-bouquet model (the definition can be found in Section 4.2). This minimization is a more general optimization problem than the previous one (eqn. (1.1)) and hence its theoretical derivations apply to more applications. Hence, the theoretical derivation of the estimation/recovery probabilistic lower bound using optimization eqn. (1.2) under the assumption that the atoms in dictionary A fulfil the multi-bouquet model² is developed in this work.

Anomaly detection has been used in many fields such as automated visual inspection [5], [6], network intrusion detection, fraud detection, fault detection, system health monitoring, event detection in sensor networks, and detecting ecosystem disturbances, etc. Supervised anomaly detection techniques are used in the problems where the definition of both “normal” and “abnormal” classes are clear. They can be defined by labelled (“normal” and “abnormal”) data sets [5], [6]. A novel dictionary-based sparse reconstruction anomaly detection technique [5], [6], inspired by a face recognition technique, is proposed. The approach

- identifies anomalies in an input data based on the sparse residuals (errors) obtained during data reconstruction using sparse representation, and
- learns the threshold to classify an element of the input data to be either a normal or an anomalous element based on its residual.

The proposed technique is based on the assumptions that anomalies are unwanted portions and cannot be reconstructed by normal data.

1.3. Optimization Technique

There are challenges in solving optimization problems with the nonsmooth L_1 -norm. This minimization is a nonsmooth composite convex optimization problem with one or multiple nonsmooth terms (L_1 -norms). Techniques to solve nonsmooth composite convex optimization problems have been rigorously investigated. It includes:

- parallelable Jacobi iteration type [7], and
- Gauss-Seidel type techniques [7].

² The definition of multi-bouquet model can be found in Chapter 4

Jacobi iterative type techniques can be parallelable while the Gauss-Seidel type technique only needs to store a vector that will be overwritten when they have been computed. The former will be effective when there are many nonsmooth terms in the optimization and is run in parallel while the latter benefits large problem when the resources for storing vectors are limited. The focus of this thesis is on the Jacobi iterative technique.

Existing state-of-the-art Jacobi techniques includes adaptive proximal average based proximal gradient (APA-APG) methods [1] and proximal average based accelerated proximal gradient (PA-APG) method [8]. However, these techniques only utilize first-order information of the approximations of the nonsmooth terms in optimization and do not address the use of second-order information of approximation counterpart of the nonsmooth terms in the optimization. Second-order information has been shown to be useful and can be effectively used in smooth convex optimization such as quasi-Newton techniques and Newton techniques. Hence, a more effective parallelable technique that utilizes both first and second-order information is potentially beneficial and is proposed here.

Theoretical derivation shows that using higher-dimensional signal/data and dictionary in sparse representation problems can achieve a higher sparse representation estimation/rp [4]. It is not surprising as there is more information in higher-dimensional data. However, the dimension of data is often constrained by the computational resource feasibility. One of the ways to tackle this is to reduce the dimension of the signal/data and the dictionary. The naïve downsampling method is always being used to reduce the dimension of atoms in the dictionary and signal/data. However, this approach does not utilize the information in each of the dimensions for selecting useful features. Hence, a dimension reduction method that selects important and removes redundant features using the information in them for a reduced signal/data and dictionary is proposed here. The proposed method also increases the discrimination between the dictionary atoms. The desired technique can be used for the following purposes in all applications:

- selection of useful features from original atoms (when Naïve sample-based sparse

representation technique is used [2]) using dimension reduction

- forming lower-dimensional dictionaries through dimension reduction in existing trained dictionaries (for all types of sparse representation technique). This allows their re-use in another system which may have lower computational resources than the one used for training the existing dictionaries without the need to re-train the dictionaries.

Note: from the best of the author's knowledge, there are no existing similar works for the stated second purpose.

1.4. Organization of Thesis

The thesis consists of five chapters:

Chapter 2 presents a literature review on sparse representation, anomaly detection, nonsmooth composite convex optimization, dimension reduction, and automated visual inspection.

Figure 1.2 shows the linkages between Chapters 3, 4, 5, and 6. The exemplar core problem used in this thesis is as shown in eqn. (1.1). However, the proposed techniques are not only limited to eqn. (1.1). A nonsmooth composite convex optimization technique which aims to improve the performance and feasibility of the state-of-the-art techniques by improving the convergence speed, and reducing the computation time is introduced in Chapter 3. Chapter 4 presents the theoretical work which highlights the performance; namely: the successful recovery probability lower bound of a recovery problem design using sparse representation problem (1.1). The optimal points are equivalent to the ground truth sparse representation and error of measurements. Chapter 5 introduces a clustering-based dictionary and signal dimension reduction technique to solve the feasibility problem of sparse representation problem (1.1) under computational resource constraints. Chapter 6 shows and investigates the performance of an approach using sparse representation to solve a real-world anomaly detection and localization problem.

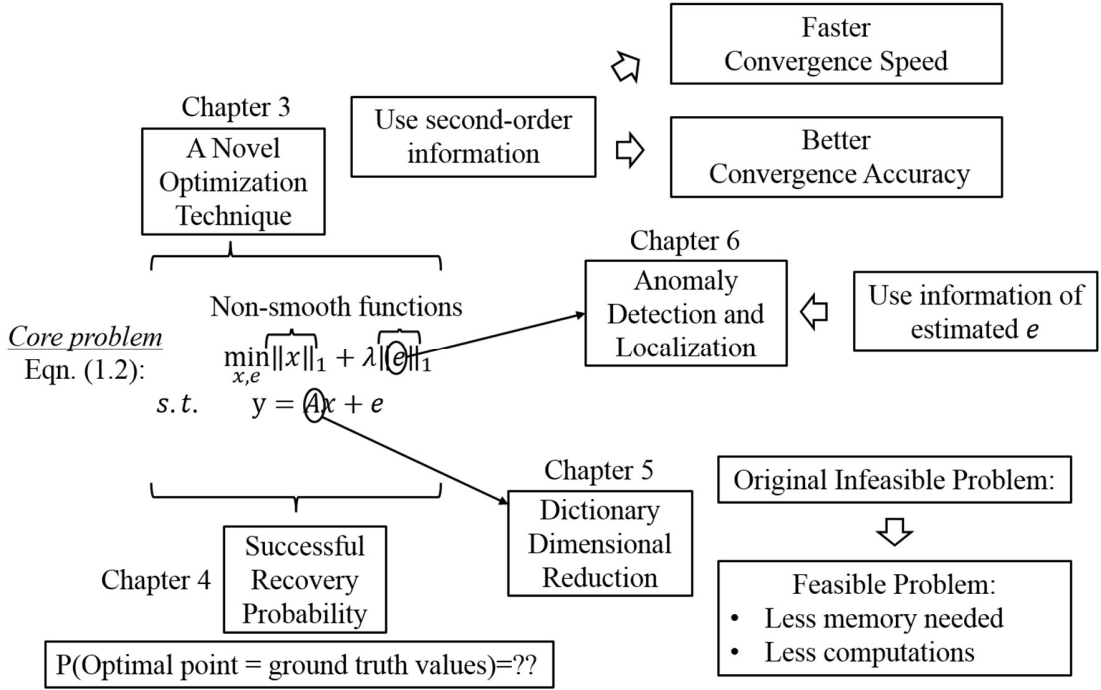


Figure 1.2 Issues being solved in each chapter

Chapter 3 introduces and elaborates on the details of a new parallelable nonsmooth composite convex optimization technique based on first and second-order information. The proposed approach is aimed to provide faster convergent speed and better convergent accuracy than the state-of-the-art methods.

Chapter 4 shows the derivation of estimation/rp of sparse representation with an L_1 -norm optimization as shown in eqn. (1.1) using a dictionary containing atoms which fulfil the multi-bouquet model.

Chapter 5 presents a clustering-based dimension reduction for the dictionary and signal under computational resource constraints. The proposed technique aims to reformulate a non-feasible sparse representation problem to a feasible problem.

Chapter 6 presents a novel sparse representation-based anomaly detection technique and its application in automated visual inspection. The proposed technique uses a core problem eqn. (1.1) to solve the inspection issue with limited resolution.

Chapter 7 concludes the thesis by reviewing the contributions of the work and highlighting the limitation and discusses the future works.

Chapter 2

Literature Review

“Any fool can know. The point is to understand.”

- Albert Einstein

In this chapter, sparse representation, nonsmooth convex optimization approaches, and automated visual inspection (AVI) are reviewed.

Section 2.1 (Sparse representation) reviews:

- *the origin of sparse representation (Section 2.1.1),*
- *the regularizations used in sparse representation problem (Section 2.1.2),*
- *a special section for L1-L1 regularizer (Section 2.1.3), and*
- *the main theory contribution in sparse representation (Section 2.1.3),*
- *the dimensionality problem and its solutions (Section 2.1.5)*

and Section 2.2 reviews automated visual inspection (AVI). Section 2.3 summarizes and concludes the chapter.

2.1. Sparse Representation

Sparse representation (a.k.a. sparse approximation), is a well-studied linear representation method (LRBM). It seeks sparse solutions for systems of linear equations [2], [9]–[12]. Sparse representation is one of the prerequisites of compressed sensing (CS) theory [2], [13]–[16]. Over the past decade, it has been widely used in many applications such as signal processing [17]–[21], image processing [22]–[26], pattern recognition [27]–[34], machine learning [35]–[39][40], etc.

2.1.1. Originality

The origin of sparse representation is related to CS [2], [41]–[44]. The earliest concept of compressed sensing was proposed by Donoho [41] that signal is suggested to be sparse or compressive. A sparse signal can be reconstructed with fewer observations (measurements) than Shannon's sampling theorem (SST). Candes et al. [43] also stated that the original signal can be reconstructed using a small portion of Fourier transformation coefficients. A theoretical analysis of CS is provided by Baraniuk [42] with a specific interpretation based on some signal reconstruction algorithms. Since then, sparse representation, encoding measurement, and reconstruction algorithm become the three prerequisite components in CS theory. Moreover, many researchers endeavour in applying the sparse representation concept to address representation and reconstruction problems in many domains [45]–[50] and found it to be very beneficial.

2.1.2. Regularizations

The original purpose of sparse representation in compressed sensing is to recover a sparse signal (sparse coefficient vector) $x \in \mathbb{R}^n$ by finding solutions to undetermined linear systems in eqn. (2.1) [41].

$$y = Ax \quad (2.1)$$

where

$y \in \mathbb{R}^m$	is a vector of observations (measurements), and
$A \in \mathbb{R}^{m \times n} (m \ll n)$	is a transform matrix (measurement matrix or dictionary).

The most natural solution is the sparsest possible x which can be estimated by optimization problem in eqn. (2.2).

$$\begin{aligned} & \min_x \|x\|_0 \\ & \text{s. t. (such that)} \quad y = Ax \end{aligned} \quad (2.2)$$

However, eqn. (2.1) is a non-deterministic polynomial-time hard (NP-hard) problem [3], [11], [51] which cannot be solved in polynomial time. Various methods have been proposed to solve the problem [52]–[61]. From a computational point of view, the approximate L_1 -norm minimization (Basis Pursuit (BP) problem [55], [62]–[67]) in eqn. (2.3) can be

considered to replace eqn. (2.2) for which there are many efficient algorithms available to solve the problem in polynomial time [3], [55], [68]. Moreover, the solutions of both eqns (2.2) and (2.3) can be equivalent with full probability under certain conditions [64], [65], [69]–[95].

$$\begin{aligned} \min_x & \|x\|_1 \\ \text{s.t. } & y = Ax \end{aligned} \quad (2.3)$$

Other than BP, many other approximation problems of eqn. (2.2) using regularizations with other norms (regularizers) are also being investigated for achieving sparsest representation and sparse signal recovery. Hence, the sparse representation problem can be generally categorized into one of the following groups based on the main regularizers used in the problem:

- L_0 -norm minimization,
- L_1 -norm minimization,
- L_p -norm ($0 < p < 1$) minimization,
- L_2 -norm minimization, and
- $L_{2,p}$ -norm ($0 < p < \infty$) minimization.

The characteristics of different norms (regularizers) and the corresponding minimization examples are as follows.

A p -norm or L_p -norm of coefficient vector x is defined as shown in eqn. (2.4),

$$\|x\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}} \quad (2.4)$$

where

- p is a positive constant, and
- n is the total number of elements of x .

The properties of the L_p -norm changes significantly with the change of p . Figure 2.1 shows the geometric interpretations of different norms and Table 2.1 summarizes the characteristics of different norms. In Figure 2.1, the value of the function of L_p -norm

(eqn.(2.4)), $f(x)$ (see eqn. (2.5)), is shown in the y-axis while the x-axis presents the value of x_i which is the i^{th} element of x . Any L_p -norms of x_i with arbitrary p have values of 0 at $x_i = 0$ and values of 1 at $x_i = 1$ or -1 .

$$f(x) = \|x\|_p^p = \sum_{i=1}^n |x_i|^p \quad (2.5)$$

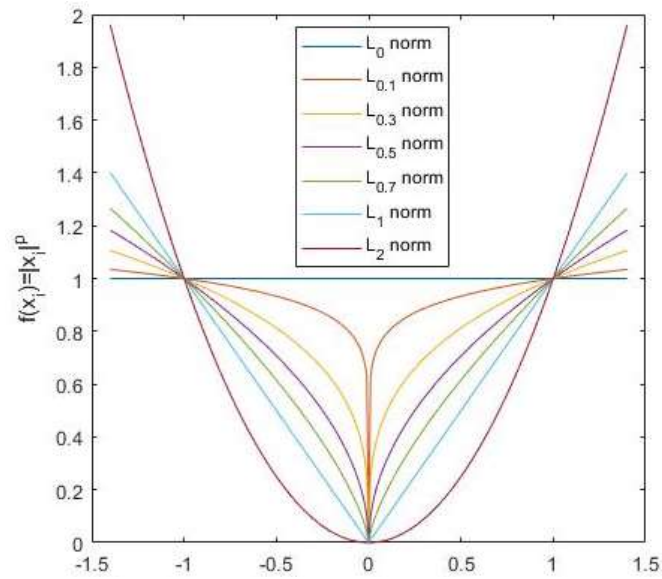


Figure 2.1 Geometric interpretations of different norms [96] - adapted from [2]

Table 2.1 Summary of the characteristics of different norms

Function	Convexity	Smoothness	Continuity	Global differentiability
L_0 -norm	No	No	No	No
L_1 -norm	Yes	No	Yes	No
L_p -norm ($0 < p < 1$)	No	No	Yes	No
L_2 -norm	Yes	Yes	Yes	Yes

L_0 -norm is nonconvex, nonsmooth, discontinuous, and globally nondifferentiable. It is always used to measure the sparsity (the number of nonzero elements) of a vector. Eqns. (2.6) to (2.9) with $p = 0$ and $p' = 1$ are several examples of typical L_0 -norm minimization [2], [44], [97]–[100].

$$\hat{x} = \underset{x}{\operatorname{argmin}} \|x\|_p \text{ s.t. } y = Ax \quad (2.6)$$

$$y = Ax \text{ s.t. } \|x\|_p \leq k \quad (2.7)$$

$$\hat{x} = \underset{x}{\operatorname{argmin}} \|x\|_p^{p'} \text{ s.t. } \|y - Ax\|_2^2 \leq \varepsilon \quad (2.8)$$

$$\hat{x} = \underset{x}{\operatorname{argmin}} \|y - Ax\|_2^2 \text{ s.t. } \|x\|_p^{p'} \leq \varepsilon \quad (2.9)$$

where

ε, p, p' and λ are positive constants, and

A is the linear matrix (dictionary/design matrix).

Eqns. (2.6) and (2.7) with $p = 0$ assume the noiseless measurements y can be perfectly mapped to Ax . Eqns. (2.8) and (2.9) with $p = 0$ and $p' = 1$ assume that the measurements $y = Ax + e$ are noisy where e represents the noise vector and $\|e\|_2^2$ is bounded by a constant ε in eqn. (2.8). L_0 -norm minimization is an NP hard (non-deterministic polynomial time hardness) problem and cannot be solved by traditional convex optimization approaches. Many works have been explored in looking for the relaxation of L_0 -norm minimization. L_1 -norm minimization [2], [44], [66], [101]–[105][52], [53] (described in the following), which can be solved by many approaches in polynomial time, are widely used to approximate the L_0 -norm minimization.

L_1 -norm which originates from the Lasso problem [2], [44], [106]–[112] is convex, nonsmooth, global nondifferentiable. Recent works show that the solutions obtained from L_1 -norm minimization are also sparse and they can be equivalent to the solutions obtained by L_0 -norm minimization with full probability under certain conditions [64], [65], [69]–[95]. Some examples of L_1 -norm minimization are shown in eqns. (2.6), (2.8) and (2.9) with $p = p' = 1$.

L_p -norm ($0 < p < 1$) is nonconvex, nonsmooth, global nondifferentiable. Due to these properties, L_p -norm minimization [45], [102], [113]–[117] is not the mainstream approach in sparse representation problem. Eqns (2.8) and (2.9) with $0 < p < 1$ and $p' = p$ are the examples of L_p -norm ($0 < p < 1$) minimization.

L_2 -norm is convex, smooth, and globally differentiable. L_2 -norm minimization [2], [3], [96], [118] has a closed-form solution. However, the solution to L_2 -norm minimization is not sufficiently sparse (see Figure 2.1). Hence, L_2 -norm minimization is not commonly used in sparse representation problem. Eqn. (2.8) with $p = p' = 2$ is an example of L_2 -norm minimization.

Figure 2.2 shows the geometry of the solutions of minimizations with different regularizers in 2-dimensional space - adapted from [2]. (a), (b) and (c) are the geometry of the solutions of the L_0 -norm, L_1 -norm, L_2 -norm minimizations, respectively.

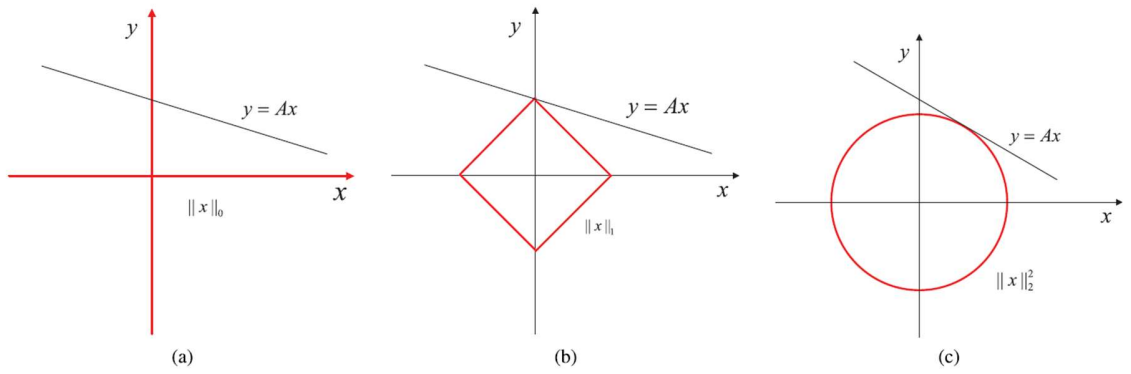


Figure 2.2 The geometry of the solutions of minimizations with different regularizers in 2-dimensional space - adapted from [2].

$L_{2,p}$ -norm ($0 < p < \infty$) of a matrix $M \in \mathbb{R}^{n \times m}$ is defined by the following eqn. (2.10):

$$\|M\|_{2,p} = \left(\sum_{i=1}^n \left(\sqrt{\sum_{j=1}^m m_{i,j}^2} \right)^2 \right)^{\frac{1}{p}} = \left(\sum_{i=1}^n \|m_i\|_2^p \right)^{\frac{1}{p}} \quad (2.10)$$

where

$$m_i = [m_{i,1}, m_{i,2}, \dots, m_{i,m}]$$

$L_{2,p}$ -norm ($0 < p < \infty$) minimization [119]–[124] has been widely used in blocked (group) sparsity problem. The solution M tends to have only few m_i (please refer to eqn. (2.10)) containing non-zero elements. An example of $L_{2,1}$ -norm minimization [125]–[127] is shown in eqn. (2.11).

$$M^* = \underset{M}{\operatorname{argmin}} \|Y - AM\|_{2,1} + \mu \|M\|_{2,1} \quad (2.11)$$

where

μ is a small positive constant,

$Y = [y_1, y_2 \dots, y_m]$ refers to a matrix composed of samples, and

$M = [m_1, m_2 \dots, m_m]$ is a coefficient matrix.

$L_{2,p}$ -norm ($0 < p < \infty$) minimization is a relaxed problem of $L_{2,0}$ -norm minimization [128] in which $L_{2,p}$ -norm is an approximation of the $L_{2,0}$ -norm as defined in eqn. (2.12).

$$\|M\|_{2,0} = \sum_{i=1}^n I(\|m_i\|_2) \quad (2.12)$$

where

$I(a)$ is an indicator function that obtains the value 1 if $a > 0$ and 0 otherwise,

$M \in \mathbb{R}^{n \times m}$ is a matrix, and

$m_i = [m_{i,1}, m_{i,2}, \dots, m_{i,m}]$

$L_{2,1}$ -norm minimization is often used in feature selection [119], [129]–[132], discriminative analysis [133]–[136] as well as rotation-invariant principle component analysis (PCA) [137], [138].

The sum of multiple regularizers is also widely used in sparse representation problems for recovering multiple sparse vectors which may have different characteristics. Moreover, a new type of compressed sensing so-called non-linear compressed sensing is investigated recently. Unlike the above-mentioned linear regularizations, its problem contains non-linear regularization. Eqn. (2.13) shows an example of L_1 -norm minimization with non-linear regularization [139]–[143].

$$\min_x \|y - \Psi(x)\|_1 \quad (2.13)$$

where $\Psi(x)$ is a non-linear mapping function of x

2.1.3. L_1 - L_1 Regularization

Inspired by BP in eqn. (2.3), Wright et al. [3] proposed a sparse representation-based classification approaches (SRC) for face recognition. Two L_1 -norm minimizations in eqns

(2.14) and (2.15) are proposed in SRC for estimating the sparse representation of the input (observed) image, respectively, for dealing with:

- small dense errors (noise), and
- sparse errors/noise (due to occlusion or corruption).

present in the input image.

$$\begin{aligned} & \min_{x,e} \|x\|_1 \\ \text{s.t.} \quad & \|Ax - y\|_2 \leq \varepsilon \end{aligned} \quad (2.14)$$

where

- ε is bound energy which bound the small dense noise,
- x is the sparse coefficient vector of the input image,
- y is the vector of observations/measurements which is formed by the input image by stacking columns or rows of the input image into a vector
- A is the dictionary that stores the information of all labelled training images using the stacking columns and rows of all normalized training images as atoms of A , and all atoms are labelled.

$$\begin{aligned} & \min_{x,e} \|x\|_1 + \|e\|_1 \\ \text{s.t.} \quad & y = Ax + e \end{aligned} \quad (2.15)$$

where e is the error vector due to occlusion or corrupted pixels.

Interestingly, Wright et al. showed in a theoretical work [3] that eqn. (2.15) can also deal with up to 100% of the error percentage (dense error) when dimension m increases (assuming the dictionary fulfilling the *bouquet model* [3]). Moreover, the work also shows that the dictionary can have arbitrary size and does not need to be overcomplete. Furthermore, eqn. (2.15) can be reformulated into a BP problem in eqn. (2.16) which can be solved by many efficient techniques such as linear programming [144], [145].

$$\begin{aligned} & \min_{x,e} \|\tilde{x}\|_1 \\ \text{s.t.} \quad & y = \tilde{A}\tilde{x} \end{aligned} \quad (2.16)$$

where

$\tilde{A} = [A \quad I]$ is a matrix of the concatenation of A and I in which I is an identity matrix,

$\tilde{x} = \begin{bmatrix} x \\ e \end{bmatrix}$ is a vector of the concatenation of x and e

Yang et al. proposed an alternating direction algorithm (a.k.a. YALL1), for solving the L_1 -norm minimization in eqn. (1.2) which is a generalized problem of eqn. (2.15).

$$\begin{aligned} & \min_{x,e} \|x\|_1 + \lambda \|e\|_1 \\ \text{s.t.} \quad & y = Ax + e \end{aligned} \quad \begin{array}{l} \text{Recap} \\ (1.2) \end{array}$$

where λ is a tradeoff parameter. Moreover, it is also interesting to note that eqn. (1.2) is also the Lagrangian function of eqn. (2.17) with $p = 1$ and there is a connection between the parameters λ and ε in eqns (1.2) and (2.17) respectively.

$$\begin{aligned} & \min_{x,e} \|x\|_1 \\ \text{s.t.} \quad & \|Ax - y\|_p \leq \varepsilon \end{aligned} \quad (2.17)$$

Some theoretical works [146] has been done on eqn. (2.17) and hence these works can be transferred for analyzing eqn. (1.2) in the future. Both eqn. (2.15) and YALL1 have been widely used in signal denoising and error correction [147]–[151], super-resolution reconstruction [152], background subtraction [153], and dictionary learning [153]. SRC is the foundation in many works such as in face recognition [62], [154]–[158], linear regression [159], etc [65].

2.1.4. Theory Contributions

Theories of sparse representation can be categorized into either problem design or optimization technique theories. This section reviews the contributions of

- main problem design theories, and
- optimization technique theories that investigate the computational complexity, the convergence analysis, and the performance of an algorithm for solving non-smooth sparse representation convex problem.

Sparse representation and compressed sensing are highly related. Many theories derived for compressed sensing can also be applied to the sparse representation. In the past decades, several concepts were introduced for certain guarantees or used for the theoretical derivations. Table 2.2 and Table 2.3 presents the categorization of problem design theories and the main events related to the concepts used in deriving these theories in chronological

order. In Table 2.3, $P_{0,1}, P_{0,2}, P_{1,1}, P_{1,2}, P_{1,3}, P_{1,4}, P_{1,5}, P_{1,6}$ and P_p represents L_0, L_1 and L_p -norm minimizations in eqns (2.2), (2.18) with $p = 0$, (2.3) (BP), (2.18) with $p = 1$, (2.14), (2.17) with $p = 1, 2, \infty$ and (2.4) respectively. The sparsest solution (SS) refers to the sparsest possible x fulfilling eqn. (2.1) when y and A are given.

$$\begin{aligned} & \min_{x,e} \|e\|_p \\ \text{s.t.} \quad & y = Ax + e \end{aligned} \tag{2.18}$$

where

- $x \in \mathbb{R}^n$ is the coefficient vector (sparse signal in compressed sensing) of a vector of observations (measurements) $y \in \mathbb{R}^m$,
- $A \in \mathbb{R}^{m \times n}$ is a transform matrix (measurement matrix or dictionary) with n atoms,
- $e \in \mathbb{R}^m$ is an unknown vector whose non-zero elements correspond to the corrupted dimensions, and
- $p \in \mathbb{R}$ is a real number.

Nonsmooth convex optimization techniques are widely used in solving sparse representation problem which involves L_1 -norm minimizations.

A multi-term convex optimization problem which contains nonsmooth composite can be represented in the following form (eqn. (2.19)).

$$F^* := \min_{x \in \mathbb{X}} F(x) = \min_{x \in \mathbb{X}} G(x) + H(x) \tag{2.19}$$

with $\mathbb{X}$ is a finite-dimensional linear space, $G(x) = \sum_{i=1}^O \alpha_i g_i(x)$ and $H(x) = \sum_{i=O+1}^{O+P} \alpha_i h_i(x)$,

where

- $\alpha_i \geq 0$ is a constant such that $\sum_i \alpha_i = 1$,
- $g_i: \mathbb{X} \rightarrow (-\infty, \infty)$ is a smooth continuously differentiable convex function,
- $h_i: \mathbb{X} \rightarrow (-\infty, \infty)$ is a nonsmooth convex function, and
- O and P are the total numbers of functions g_i and h_i in the smooth and nonsmooth convex composites, $G(x)$ and $H(x)$.

Table 2.2 Problem design theory categorization

Perspective	Category		Reference
Recovery Target	The sparsest representation		[52], [53], [59], [64], [73], [83], [85], [90]–[94], [96], [160]–[163]
	The sparse solution		[61], [64], [67], [164]–[168]
	The original sparse signal or the sparse coefficient vector x_o		[55], [56], [64], [70], [72], [76], [82], [86], [88], [89], [95], [161]–[163], [169]–[180]
	The sign/support of the sparsest representation or the sparse solution		[64], [181]–[192]
Observed measurement quality	Noiseless		[52], [53], [55]–[61], [64], [67], [77], [82], [85], [86], [89], [90], [92], [95], [96], [161], [167], [170], [180], [186], [187], [193]–[198]
	Noisy		[41], [55], [64], [73], [84], [88], [89], [95], [96], [107], [112], [168], [171], [179], [182], [183], [186], [188]–[193], [195], [197], [199]–[206]
Result type	Uniqueness	The uniqueness of the original problem (e.g. L_0 -norm minimization for sparsest representation)	[59], [64], [73], [81], [83], [85], [90]–[94], [96], [160], [162], [163]
		The uniqueness of the relaxed problem (e.g. L_1 -norm minimization for sparsest representation)	[64], [73], [77], [78], [81]–[88], [91], [92], [146], [161], [162], [167], [169], [181], [185], [207]–[209]
	Equivalence between the original and the relaxed problems (e.g. L_1 and L_0 - norm minimization equivalence)		[52], [53], [64], [65], [79], [81], [83], [85], [161]–[163]
	Exact recovery of original signal		[64], [76], [82], [89], [95], [161], [162], [169], [170], [176], [207], [210]
	Stability of the problem		[53], [64], [71], [73], [77], [88], [89], [92], [93], [95], [96], [112], [163], [170], [171], [186], [187], [193]–[195], [199]–[204], [211]–[227]

	Successful recovery probability using the problem	[64], [77], [81], [169], [173], [186], [224]
Theory/Principle Used	Uncertainty principle (UP)	[90], [169], [196], [228], [229]
	Unique representation property (URP)	[64], [167]
	Mutual coherence (MC) (including sub-mutual coherence (SMC1) scaled mutual coherence (SMC2), coherence rank (CR), and sub-coherence(SC))	[59], [64], [65], [71], [73], [90]–[94], [96], [163], [193], [194], [198], [211], [221], [222], [229]–[232]
	Spark	[64], [79], [92], [96], [160], [233]
	Exact recovery condition (ERC)	[64], [82], [83], [162], [207]
	Uniform uncertainty principle (UUP)	[234]
	Exact reconstruction principle (ERP)	[234]
	Restricted isometry property (RIP)	[64], [69]–[77], [89], [95], [124], [160], [163], [170], [171], [193], [199], [205], [210], [212], [220], [235]–[237]
	Concentration measure (CM)	[89], [186], [234], [238]
	Null space (NS)	[41], [64], [239]
	Outwardly k -neighborliness property (OKNP)	[64], [86]
	Half k -balancedness, k -thickness and k -neighborliness (HKBTN)	[64], [78], [80]
	d-skeleton (DS)	[197]
	Verifiable conditions (VC)	[64], [84], [88]
	Restricted eigenvalue condition (REC)	[64], [223]
	Compatibility condition (CC)	[64], [240]
	Null space property (NSP)	[64], [73], [77], [78], [80], [81], [87], [90], [92], [94], [160], [178], [220]
	$H_{s,q}$ condition (HC)	[64], [225], [226]
	binary ϵ -stable embedding property (BeSEP)	[186]
	Range space property (RSP)	[53], [61], [64], [85], [161], [173], [204], [213], [214]
	Union free families (UFFs)	[187] [241]
	Restricted range space property (RRSP)	[64], [181], [185]
	Hoffman's Lemma (HL)	[64], [213]

Table 2.3 Main events of theoretical concepts

Year	Event
1989	Donoho and Stark used <i>UP</i> to prove that a sparse wideband signal can be recovered from noiseless narrowband data by $P_{1,1}$, provided the signal is sufficiently [196].
1997	Rao and Gorodnitsky stated that if $y = Ax^*$, $A \in \mathbb{R}^{m \times n}$ and $\ x^*\ _0 = p < \frac{m}{2}$, x^* is the unique <i>SS</i> under <i>URP</i> [167].
2001	Donoho and Huo proved that a unique highly sparse representation can be recovered from a discrete-time signal by both $P_{0,1}$ and $P_{1,1}$ if it is representable as a highly sparse superposition of atoms from a time/frequency dictionary via <i>UP</i> [90].
2001	Elad and Bruckstein showed the uniqueness of the <i>SS</i> for a linear system in which dictionary A is a concatenation of two orthobases via <i>MC</i> and <i>UP</i> . [229]
2003	Parallel to the work presented by Rao and Gorodnitsky in 1997 [167], Donoho and Elad show the uniqueness of the <i>SS</i> via <i>spark</i> [64], [79], [92], [96].
2003	As computing the <i>spark</i> is an NP-hard problem, Donoho and Elad extended Elad and Bruckstein's work in 2001 [229] and presented a simpler version of the uniqueness of the <i>SS</i> via <i>MC</i> [90]–[93], [229], [242]
2004	Tropp introduced <i>ERC</i> for proving a sufficient condition of <i>SS</i> recovery with <i>OMP</i> (<i>Orthogonal Matching Pursuit</i>) [243] and $P_{1,1}$ [82].
2004	Donoho and Tsaig formally introduced <i>compressed sensing</i> and a compressed data acquisition protocol via a nonlinear procedure. Moreover, BP was shown to be able to achieve near-optimal sparsest representation when the dictionary is under certain conditions. The robustness proofs were provided with <i>NS</i> [41], [239], [244].
2005	Candès and Tao proved a stronger theorem that guarantees both <i>UUP</i> and <i>ERP</i> [232] with a newly introduced <i>RIP</i> and <i>CM</i> [89]. The theorem shows that under suitable conditions on the dictionary A , a corrupted signal/image can be recovered by $P_{1,2}$ even the corruption is significant. <i>RIP</i> is used to prove the equivalence between L_0 and L_1 -norm in sparse signal recovery. When <i>RIP</i> is fulfilled, the solutions of the two following minimization pairs are the same respectively: (1). $P_{0,1}$ and $P_{1,1}$ when x is sufficiently sparse, and (2). $P_{0,2}$ and $P_{1,2}$ when e is sufficiently sparse.
2005	Donoho and Tanner introduced <i>OKNP</i> for the successful recovery of a sufficient sparse nonnegative signal [86].
2005	Zhang introduced <i>HKBTN</i> and derived an equivalent form of the results derived by Donoho and Tanner in 2005 [86] with <i>HKBTN</i> [78], [80].
2006	Candès and Tao used <i>UUP</i> , <i>ERP</i> , and <i>MC</i> to prove that signal made out of spikes can be recovered by a $P_{1,1}$ with overwhelming probability with considering the recovered signal encounter power-law decay [232].

2007	Sharon et al. extended Candès and Tao's work in 2005 [89]. with proposing a concept of DS [197]. It is used to determine the breakdown points B_1 and B_2 (defined below) for the following minimization pairs: (1). $P_{0,1}$ and $P_{1,1}$ and (2). $P_{0,2}$ and $P_{1,2}$ The solutions of minimization pairs will be the same respectively under the following conditions: (1). $B_1 \geq \ x^*\ _0$ and (2). $B_2 \geq \ e^*\ _0$ where x^* and e^* are solutions of the minimization pairs respectively.
2009	Juditsky et al. introduced a computational efficient verifiable sufficient condition (VC) for recovery of sufficient sparse signal via $P_{1,3}$ [84], [88], [245].
2009	Bickel et al. introduced REC and showed that the stability of LASSO can be guaranteed under REC [223].
2009	Geer and Bühlmann presented CC in studying the stability of LASSO [240].
2009	Cohen et al. introduced NSP which provides the degree of robustness of P_p in signal reconstruction (including non-sparse signal) [77]. The works also used RIP and CM to provide the necessary and sufficient conditions of sparsest representation via $P_{1,1}$ [77].
2011	Juditsky et al. introduced HC in studying the stability of multiple a block sparse signal recovery problem and two sparse signal recovery problems (L_1 -norm minimizations) [225], [226].
2013	Jacques et al. introduced $B\epsilon SEP$, which characterizes the robustness of the 1-bit compressed sensing measurement process to sign changes. Jacques et al. provided the lower bound on the best achievable reconstruction error and demonstrated nearly optimal error decay is achievable with overwhelming probability when the measurement mappings are provided by i.i.d. random Gaussian matrices for both noiseless and noisy scheme [186].
2013	Zhao introduces $SMC1$, $SMC2$, CR , and SC to improve the proofs of uniqueness conditions for the sparsest solution from $P_{1,1}$ with MC [198].
2013	Zhao introduced RSP and used it to prove the uniqueness of the sparsest solution of $P_{1,1}$ [85], [161].
2013	Gopi et al. introduced a 1-bit compressed sensing approach using $UFFs$ of sets for both noiseless and noisy scheme. Gopi et al. proved that the approach can achieve successful support recovery for a nonnegative signal with overwhelming probability via concentration measure [241].
2014	Cai and Zhang improved the bound of constant δ_{2k} which is used in RIP to $\delta_{2k} < \frac{1}{\sqrt{2}}$ [76].
2016	Zhao and Xu introduced an $RRSP$ for the uniqueness of 1-bit basis pursuit and proposed a necessary assumption for successful recovery of 1-bit measurements (the sign/support of the inner products of the sparse signal with measurement vectors) [181].
2018	Zhao et al. used HL in studying the stability of L_1 -norm minimizations ($P_{1,1}P_{1,4}$, $P_{1,5}$, $P_{1,6}$) [213].

A nonsmooth convex optimization technique can be generally categorized into one of the following two categories:

- Jacobi iteration techniques [7], and
- Gauss-Seidel techniques [7].

Table 2.4 shows the pros and cons of the two technique types.

Table 2.4 Pros and cons of Jacobi iteration techniques and Gauss-Seidel techniques

	Jacobi Iteration techniques	Gauss-Seidel techniques
Pros	can be accelerated with parallel computing	do not need large storage to store each copy of vectors for parallel computing and hence it is more suitable for large problem
Cons	need large storage to store each copy of vectors for parallel computing and hence it may less suitable for large problem	cannot be accelerated with parallel computing

Not all the techniques can solve all the problems, some techniques only work for problems which contain one or two nonsmooth convex terms.

Optimization problem in eqn. (2.19) has been widely used in machine learning, such as overlapping group Lasso [246]–[248], graph guided fused Lasso [249], [250], graph-guided logistic regression [251]–[253], regularized risk minimization problems [254], [255], simultaneous low-rank and sparsity [256][257], variational image restoration [249], [258], [259] and sum of multiple sparsities (L_1 -norms) [3]–[6], [147], [260]. Prior information such as structured sparsity [261]–[263] and nonnegativity[264]–[266] is introduced as multi-term nonsmooth regularizer $g(x)$. It has been shown that the regularization term enhances the solution's accuracy. Among all applications, the main challenge is the multi-term nonsmooth composite $g(x)$ as it becomes complex when $P > 1$.

Many Jacobi iteration techniques have been proposed for nonsmooth composite convex optimization. First-order accelerated gradient type methods which enjoy $O(\frac{1}{k^2})$ were proposed by Nesterov [267]–[269] (APG method) for $P = 0$ and by Beck and Teboulle

[270], [271] (FISTA method) for $P = 1$. A subgradient type method [272], [273] with slow iteration complexity of $O(\frac{1}{\sqrt{k}})$ was proposed for $P > 1$. Nesterov proposed a smoothed accelerated proximal gradient (S-APG) method [8], [274] which can achieve $O(\frac{1}{k})$ when the smoothing parameter is smaller than $O(\frac{1}{k})$. However, the small smoothing parameter causes the resultant small iteration step size which can be inefficient and problematic in practical applications. In order to make the method to be more practical, several adaptive smoothing techniques [275], [276] inspired by Nesterov's smoothing approach were proposed. However, the smoothing technique may destroy the important structure contained in those nonsmooth terms. The larger the value of the P , the higher the values of error which is linearly increasing with the value of P . To solve the approximation error issue, proximal average approximation which can reduce the error by utilizing the proximal mapping information of each nonsmooth term is proposed to tackle the problem [277]. Proximal average-based accelerated proximal gradient (PA-APG) method [8] is a method that applies proximal average approximation and the accelerated proximal gradient technique. The results of PA-APG is promising when compared against the Nesterov's smoothed accelerated proximal gradient method [8]. However, the method only enjoys $O(\frac{1}{k^2})$ when a small approximation parameter is used. The small approximation parameter causes an insignificant change in both objective function value and estimated variables in each iteration. Hence, the converge rate of PA-APG is slow and the approach becomes impractical for high precision problem [1]. Adaptive proximal average based proximal gradient (APA-APG) method [1] extends PA-APG with a novel adaptive parameter tuning technique. APA-APG enjoys $O(\frac{1}{k})$ iteration complexity with lesser iteration for convergence than PA-APG due to the large change of function and the estimated variables in the beginning. However, its ultimate convergence also relies on the approximation parameter which is reduced in each iteration and is ultimately small. Hence, the convergence is also slow in the end. Some other Jacobi iteration approaches which can be used for $P > 1$ include the three operator splitting method [266], the primal-dual splitting method [278]–[284] such as the majorized alternating direction method of multipliers (ADMM) [285] and the linearized alternating direction method (LADM) [286], the

forward-backward splitting method [287] and Douglas Rachford splitting method [288], a primal splitting method such as the generalized forward-backward splitting method [289], fast proximity method [290].

A recently proposed Gauss-Seidel technique, GSOS, was a trial in accelerating the optimization over the Jacobi iteration techniques [7]. However, it lost the availability of parallel computing acceleration which can be applied in estimating the individual proximal mappings of all nonsmooth terms in each Jacobi iteration of proximal average based methods. However, it is a good technique for large problems as it does not need many copies of matrix and vector values to be saved in the storage for parallel computing.

2.1.5. Curse of Dimensionality

High dimensionality is a curse in many domains including sparse representation[291]–[296]. In sparse representation, the causes of the curse can be due to an enormous amount of training data [295], [296], high-dimensional feature space [292], [293], and the size of the dictionary [294]. These problems can be relieved by applying data reduction techniques; namely, sample reduction and dimension reduction [297]. Sample reduction solves the enormous amount of data problem by selecting only representation samples to be the training sets [298]–[300]. Dimension reduction such as feature engineering [301]–[313], feature learning [33], [314]–[318], and feature selection [319]–[326] can transform the original features from high to low-dimensional space. Feature engineering is a process of feature generation and selection. It can be manual or automated. Manual feature generation relies on the knowledge of experts while automated feature generation relies on the feature exploration algorithm. Feature generation can involve feature crafting, extraction, fine-tuning, and transformation. Feature selection involves identifying and selecting representative and relevant features from an original feature set. Feature learning learns the representative and relevant features from the data through algorithms in a supervised, semi-supervised, or unsupervised manner. Sample reductions can be applied to reduce the sizes of sample dictionaries as well as dimension reductions that can be used to reduce the sizes of feature-based dictionaries in which the columns and rows of the dictionaries representing training samples or features. Meanwhile, dictionary learning techniques of feature learning

can be used to generate objective-specific dictionaries with arbitrary sizes [151], [327]–[335].

2.2. Automated Visual Inspection (AVI) and Its Techniques

Visual inspection (VI) is a category of inspection defined as follows:

- “Visual inspection is defined as the process of using the unaided eye, alone or in conjunction with various aids, as the sensing mechanism from which judgments may be made about the condition of a unit to be inspected.” [336], and
- “... Visual and optical tests are those that use probing energy from the visible portion of the electromagnetic spectrum. Changes in the light properties after contact with the test object may be detected by human or machine vision. Detection may be enhanced or made possible by mirrors, borescopes, or other vision-enhancing accessories.” [336]

VI is used to identify anomalies (defects, is a shortcoming, imperfection, or lack [337]) which deviate from expected product’s specification [338] and exceed the tolerance of deviation acceptance on inferior products with visible visual information [339]. It is one of the widely used and important inspection techniques in production or maintenance. “Over 80 per cent of inspections on large transport category aircraft are visual inspections. ” is claimed in a report published by Federal Aviation Administration (FAA) in 1997 [340]. Failure of inspection can be disastrous (i.e. Aircraft crash caused by engine breakdown [341]). Moreover, proper visual inspection procedures can improve the business process with cost and time in preventing redundant post-processing of inferior products at (subsequent) stages. It also reduces the inferior product which can adversely affect the reputation of the product in the market. The inspection also plays an important role in repairing/reworking. Detailed inspection can provide information that helps in decision making either in the type of repairing/reworking process or whether to dispose of the inferior product [342], [343].

AVI is a sub-category of visual inspection which aims to complement and improve on human visual inspection. It can eliminate inspection failure due to human factors. In studies, human inspection is highly affected by human factors (such as subjectivity,

mentality, and experience [336], [344]), environment [344], etc. It causes inconsistencies in inspection results and varying performance among different inspectors [345]. Hence, AVI is a potentially objective alternative support to human visual inspection.

Visual inspection problem can be broadly defined by the following factors:

- the complexity of the surface's characteristics (shape, colours, and materials),
- viewpoints (relative direction between the camera and the inspected object),
- illumination conditions, and
- characteristics of defects.

The details of each factor are shown in Figure 2.3. The surface characteristics can vary from simple to complex (i.e. the shape of the surface can be plain, have many edges and curvatures or/and contain multiple components in which materials and colours are different, etc.). The viewpoints can vary from one viewpoint to multiple discrete viewpoints as well as continuous viewpoints (video/image sequence). Moreover, the illumination can vary from one to multiple lighting sources. Furthermore, the lighting direction can be either independent of viewpoints or dependent on the surface material's reflection characteristics which may be either isotropic or anisotropic. Lastly, the defect's characteristics can be diverse (i.e. the difference between defects of the same type such as shape, appearance or size can be diverse [5]).

2.2.1. Surface Defects

The characteristics of surfaces as well as surface defects can be very diverse. The problem of surface defects in many industries includes manufacturing (such as textile, random texture product, electronic device and aircraft engine's turbine blade, etc.), construction, fruit and vegetable industries, etc.

The surface defect is the imperfection that does not fit the expected normal outcome and exceeds the toleration of the specification of the product/components. Figure 2.4 shows four examples of various surface defects [346]–[349]. The causes of surface defects can be diverse. It can be caused by an abnormal material formation which deviates from the

expected outcome due to imprecise environment control, external factors such as damage caused by assembly tools, etc. Fabric, textile, metallic, food, and beverage, products, etc. are commonly inspected by AVI systems.

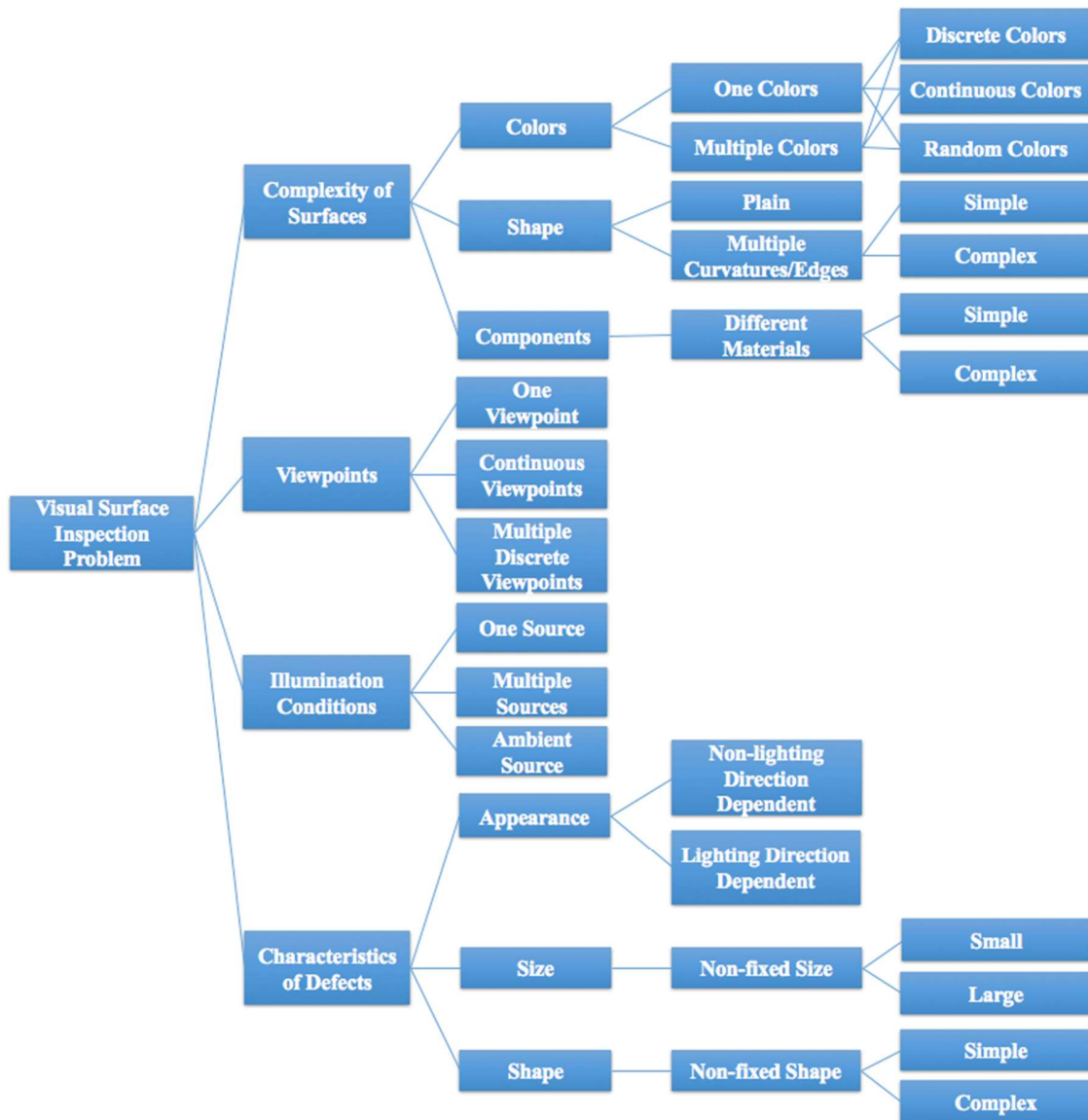


Figure 2.3 Factors of visual inspection problem

The characteristics of various surface defects are presented in Appendix A.1.

2.2.2. AVI Techniques

AVI techniques can be categorized based on the characteristics of AVI problems that they are adequate for. The characteristics of the AVI problem is generally determined by four factors:

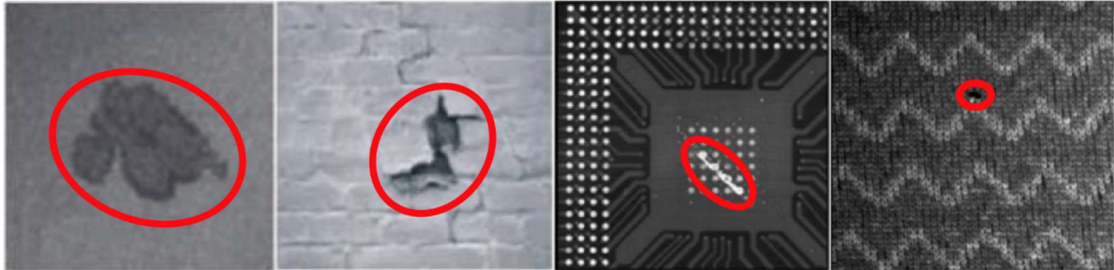


Figure 2.4 Examples of surface defect: (left to right) metallic defect [350], textural defect [347], PCB defect [348] and fabric defect [349]

- complexity of surface's characteristics (shape, colors and materials),
- illumination conditions,
- viewpoints, and
- characteristics of defects.

Surface's characteristics include:

- uniform surface
- uniform random texture surface
- patterned textured surface
- non-uniform and non-complex surface, and
- complex surface.

while lighting conditions can be uniform or non-uniform.

Figure 2.5 shows the general flow of automated visual inspection. It consists of:

- image acquisition
- preprocessing of acquired images,
- feature extraction from preprocessed images,
- region of interest (ROI, defect/anomaly) detection and/or localization based on the information of the features extracted, and

- possibly ROI (defect/anomaly) classification with another set of features extracted at ROI.

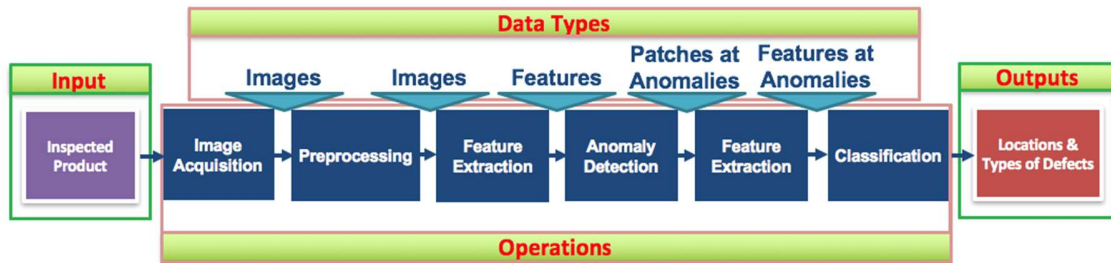


Figure 2.5 General flow of the automated visual inspection

The table presents a summary of various state-of-the-art AVI techniques published by the research community in these 2 decades. The table shows the number of the class to be classified, materials to be inspected, and the publication year. More details and elaboration of the proposed techniques such as classifiers, features, and names of the inspected defect classes, empirical results, inspected surface characteristics (shape, colours, and materials), illumination condition used and inspection types of the techniques can be found in Section A.2 in Appendix A.

Each technique is suited for specific inspection problems, specific surface characteristics, specific geometry, specific material, and/or specific expected characteristics of anomalies. Moreover, only one technique can be used to inspect complex surfaces under challenging non-uniform illumination. However, the technique utilizes the characteristics of specific expected anomalies and the specific normal surface which are different from the other inspection problems.

Table 2.5 Published AVI Techniques

Techniques	Number of Class	Surface Materials	Published Year	Reference
O-TM	2	Photomask	2001	[351]
Z-GA	2	Bumpy aluminium	2002	[352]
K-KNN	14	Paper	2006	[353]
X-TM	2	Ceramic tiles	2006	[354]
N-M	2	Patterned textured plain surface item	2008, 2010	[355], [356]
S-I	2	Steel	2008	[357]

J-OSVM	2	Tile	2009	[358]
A-PHOT	2	Any kind of textured surface item	2010	[359]
R-C4.5	3	Powder-coated metal	2011	[360]
A-WHC	2	Patterned textured fabric	2011	[361]
Z-SVM	2 (one-against-one)	Strongly reflected copper strip	2011	[362]
M-MPCNN	7	Steel Strip	2012	[363]
S-TM	2	Printed circuit board (PCB) assembly	2012	[364]
A-WHC-2	2	Patterned textured fabric	2012	[365]
B-HMMPP	2	PCB assembly	2013	[366]
W-TM	2	Integrated circuit (IC)	2013	[348]
R-SW	2	Patterned textured fabric	2013	[349]
G-SVM	2 (defect and background)	Steel Strip	2013	[367]
R-T	2	Any kind of textured surface item	2014	[347]
S-FL	2	Patterned textured fabric	2014	[368]
G-S	2	Random textured fabric	2014	[369]
L-FDCT	2 (defect and non-defect)	Ferrite magnetic tile	2014	[370]
Y-SVM	2	Steel billets	2014	[371]
K-FPM	2	Wafer	2014	[372]
C-FFANN	4	Extruded aluminum	2015	[373]
T-MS-SSIM	2	Polymer film	2015	[374]
L-T	2	Steel strip	2015	[375]
C-TWSVM	4	Steel strip	2015	[350]
G-ELM	2 (defect and background)	Steel Strip	2015	[346]
G-T	2	Steel strip	2015	[376]
P-RF	2	Display panel module	2016	[377]
L-DCDR	2	Steel bridge	2016	[378]
Z-PSVM	2	Asphalt pavements	2016	[379]
J-SL	2	Steel-plate	2016	[380]
J-PW	2	Ball grid array (BGA) on PCBs	2017	[381]
K-TM	2 (defect and non-defect)	Planar surfaces of industrial products	2017	[382]
C-SR	2	Nanofibrous products	2017	[383]
R-T-2	2	Railway-track	2017	[384]
C-SOMO	2	Concrete bridge	2017	[385]
D-SDDP	2	Subway networks	2017	[386]
O-FWT	2	Asphalt pavements	2017	[387]
R-DL	2	Hot-rolled steel strip and wood	2018	[388]

B-T	2	BGA on PCBs	2018	[389]
C-NB-CNN	2	Nuclear power plant component	2018	[390]
Y-DCNN	2	Concrete	2018	[391]
W-TM-2	2	Steel strip	2018	[392]
M-DL	5	Metallic gearboxes	2018	[393]
G-BODI	2	Rail	2018	[394]
L-SVM	2	Rail	2018	[395]
O-FL	2	Glass	2018	[396]
H-ROI	2 (defect and background)	Sewer pipelines	2018	[397]
K-DL	2 (defect and background)	Sewer pipelines	2018	[398]
M-RF	2	Sewer pipelines	2018	[399]
S-DT	2	Thin-walled metal canister	2019	[400]
E-MR-ELM	2	Patterned textured fabric	2019	[401]
C-FCN	2	Concrete	2019	[402]
W-R-CNN	2 (two damage categories)	Historic masonry structure	2019	[403]
C-DL	2	Steel bridge	2019	[404]

2.2.3. Challenges and Issues

Although AVI is seen as a potential replacement of human inspectors due to its advantages in speed and consistency; however, there exist many issues in the AVI tasks as well. These issues are due to the difficulty of designing artificial intelligent algorithms which either duplicate human intelligence during the inspection or are more intelligent than human in the inspection.

These difficulties are generally due to the complexity of the characteristics of the inspected product (or item). For examples, products' material, the geometrical shape of the product, the characteristics of each inspected defect class, and the availability and the quality of data. Problems with complex products can degrade the level of information for unacceptable defects due to the difficulty of identifying it from the diverse redundant information. To detect and identify complex defects, it may need extra information such as multiple images acquired under multiple illumination conditions, etc. Other than that, more than one defect in an image could cause computational algorithms to hardly learn the diverse information.

Lack of defect training samples and mislabeling of data also poses a challenge to the robustness of the algorithms.

2.3. Summary

This chapter reviews several sparse representation related topics; namely:

- the origin of sparse representation,
- categorization of sparse representation techniques,
- the properties of different norms and their minimizations,
- sparse reconstruction with L_1 -minimization,
- sparse reconstruction for face recognition with L_1 -minimization, and
- theoretical sparse representation recovery probabilistic bound with L_1 -minimization.

Furthermore, it reviews several proposed works related topic in problem design and optimization techniques; namely:

- nonsmooth composite convex optimization
- dimension reduction, and
- anomaly detection.

Lastly, an application domain, automated visual inspection (AVI), is reviewed.

Many parallelizable Jacobi Iteration techniques for nonsmooth composite convex optimization have been proposed. However, all of them use at most first-order information in the optimization. Integrating second-order information to accelerate the convergence and also to improve the optimization results is desired.

Theoretical bound for sparse representation and error recovery with L_1 -norm minimization in eqn. (1.1) and the assumption that dictionary A fulfills a bouquet model[147] has been studied. However, the bound only can be used for the problems which follow the dictionary assumption such as face recognition. The bound for those problems which do not fulfils the assumption is desirable. Moreover, the extension of bound for other L_1 -norm minimizations which are better choices than the one in eqn. (1.1) for the other problems is also desired.

In dictionary-based approaches such as sparse representation problem involves dictionary training. Reuse of the trained dictionaries frequently happened. However, a large dictionary that consumes much memory storage cannot be reused directly in a computational unit that has low memory capacity. Moreover, the convergence for a large-scale problem with high dimensional dictionaries is slow for those computation units which has very low computational power. Hence, a dimension reduction technique that forms a new lower-dimensional dictionary without retraining the dictionary is desirable. The optimization results obtained using the new low dimensional dictionary need to be highly similar to the results obtained using the original high dimensional dictionary.

L_1 -norm minimizations such as the one in eqns (1.1) and (1.2) are useful in some sparse coefficient vector and error recovery problems. It can be used as a template-based approach for recovering the noiseless original data and the corrupting error from a corrupted input/observed data. In this case, the error in the L_1 -norm minimization is caused by the anomalies and the locations of these anomalies can be detected from the information in the error vector while the dictionary contains all the templates. Hence, the anomaly detection approach which utilizes L_1 -norm minimization can be explored for the template-based anomaly detection problem. Further extension of the newly explored anomaly detection approach for practical problems such as AVI is highly desired. The approach is particularly useful for those difficult AVI problems which involve complex surface and non-uniform illumination. In the review, only one AVI approach is suitable for the difficult AVI problems; however, it can only be used for a specific problem in which the normal surface contains the specific shape information described in that work.

Chapter 3

A Novel Quasi-Newton Technique for Composite Convex Minimization

“A smooth sea never made a skilled sailor.”

- Franklin D. Roosevelt

This chapter presents a parallel approach that focuses on but is not limited to an optimization technique issue; namely: nonsmooth composite convex optimization, of sparse representation problem by improving the performance from the state-of-the-art techniques, please refer to Figure 1.1 and Figure 1.2. A fast parallelable Jacobi iteration type optimization technique for convex composite which contains multiple nonsmooth convex functions is presented. The method consists of a combination of second-order quasi-newton method, multiple first-order methods and parallelable accelerated proximal average and backtracking techniques. The first-order methods include the steepest gradient descent for L_2 and L_1 -norms and Nesterov’s accelerated gradient descent methods. The proposed method has a convergence rate with a lower bound of $O\left(\frac{1}{k^2}\right)$ and is able to enjoy superlinear convergence when proper second-order information is available. Unlike other works that only use first-order information of proximal average; here, the proposed technique uses both first and second-order information of proximal average. The extensive first and second-order information of proximal averages with different smoothing parameters provide more information about good descent direction. In the experiment, a problem in which proximal mapping subproblems is more computationally expensive than the examples given in

the previous works [1], [277] is used. The problem illustrates the potential of the proposed method in large scale problem such as those problems with high-resolution images. Note that the method is easily adaptable to suit those problems without computational issues. The experimental results show that the proposed method converges significantly superior to the state of art methods (e.g. Adaptive proximal average based proximal gradient, APA-APG1 and APA-APG2, methods [1] which enjoy $O\left(\frac{1}{k}\right)$ convergence) in all cases of the examples.

In this chapter, the techniques integrated into the proposed method and their benefits are introduced in Section 3.1. In Section 3.2, the proximal average function and its proximal mapping and gradient are introduced. Section 3.3 shows how quasi-Newton can be applied to the objective function of proximal mappings. The proposed method and algorithms are introduced in Section 3.4. In Section 3.5, the convergence analyses of each first and second-order methods, which are integrated into the proposed method, are presented. Section 3.6 presents the experiments for comparing the performance of the proposed algorithms and the state-of-the-art algorithms on a composite convex optimization problem and Section 3.7 concludes this chapter.

3.1. Integrated Techniques and the Benefits

All previous parallelable Jacobi iteration type methods are at most first-order type methods which only utilize first-order information in their optimizations. This chapter proposes a novel method which combines second-order and first-order approaches to accelerate the optimization problem (2.19) as follows.

$$F^* = \min_{x \in \mathbb{X}} F(x) = \min_{x \in \mathbb{X}} G(x) + H(x) \quad \text{Recap (2.19)}$$

where

$\mathbb{X}$	is a linear and finite-dimensional space,
$G(x)$	is a smooth convex composite function and is defined as $G(x) = \sum_{i=1}^O \alpha_i g_i(x)$ in which $\alpha_i \geq 0$ and O is the total numbers of functions g_i ,
$g_i: \mathbb{X} \rightarrow (-\infty, \infty)$	is a smooth continuously differentiable convex function,
$H(x)$	is a nonsmooth convex composite function and is defined as $H(x) = \sum_{i=O+1}^{O+P} \alpha_i h_i(x)$ in which $\alpha_i \geq 0$ and P is the total numbers of functions h_i ,
$h_i: \mathbb{X} \rightarrow (-\infty, \infty)$	is a nonsmooth convex function, and
α_i	are coefficients which fulfill $\alpha_i \geq 0$ and $\sum_i \alpha_i = 1$.

Recall the problem in eqn. (2.19) described in Chapter 2, it is a multi-term convex optimization problem which contains nonsmooth composite.

When the proposed method is applied, the computations which are associated with both the smooth and nonsmooth convex functions, $g_i(x)$ and $h_i(x)$, in the optimization problem (2.19) are treated in the same manner. For simplicity, the problem in eqn. (3.2) is used to represent the optimization problem (2.19) in the content.

$$F^* = \min_{x \in \mathbb{X}} F(x) = \min_{x \in \mathbb{X}} \sum_{i=1}^N \alpha_i f_i(x) \quad (3.1)$$

where $f_i(x)$ can be either smooth or non-smooth convex function and $N = O + P$.

Unlike the previous works, the approach utilizes both first and second-order information. The state of art second-order method, Broyden-Fletcher-Goldfarb-Shanno algorithm (BFGS quasi-Newton method), [405]–[408] is used. The first-order methods which are used in the proposed method include steepest gradient descent for L_2 and L_1 -norms as well as Nesterov's accelerated gradient descent. Moreover, parallel accelerable techniques, backtracking inexact line search and proximal average approximation, are also used. The method has a convergence rate with a lower bound of $O\left(\frac{1}{k^2}\right)$ and has superlinear convergence when proper second-order information is available. The proposed methods do

not suffer from the slowing down of convergence which has been faced by numerous works when a small approximation parameter of proximal average is used [1], [8].

3.2. Proximal Average Function and Its Proximal Mapping and Gradient

In this section, the definition of proximal function and its mapping and gradient are first introduced. After that, the proximal average function and its gradient which are the keys information to be used in the proposed optimization method (is introduced in Sections 3.3 and 3.4) are presented. Furthermore, the relationship between proximal mapping and proximal average mapping are elaborated.

Proximal mapping of a function $f(x)$ is defined by eqn. (3.2)

$$Prox_f^\gamma(x) = \underset{u \in \mathbb{X}}{\operatorname{argmin}} f(u) + \frac{1}{2\gamma} \|u - x\|^2 \quad (3.2)$$

and its corresponding minimization problem is given by eqn. (3.3)

$$f_\gamma(x) = \min_{u \in \mathbb{X}} f(u) + \frac{1}{2\gamma} \|u - x\|^2 \quad (3.3)$$

where

$\mathbb{X}$ is a finite-dimensional linear space,

$x, u \in \mathbb{X}$,

γ is a parameter, and

$f_\gamma(x)$ is a convex and smooth function which shares the same minimizer of $f(x)$.

It always represents the smooth version of $f(x)$ which may be a nonsmooth function.

Fixed point theory shows that minimizer x^* of $f(x)$ is also the minimizer of $f_\gamma(x^*)$; hence, $Prox_f^\gamma(x^*) = x^*$ [409].

The gradient of $f_\gamma(x)$ is described by eqn. (3.4).

$$\nabla f_\gamma(x) = \frac{1}{\gamma} (x - Prox_f^\gamma(x)) \quad (3.4)$$

where $\nabla f_\gamma(x)$ is Lipchitz continuous with Lipchitz constant at most of γ^{-1} .

Note that $f_\gamma(x)$ and $\nabla f_\gamma(x)$ are the keys to this chapter. $f_\gamma(x)$ and $\nabla f_\gamma(x)$ play a critical role in the optimization of $f(x)$ as they represent unlimited objective functions (with different γ) and shares the same minimizers with $f(x)$. Each gradient of various $f_\gamma(x)$ provides the information on the descent direction for minimizing $f(x)$ and hence provides abundant information when various γ is used. Moreover, these unlimited $\nabla f_\gamma(x)$ also provided the unlimited $\nabla^2 f_\gamma(x)$ information when it is used with quasi-Newton methods. With our proposed methods, this information can potentially lead to a superior optimization convergence in contrast with other methods such as the first-order methods.

From (3.3), it is easy to derive that $\lim_{\gamma \rightarrow 0} F_\gamma(x) = F(x)$ in which $F(x)$ is defined in eqn. (2.19). The equality implies that $\lim_{\gamma \rightarrow 0} \nabla F_\gamma(x) = \nabla F(x)$ and $\lim_{\gamma \rightarrow 0} \nabla^2 F_\gamma(x) = \nabla^2 F(x)$.

Note that $\nabla F_\gamma(x)$ and $\nabla^2 F_\gamma(x)$ are not equal to $\nabla F(x)$ and $\nabla^2 F(x)$ respectively when γ is not equal or approaching to 0; however, they provide extra information on the descent direction especially when $\lim_{\gamma \rightarrow 0} \nabla F_\gamma(x)$ and $\lim_{\gamma \rightarrow 0} \nabla^2 F_\gamma(x)$ are not useful. E.g. when $F(x) = \|x\|_1$, $\lim_{\gamma \rightarrow 0} \nabla^2 F_\gamma(x) = 0$.

The following presents the definition of the proximal average function of a composite function. Its proximal mapping contains a key property which is critical to the proposed methods and making unsolvable or complex proximal mapping solvable with sums of simpler proximal mappings.

Definition 1 (Reference [410], **Definition 4.1**)

The proximal average function of $F(x) = \sum_{i=1}^N \alpha_i f_i(x)$ with parameter $\gamma > 0$ is defined via the following optimization problem described in eqn. (3.5).

$$\tilde{F}_\gamma(x) := \inf_{\sum_{i=1}^N \alpha_i u_i = x} \left\{ \sum_{i=1}^N \alpha_i f_i(u_i) + \frac{1}{2\gamma} \sum_{i=1}^N \alpha_i \|u_i\|^2 - \frac{1}{2\gamma} \|x\|^2 \right\} \quad (3.5)$$

where

$f_i(u_i)$ is either a smooth or a non-smooth term,
 α_i is the coefficient of $f_i(u_i)$,
 N is the total number of $f_i(y_i)$ in $F(x)$ which fulfils $F(x) = \sum_{i=1}^N \alpha_i f_i(x)$, and
 γ is a parameter.

The following shows the key relationship between proximal mappings of the individual function f_i and the proximal average function. The relationship enables the parallel acceleration of the computations in determining the proximal mappings of the proximal average function. It can dramatically increase the convergence speed, especially for those computationally expensive proximal mappings.

Proposition 1 ((Reference [410], Theorem 6.7))

$$\text{Prox}_{\tilde{F}_\gamma}^\gamma(x) = \sum_{i=1}^N \text{Prox}_{f_i}^\gamma(x) \quad (3.6)$$

where

γ is a parameter,
 $\tilde{F}_\gamma$ is the function $\tilde{F}_\gamma(x)$ defined in eqn. (3.5),
 f_i is the function $f_i(x)$,
 N is the total number of $f_i(y_i)$ in $F(x)$ which fulfils $F(x) = \sum_{i=1}^N \alpha_i f_i(x)$, and
 $\text{Prox}_{f_i}^\gamma(x)$ is the function $\text{Prox}_f^\gamma(x)$ defined in eqn. (3.2) where $f(x) = f_i(x)$,

and the gradient of the minimization problem described in eqn. (3.7) is defined by eqn. (3.8).

$$\bar{F}_\gamma(x) = \min_{u \in \mathbb{X}} \tilde{F}_\gamma(u) + \frac{1}{2\gamma} \|u - x\|^2 \quad (3.7)$$

$$\nabla \bar{F}_\gamma(x) = \frac{1}{\gamma} \left(\text{Prox}_{\tilde{F}_\gamma}^\gamma(x) - x \right) := \frac{1}{\gamma} \left(\sum_{i=1}^N \text{Prox}_{f_i}^\gamma(x) - x \right) \quad (3.8)$$

Eqn. (3.8) allows one to compute $\nabla \bar{F}_\gamma(x)$ with simpler proximal mapping composite $\sum_{i=1}^N \text{Prox}_{f_i}^\gamma(x)$ instead of using its possibly much more complex proximal mapping

$Prox_{\bar{F}_\gamma}^\gamma(x)$. It also allows us to use the unlimited information of $\nabla \bar{F}_\gamma(x)$ and $\nabla^2 \bar{F}_\gamma(x)$ with various γ to get the minimizers of $\bar{F}_\gamma(x)$.

Note that when **Theorem 8.5** in [410], $\lim_{\gamma \rightarrow 0} \tilde{F}_\gamma(x) = F(x)$ and the fact that $\lim_{\gamma \rightarrow 0} \bar{F}_\gamma(x) = \tilde{F}_\gamma(x)$ are applied, $\lim_{\gamma \rightarrow 0} \bar{F}_\gamma(x) = F(x)$ is achieved. It also implies that $\nabla \tilde{F}_\gamma(x)$ and $\nabla^2 \tilde{F}_\gamma(x)$ are first and second order derivatives of the smooth version of $F(x)$ when γ is near to zero.

The only issue here is whether the minimizers of $F(x)$ and $\bar{F}_\gamma(x)$ are the same. If they are not, how much difference would exist between them and how this difference affects the optimization convergence? This issue will be elaborated in Section 3.5.

In the following section, how the quasi-Newton method which allows us to enjoy the superlinear convergence complexity with proximal mappings is elaborated.

3.3. Quasi-Newton Technique with Proximal Mappings

As the gradient $\nabla \bar{F}_\gamma(x)$ (described in eqn. (3.8)) can be determined, first and second-order techniques such as quasi-Newton, steepest gradient descent and accelerated gradient descent techniques can be applied. However, all previous works only consider techniques which only use the first-order information. In this section, how this gradient can be used to approximate the Hessian matrix and its inverse and hence second-order method can be used is elaborated. Also, how this gradient is linked to the approximation of the Hessian matrix and its inverse is described. Note that the state-of-the-art Broyden-Fletcher-Goldfarb-Shanno algorithm (BFGS quasi-Newton technique) is used in the elaboration and it can be replaced by L-BFGS (a limited-memory version of BFGS), BFGS's low memory extension when memory issues are of concern.

A Newton step, Δx , is defined as a step to move toward the location where the gradient is zero from the current point based on the second-order Taylor series approximation of the objective function at that point.

The Newton step at optimization iteration k (for minimization problem of $\bar{F}_\gamma$) is defined as by eqn. (3.9),

$$\Delta x_k = -B_k^{-1} \nabla \bar{F}_\gamma(x_k) \quad (3.9)$$

where B_k^{-1} is the approximation of the Hessian matrix at the point x_k . B_0 can be initiated with the identity matrix or it should be initiated with positive definite matrix (it guarantees that the Newton step is in the descent direction when x_k is far from minimum and enjoy quadratic convergence of Newton's method when x_k is near to the minimum [411]).

When the inexact line search is used, the Newton step can be redefined as follows in eqn. (3.10),

$$\Delta x_k = -\alpha'_k B_k^{-1} \nabla \bar{F}_\gamma(x_k) \quad (3.10)$$

where α'_k is the step size.

B_k can be approximated using eqn. (3.11),

$$B_k = B_{k-1} + \frac{s_{k-1} s_{k-1}^T}{s_{k-1}^T \Delta x_{k-1}} - \frac{B_{k-1} \Delta x_{k-1} (B_{k-1} \Delta x_{k-1})^T}{\Delta x_{k-1}^T B_{k-1} \Delta x_{k-1}} \quad (3.11)$$

where $s_{k-1} = \nabla \bar{F}_\gamma(x_k) - \nabla \bar{F}_\gamma(x_{k-1})$

A good property of Hessian matrix approximation with BFGS is that B_k is always positive definite if B_{k-1} is positive definite and α_{k-1} is chosen by applying Wolfe conditions.

Note that another BFGS technique which estimates the inverse of B_k instead of B_k with Sherman-Morrison Formula [412]–[414] is not used as eqn. (3.11) gives us the extra information about whether B_k is nearly singular or singular and the B_k approximation is needed to be restarted with $B_k = B_0$.

3.4. Proposed Method and Algorithms with Combination of Gradient Descent and Quasi-Newton Techniques

Note that the method is proposed with the motivation of fully utilizing the first and second-order information (of the objective function of the minimization of the proximal average

function) before recalculating its proximal mappings. Hence, both gradient descent and quasi-Newton techniques are applied to discover useful descent direction in the next step with backtracking-based inexact line search before the next iteration. The intention of using the combinational technique instead of only using the quasi-Newton technique is that the descent directions computed by first-order techniques are sometimes more useful when the second-order information is unreliable (approximated Hessian matrix is nearly singular or singular) or unavailable (the second derivative is zero). The combinational technique chooses the best next step (which has the lowest value of the objective function) based on the descent directions derived from all first and second-order techniques. Hence, this method also guarantees that the convergence rate will be better than the methods which only use any descent direction of any first or second-order technique. In the proposed combinational technique,

- a second-order technique, a BFGS quasi-newton technique, and
- three first-order techniques, steepest descents for L_1 and L_2 -norms and Nesterov's accelerated gradient descent techniques

are used.

Algorithms B.1 (with **Function B.1**) and **B.2** (with **Function B.2**) which are in Appendix B show the algorithms with the proposed proximal average quasi-Newton methods (PA-QN) with and without using Wolfe conditions in choosing the step size respectively.

PA-QN with Wolfe conditions (**Algorithm B.1**) makes sure that each objective function and gradient is sufficiently decremented in each iteration. **Lemma 1** is described as follows, the second Wolfe condition is almost always held if $m_2 \approx 1$. Therefore, the validation of the second Wolfe condition is almost redundant and can always be ignored to save on computation time when $m_2 \approx 1$ is used. This can be highly computationally expensive if backtracking is used in the inexact line search. Hence, it is sufficient to work only with the first Wolfe condition (Armijo rule) so that it does not need to compute next point's gradient which has to be computed with the proximal mappings in the next iteration to determine whether the second condition is fulfilled. The algorithm searches in multiple potential descent directions until the condition is fulfilled.

Lemma 1

$$-p_k^T \nabla \bar{F}_Y(x_{k+1}) \leq -m_2 p_k^T \nabla \bar{F}_Y(x_k) \quad (3.12)$$

where

p_k is the descent direction,

$\nabla \bar{F}_Y(x)$ is defined in eqn. (3.8), and

m_2 is a parameter,

almost always held when $m_2 \rightarrow 1$.

Note that eqn. (3.12) is a curvature condition (the second condition of Wolfe conditions) which ensure the slope $\nabla \bar{F}_Y(x_{k+1})$ will be at least m_2 times larger than the slope $\nabla \bar{F}_Y(x_k)$.

Proof:

The non-expansiveness of proximal mapping [409] states that (see eqn. (3.13)) the distance between $Prox_{\bar{F}_Y}^\gamma(x_{k+1})$ and $Prox_{\bar{F}_Y}^\gamma(x_k)$ (left-hand side of eqn. (3.13)) is less than or equal to the inner product of $(x_{k+1} - x_k)$ and $(Prox_{\bar{F}_Y}^\gamma(x_{k+1}) - Prox_{\bar{F}_Y}^\gamma(x_k))$ (right-hand side of eqn. (3.13)).

$$\|Prox_{\bar{F}_Y}^\gamma(x_{k+1}) - Prox_{\bar{F}_Y}^\gamma(x_k)\|^2 \leq (x_{k+1} - x_k)^T (Prox_{\bar{F}_Y}^\gamma(x_{k+1}) - Prox_{\bar{F}_Y}^\gamma(x_k)) \quad (3.13)$$

Cauchy-Schwarz inequality [415] states that (see eqn. (3.14)) the norm of the inner product of $(x_{k+1} - x_k)$ and $(Prox_{\bar{F}_Y}^\gamma(x_{k+1}) - Prox_{\bar{F}_Y}^\gamma(x_k))$ (left-hand side of eqn. (3.14)) is less than the product of the norm of $(Prox_{\bar{F}_Y}^\gamma(x_{k+1}) - Prox_{\bar{F}_Y}^\gamma(x_k))$ and the norm of $(x_{k+1} - x_k)$ (right-hand side of eqn. (3.14)).

$$\begin{aligned} & \left\| (x_{k+1} - x_k)^T (Prox_{\bar{F}_Y}^\gamma(x_{k+1}) - Prox_{\bar{F}_Y}^\gamma(x_k)) \right\| \\ & \leq \|Prox_{\bar{F}_Y}^\gamma(x_{k+1}) - Prox_{\bar{F}_Y}^\gamma(x_k)\| \|x_{k+1} - x_k\| \end{aligned} \quad (3.14)$$

Combining eqns (3.13) and (3.14), the following can be derived; see eqn. (3.15).

$$\left\| \text{Prox}_{\tilde{F}_Y}^\gamma(x_{k+1}) - \text{Prox}_{\tilde{F}_Y}^\gamma(x_k) \right\| \leq \|x_{k+1} - x_k\| \quad (3.15)$$

From eqn. (3.15), the following inequality in eqn. (3.16) holds.

$$\begin{aligned} \|x_{k+1} - x_k\|^2 - (x_{k+1} - x_k)^T \left(\text{Prox}_{\tilde{F}_Y}^\gamma(x_{k+1}) - \text{Prox}_{\tilde{F}_Y}^\gamma(x_k) \right) \\ \geq 0 \end{aligned} \quad (3.16)$$

From eqn. (3.16),

$$\begin{aligned} (x_{k+1} - x_k)^T \left[(x_{k+1} - x_k) - \left(\text{Prox}_{\tilde{F}_Y}^\gamma(x_{k+1}) - \text{Prox}_{\tilde{F}_Y}^\gamma(x_k) \right) \right] &\geq 0 \\ (x_{k+1} - x_k)^T \left[\left(x_{k+1} - \text{Prox}_{\tilde{F}_Y}^\gamma(x_{k+1}) \right) - \left(x_k - \text{Prox}_{\tilde{F}_Y}^\gamma(x_k) \right) \right] &\geq 0 \\ p_k^T \left(\nabla \bar{F}_Y(x_{k+1}) - \nabla \bar{F}_Y(x_k) \right) &\geq 0 \\ -p_k^T \nabla \bar{F}_Y(x_{k+1}) &\leq -p_k^T \nabla \bar{F}_Y(x_k) \\ -p_k^T \nabla \bar{F}_Y(x_{k+1}) &\leq -m_2 p_k^T \nabla \bar{F}_Y(x_k) + (m_2 - 1) p_k^T \nabla \bar{F}_Y(x_k) \end{aligned} \quad (3.17)$$

As $\lim_{m_2 \rightarrow 1} (m_2 - 1) p_k^T (\nabla \bar{F}_Y(x_k)) = 0$; hence, **Lemma 1** is proven that (3.12) is almost always held when $m_2 \rightarrow 1$.

Accelerated backtracking for inexact line search using parallel computing can be applied when one or more of individual proximal mappings are computationally expensive. Hence, Greedy PA-QN is likely to outperform PA-QN under the Wolfe conditions for high computation problem. It takes a small portion of time (e.g. 10% of the computation time of proximal mappings) to track as many points as possible to determine the point which decreases the objective function the most before the much more computationally expensive in the next iteration.

3.5. Convergence Analysis

In this section, the convergence bounds of the methods which are integrated into the proposed method; namely

- the second-order method, quasi-Newton method, and

- the other first-order methods,
 - steepest gradient descent for L_2 -norm,
 - steepest gradient descent for L_1 -norm, and
 - Nesterov's accelerated gradient descent method

are analyzed. The final convergence bound can be easily derived by taking the lowest bounds of each method.

Before the analysis starts, an equation and inequality in (Reference [1], **Lemma 1**) are highlighted. They are as follow:

Lemma 2

Suppose that

- $\alpha_i \geq 0$ satisfying $\sum_i \alpha_i = 1$, and
- $\tilde{F}_\gamma(x)$ is the proximal average of $F(x)$ with approximation parameter $\gamma > 0$,

Then the following statements in eqns (3.18) and (3.19) hold for any γ :

$$\tilde{F}_\gamma(x) = \sup_{u \in Q} \left\{ \sum_{i=1}^N \alpha_i \langle u_i, x \rangle - \sum_{i=1}^N f_i^*(u_i) - \frac{\gamma}{2} \|u\|_C^2 \right\} \quad (3.18)$$

$$\tilde{F}_\gamma(x) + \frac{\gamma}{2} \|u_\gamma^*(x)\|_C^2 \leq F(x) \leq \tilde{F}_\gamma(x) + \frac{\gamma}{2} \|u_0^*(x)\|_C^2 \quad (3.19)$$

where

$C = E - D^*D$ in which $D = [\alpha_1, \dots, \alpha_N]$, $E = \text{Diag}(\alpha_1, \dots, \alpha_N)$, $0 \preceq C \preceq I_N$ and

I_N is a $N \times N$ -dimensional identity matrix

$f_i^*(u_i) = \sup_{u_i \in Q_i} \{\langle u_i, x \rangle - f_i(u_i)\}$ in which $Q_i = \text{dom} g_i^*$,

Q is the Cartesian product of Q_i for $i = 1, \dots, N$ such that $Q = Q_1 \times \dots \times Q_N$,

$\|u\|_C = \sqrt{\langle u \cdot C \rangle}$, and

$u_\gamma^*(x) \in U_\gamma^*(x)$ in which $U_\gamma^*(x)$ denotes the set of (3.18) with parameter $\gamma > 0$.

They are used in the convergence analyses.

In the analysis, the following Lemma which shows the relationship between the objective function $\tilde{F}_\gamma(x)$ (see the definition in eqn.(3.7)) used in the proposed algorithms and the

original objective function $F(x)$. It plays a critical role in the following convergence analyses.

Lemma 3

Suppose that

- $\varepsilon = F(x) - F(x^*)$, and
- $\bar{\varepsilon}_\gamma = \bar{F}_\gamma(x) - \bar{F}_\gamma(x_\gamma^*)$,

then

- ε will be bound by $\bar{\varepsilon}_\gamma + \Delta\varepsilon(\gamma)$ (see eqn.(3.20)), and
- the limit of $\bar{\varepsilon}_\gamma$ as γ approaches 0 equals ε (see eqn. (3.31)).

$$\bar{\varepsilon}_\gamma + \Delta\varepsilon(\gamma) \geq \varepsilon \quad (3.20)$$

$$\lim_{\gamma \rightarrow 0} \bar{\varepsilon}_\gamma = \varepsilon \quad (3.21)$$

where

$$\begin{aligned} \Delta\varepsilon(\gamma) = D_\gamma + \frac{\gamma}{2} [2\nabla\bar{F}_{\gamma'}(x)\nabla\bar{F}_\gamma(x) - (1 + \gamma m_{\gamma'})\nabla\bar{F}_\gamma^2(x) \\ + \|u_0^*(x)\|_C^2 - \|u_\gamma^*(x)\|_C^2], \end{aligned} \quad (3.22)$$

x^* and x_γ^* are the minimizers $F(x)$ of $\bar{F}_\gamma(x)$ respectively,

$$D_\gamma = F(x_\gamma^*) - F(x^*),$$

$u_0^*(x) \in U_0^*(x)$, and

$m_{\gamma'} > 0$ is a constant.

Proof:

Based on the definitions of $Prox_f^\gamma(x)$ and $f_\gamma(x)$ in eqns (3.2) and (3.3), $Prox_{\bar{F}_\gamma}^\gamma(x)$ is the global minimizer of minimization $\bar{F}_\gamma(x)$ and $\bar{F}_\gamma(x)$ can be expressed as in eqn. (3.23).

$$\bar{F}_\gamma(x) = \tilde{F}_\gamma\left(Prox_{\bar{F}_\gamma}^\gamma(x)\right) + \frac{1}{2\gamma} \left\|Prox_{\bar{F}_\gamma}^\gamma(x) - x\right\|^2 \quad (3.23)$$

where

γ is a parameter,

$\tilde{F}_\gamma(x)$ is defined in eqn. (3.5), and

$Prox_{\bar{F}_\gamma}^\gamma(x)$ is defined in eqn. (3.6).

Using the inequality of $F(x) \leq \tilde{F}_\gamma(x) + \frac{\gamma}{2} \|u_0^*(x)\|_C^2$ (see eqn. (3.19)), the inequality in eqn. (3.24) can be derived.

$$\bar{F}_\gamma(x) \geq F\left(\text{Prox}_{\tilde{F}_\gamma}^\gamma(x)\right) - \frac{\gamma}{2} \left\| u_0^*\left(\text{Prox}_{\tilde{F}_\gamma}^\gamma(x)\right) \right\|_C^2 + \frac{1}{2\gamma} \left\| \text{Prox}_{\tilde{F}_\gamma}^\gamma(x) - x \right\|^2 \quad (3.24)$$

As $\lim_{\gamma \rightarrow 0} \bar{F}_\gamma(x) = F(x)$ stated in Section 3.2; hence, the equality in eqn. (3.25) is valid when a constant γ' which is near to zero and larger than 0 is introduced. Eqn. (3.25) implies that $\bar{F}_{\gamma'}(x)$ is the smoothed version of $F(x)$.

$$F\left(\text{Prox}_{\tilde{F}_\gamma}^\gamma(x)\right) = \bar{F}_{\gamma'}\left(\text{Prox}_{\tilde{F}_\gamma}^\gamma(x)\right) \quad (3.25)$$

By using the equality in eqn. (3.25) and the definition of proximal mapping in eqn. (3.4), the following inequality can be derived from (3.24),

$$\bar{F}_\gamma(x) \geq \bar{F}_{\gamma'}(x - \gamma \nabla \bar{F}_\gamma(x)) - \frac{\gamma}{2} (\|u_0^*(x)\|_C^2 - \nabla \bar{F}_\gamma^2(x)) \quad (3.26)$$

By assuming $\bar{F}_{\gamma'}(x)$ is strongly convex, there is a constant $m_{\gamma'} > 0$ such that the inequality in eqn. (3.27) is fulfilled.

$$\nabla^2 \bar{F}_{\gamma'}(x) \succeq m_{\gamma'} I_N \quad (3.27)$$

where I_N is a $N \times N$ -dimensional identity matrix.

Using both the inequalities in eqns (3.26) and (3.27), the following inequality (eqn. (3.28)) holds,

$$\bar{F}_{\gamma'}(x + \Delta x) \geq \bar{F}_{\gamma'}(x) + \Delta x \nabla \bar{F}_{\gamma'}(x) + \frac{1}{2} \Delta x^T m_{\gamma'} \Delta x \quad (3.28)$$

Applying the inequality in eqn. (3.28) on $\bar{F}_{\gamma'}(x - \gamma \nabla \bar{F}_\gamma(x))$ using $\Delta x = -\gamma \nabla \bar{F}_\gamma(x)$ in the inequality in eqn. (3.26) and the fact of the equality $F(x) = \bar{F}_{\gamma'}(x)$ implied by eqn. (3.25), the inequality in eqn. (3.29) is formed.

$$\begin{aligned}\bar{F}_\gamma(x) \geq F(x) - \gamma \nabla \bar{F}_{\gamma'}(x) \nabla \bar{F}_\gamma(x) + \frac{\gamma^2 m_{\gamma'}}{2} \nabla \bar{F}_\gamma^2(x) \\ - \frac{\gamma}{2} (\|u_0^*(x)\|_C^2 - \nabla \bar{F}_\gamma^2(x))\end{aligned}\quad (3.29)$$

Rearranging eqn. (3.29), the following inequality (eqn. (3.30)) is formed.

$$\bar{F}_\gamma(x) \geq F(x) - \frac{\gamma}{2} (2 \nabla \bar{F}_{\gamma'}(x) \nabla \bar{F}_\gamma(x) - (1 + \gamma m_{\gamma'}) \nabla \bar{F}_\gamma^2(x) + \|u_0^*(x)\|_C^2) \quad (3.30)$$

Minus $\bar{F}_\gamma(x_\gamma^*)$ at both sides, the following inequality (eqn. (3.31)) is formed.

$$\begin{aligned}\bar{F}_\gamma(x) - \bar{F}_\gamma(x_\gamma^*) \geq F(x) - \bar{F}_\gamma(x_\gamma^*) \\ - \frac{\gamma}{2} (2 \nabla \bar{F}_{\gamma'}(x) \nabla \bar{F}_\gamma(x) - (1 + \gamma m_{\gamma'}) \nabla \bar{F}_\gamma^2(x) + \|u_0^*(x)\|_C^2)\end{aligned}\quad (3.31)$$

where x_γ^* is the minimizer of $\bar{F}_\gamma(x)$.

By applying the inequality of $\tilde{F}_\gamma(x) + \frac{\gamma}{2} \|u_\gamma^*(x)\|_C^2 \leq F(x)$ (see eqn. (3.19)), the following inequality (eqn. (3.32)) can be derived from eqn. (3.31).

$$\begin{aligned}\bar{F}_\gamma(x) - \bar{F}_\gamma(x_\gamma^*) \geq F(x) - \left(F(x_\gamma^*) - \frac{\gamma}{2} \|u_\gamma^*(x)\|_C^2\right) \\ - \frac{\gamma}{2} (2 \nabla \bar{F}_{\gamma'}(x) \nabla \bar{F}_\gamma(x) - (1 + \gamma m_{\gamma'}) \nabla \bar{F}_\gamma^2(x) + \|u_0^*(x)\|_C^2)\end{aligned}\quad (3.32)$$

Note that the minimizer x_γ^* of $\tilde{F}_\gamma(x)$ is also the minimizer of $\bar{F}_\gamma(x)$ and; hence $Prox_{\tilde{F}_\gamma}^\gamma(x_\gamma^*) = x_\gamma^*$ based on the fixed point theory [409].

$F(x_\gamma^*) \geq F(x^*)$ and the following inequality (eqn. (3.33)) is formed where x_γ^* is the minimizers of both $\tilde{F}_\gamma(x)$ and $\bar{F}_\gamma(x)$, x^* is the minimizer of $F(x)$ and $D_\gamma = F(x_\gamma^*) - F(x^*)$.

$$\begin{aligned}\bar{F}_\gamma(x) - \bar{F}_\gamma(x_\gamma^*) \geq F(x) - F(x^*) - D_\gamma - \frac{\gamma}{2} [2 \nabla \bar{F}_{\gamma'}(x) \nabla \bar{F}_\gamma(x) \\ - (1 + \gamma m_{\gamma'}) \nabla \bar{F}_\gamma^2(x) + \|u_0^*(x)\|_C^2 - \|u_\gamma^*(x)\|_C^2]\end{aligned}\quad (3.33)$$

By definition of $\bar{\varepsilon}_\gamma$, ε and $\Delta \varepsilon(\gamma)$ in **Lemma 3**, the inequality in eqn. (3.20) can be formed

by rearranging the inequality in eqn. (3.33) but with

$$\varepsilon = F(x) - F(x^*),$$

$$\bar{\varepsilon}_\gamma = \bar{F}_\gamma(x) - \bar{F}_\gamma(x_\gamma^*), \text{ and}$$

$$\Delta\varepsilon(\gamma) = D_\gamma + \frac{\gamma}{2} \left[2\nabla\bar{F}_{\gamma'}(x)\nabla\bar{F}_\gamma(x) - (1 + \gamma m_{\gamma'})\nabla\bar{F}_\gamma^2(x) + \|u_0^*(x)\|_C^2 - \|u_\gamma^*(x)\|_C^2 \right]$$

The eqn. (3.20) in **Lemma 3** is proven.

It is easy to derive that $\lim_{\gamma \rightarrow 0} \bar{F}_\gamma(x) = \tilde{F}_\gamma(x)$ from the definitions of $\bar{F}_\gamma(x)$ and $\tilde{F}_\gamma(x)$ in eqns (3.5) and (3.7) respectively.

As **Theorem 8.5** in [410] states that $\lim_{\gamma \rightarrow 0} \tilde{F}_\gamma(x) = F(x)$ and; hence, $\lim_{\gamma \rightarrow 0} x_\gamma^* = x^*$.

Finally, $\lim_{\gamma \rightarrow 0} \bar{F}_\gamma(x) - \bar{F}_\gamma(x_\gamma^*) = F(x) - F(x^*)$ and; hence eqn. (3.34) is established.

$$\lim_{\gamma \rightarrow 0} \bar{\varepsilon}_\gamma = \varepsilon \tag{3.34}$$

The equality in eqn. (3.21) in **Lemma 3** is proven.

Note that D_γ is 0 when optimal sets of all $f_i(x)$ contain the minimizer of $F(x)$. It has been proven with **Corollaries 5.6** and **6.4** in [410].

Eqns (3.20) and (3.21) in **Lemma 3** are a critical component in the convergence analyses of the methods; namely:

- quasi-newton method (Section 3.5.1),
- steepest gradient descent method (Section 3.5.2), and
- Nesterov's accelerated gradient descent method (Section 3.5.3).

which are integrated into the proposed method.

Lemma 4

If

- $\epsilon \geq \Delta\epsilon(\gamma)$, and
- $\epsilon_\gamma \geq \bar{\epsilon}_\gamma$

are guaranteed, the error bound ϵ is guaranteed for ϵ such as that the inequality now is shown in eqn. (3.35).

$$\epsilon \geq \bar{\epsilon}_\gamma + \Delta\epsilon(\gamma) \geq \epsilon \quad (3.35)$$

where

$\Delta\epsilon(\gamma)$ and $\bar{\epsilon}_\gamma$ are defined in eqn. (3.20), and

ϵ_γ is defined as following equality (eqn. (3.36)),

$$\epsilon_\gamma = \epsilon - \Delta\epsilon(\gamma) \quad (3.36)$$

Proof:

The Lemma can be directly proven using **Lemma 3**.

Hence, if $\Delta\epsilon(\gamma) > \epsilon$ for a particular range of γ , the γ (e.g. reduce the γ) can be always tuned so that $\epsilon \geq \Delta\epsilon(\gamma)$ to have the error bound of ϵ (defined in eqn. (3.20)) of $F(x)$ with $\epsilon \geq \epsilon$ as well as the bound ϵ_γ for error bound $\bar{\epsilon}_\gamma$ of $\bar{F}_\gamma(x)$ with $\epsilon_\gamma \geq \bar{\epsilon}_\gamma$ are guaranteed.

Lemma 5

When γ is assigned as a function which has the same complexity with the convergence complexity of

- $\bar{\epsilon}_\gamma$, and
- $\lim_{k \rightarrow \infty} D_\gamma = 0$,

the convergence complexity of ϵ will be the same as the convergence complexity of $\bar{\epsilon}_\gamma$.

Proof:

It can be directly proven from eqn. (3.20) of **Lemma 3**.

3.5.1. Quasi-Newton Method

This subsection shows that x converges toward x^* superlinearly and $F(x)$ converges to $F(x^*)$ R-linearly when quasi-Newton method and some assumptions are applied.

Lemma 6

When

- BFGS quasi-Newton method is used with a step size α_k equals to 1, and
- either γ
 - is fixed and nearly equal to 0, or
 - changes dynamically and converges to near to 0 within a small iteration number,

x converges toward x^* superlinearly.

Proof:

Theorem 6.6 in [416] and **Theorem 4.11** in [417] show that x converges to the minimizer x_γ^* superlinearly as shown in eqn. (3.37) when the Hessian matrix is Lipschitz continuous at x_γ^* .

$$\lim_{k \rightarrow \infty} \frac{\|B_k - \nabla \bar{F}_\gamma(x_\gamma^*)\|}{\|x_{k+1} - x_k\|} = 0 \quad (3.37)$$

Theorem 6.6 in [416] and **Theorem 4.15** in [417] also states that the Wolfe conditions are satisfied when x is closed to the solution and step size α_k (see **Algorithms B.1** and **B.2** in the Appendix B) is equal to 1.

As $\lim_{\gamma \rightarrow 0} x_\gamma^* = x^*$; hence x converges toward x^* superlinearly with BFGS quasi-Newton method and step size α_k equal to 1 when γ is nearly equal to 0 is proven. It is also true if γ changes dynamically and also converges to near zero within a small iteration number.

Hence, x converges toward x^* superlinearly when

- either γ is fixed and nearly equal to 0 or γ changes dynamically and converges to

near to 0 within a small iteration number,

- BFGS quasi-Newton method is applied, and
- step size α_k equals to 1.

Lemma 6 is proven.

Lemma 7

When

- *BFGS quasi-Newton method is used with*
 - *the backtracking inexact line search, and*
 - *the Wolfe conditions and,*
- *γ which obeys one of the following conditions:*
 - *is nearly equal to 0,*
 - *changes dynamically and converges to near to 0 within a small iteration number, or*
 - *is a function which converges R-linearly and; hence causes the $\lim_{k \rightarrow \infty} D_\gamma = 0$,*

$F(x)$ converges to $F(x^*)$ R-linearly.

Proof:

Lemma 4.14 in [417] shows that $\bar{F}_\gamma(x)$ converges to $\bar{F}_\gamma(x_\gamma^*)$ R-linearly with following inequality (eqn. (3.38)),

$$\bar{F}_\gamma(x_{k+1}) - \bar{F}_\gamma(x_\gamma^*) \leq \left[1 - m_1(1 - m_2) \frac{l}{2L} \cos^2 \theta \right] (\bar{F}_\gamma(x_k) - \bar{F}_\gamma(x_\gamma^*)) \quad (3.38)$$

when $\nabla \bar{F}_\gamma(x)$ is Lipschitzian, with respective constants l and L .

As $\lim_{\gamma \rightarrow 0} x_\gamma^* = x^*$ and $\lim_{\gamma \rightarrow 0} \bar{F}_\gamma(x_\gamma^*) = F(x^*)$; hence $F(x)$ converges toward $F(x^*)$ R-linearly with BFGS quasi-Newton method and the Wolfe conditions. It is also true when γ changes dynamically and converges to near to 0 within a small iteration number.

With **Lemma 5**, $F(x)$ converges R-linearly toward $F(x^*)$ with BFGS quasi-Newton method and Wolfe conditions when γ is a function which converges R-linearly and hence

causes the $\lim_{k \rightarrow \infty} D_\gamma = 0$.

Hence, $F(x)$ converges to $F(x^*)$ R-linearly when BFGS quasi-Newton method is used with backtracking inexact line search, the Wolfe conditions and γ which obeys one of the following conditions:

- is nearly equal to 0,
- changes dynamically and converges to near to 0 within a small iteration number, or
- is a function which converges R-linearly and hence causes the $\lim_{k \rightarrow \infty} D_\gamma = 0$.

Lemma 7 is proven.

3.5.2. Steepest Gradient Descent Method

In this subsection, $F(x)$ converges to $F(x^*)$ Q-linearly with steepest gradient descent method for an arbitrary norm and some assumptions is shown.

Lemma 8

When

- *steepest descent method for an arbitrary norm is used with the backtracking inexact line search, and*
- *γ obeys one of the following conditions:*
 - *is nearly equal to 0,*
 - *changes dynamically and converges to near to 0 within a small iteration number, or*
 - *is a function which converges Q-linearly and hence causes the $\lim_{k \rightarrow \infty} D_\gamma = 0$,*

$F(x)$ converges to $F(x^*)$ Q-linearly.

Proof:

Section 9.4.3 in [418] and **Theorem 2.15** in [417] have shown that $\bar{F}_\gamma(x)$ converges to $\bar{F}_\gamma(x_\gamma^*)$ Q-linearly with the following inequality (eqn. (3.39)).

$$\bar{F}_\gamma(x_{k+1}) - \bar{F}_\gamma(x_\gamma^*) \leq c(\bar{F}_\gamma(x_k) - \bar{F}_\gamma(x_\gamma^*)) \quad (3.39)$$

and hence with the following inequality (eqn. (3.40)).

$$\bar{F}_\gamma(x_k) - \bar{F}_\gamma(x_\gamma^*) \leq c^k(\bar{F}_\gamma(x_0) - \bar{F}_\gamma(x_\gamma^*)) \quad (3.40)$$

where $c < 1$ is a constant when steepest descent method for an arbitrary norm and backtracking inexact line search are used.

As $\lim_{\gamma \rightarrow 0} x_\gamma^* = x^*$ and $\lim_{\gamma \rightarrow 0} \bar{F}_\gamma(x_\gamma^*) = F(x^*)$, hence $F(x)$ converges toward $F(x^*)$ Q-linearly with steepest descent method for an arbitrary norm and backtracking inexact line search. It is also true when γ changes dynamically and converges to near to 0 within a small iteration number.

With **Lemma 5**, $F(x)$ converges toward $F(x^*)$ Q-linearly with the steepest descent method for an arbitrary norm and backtracking line search when γ is a function which converges Q-linearly and hence causes the $\lim_{k \rightarrow \infty} D_\gamma = 0$.

Hence, $F(x)$ converges to $F(x^*)$ Q-linearly, when steepest descent method for an arbitrary norm is used with backtracking inexact line search and γ obeys one of following conditions:

- is nearly equal to 0,
- changes dynamically and converges to near to 0 within a small iteration number, or
- is a function which converges Q-linearly and hence causes the $\lim_{k \rightarrow \infty} D_\gamma = 0$.

Lemma 8 is proven.

3.5.3. Nesterov's Accelerated Gradient Descent Method

In this subsection, $F(x)$ converges to $F(x^*)$ with the complexity of $O(\frac{1}{k^2})$ when Nesterov's accelerated gradient descent method is used with some assumptions is shown.

Lemma 9

When

- *Nesterov's accelerated gradient descent method is applied, and*
- *either γ*
 - *is nearly equal to 0, or*

- *changes dynamically and converges to near to 0 within a small iteration number,*

$F(x)$ converges to $F(x^*)$ with the complexity of $O(\frac{1}{k^2})$.

Proof:

Recap the following updates used in **Algorithms B.1** and **B.2**.

$$\eta_{-1} = 1, \eta_k = \frac{1 + \sqrt{1 + 4\eta_k^2}}{2}, \rho_k = \frac{1 - \eta_k}{\eta_{k+1}} \text{ and } x_{k+1} = (1 - \rho_k)\check{x}_k + \rho_k\check{x}_{k-1}$$

where $\check{x}_k = \text{Prox}_{\bar{F}_\gamma}^\gamma(x_{k-1})$ when Nesterov's accelerated gradient descent method is used.

Theorem 3.19 in [419] shows that the convergence complexity of the above updates is $O(\frac{1}{k^2})$ as shown in the following inequality (eqn. (3.41)).

$$\bar{F}_\gamma(y_k) - \bar{F}_\gamma(x_\gamma^*) \leq \frac{2\|x_1 - x_\gamma^*\|}{\gamma k^2} \quad (3.41)$$

By assuming strong convexity of $\bar{F}_\gamma(x)$, the following inequality (eqn. (3.42)) is formed.

$$\bar{F}_\gamma(x_k + \Delta x) \geq \bar{F}_\gamma(x) + \Delta x \nabla \bar{F}_\gamma(x) + \frac{1}{2} \Delta x^T m_\gamma \Delta x \quad (3.42)$$

where m_γ satisfied the following inequality (eqn. (3.43)).

$$\nabla^2 \bar{F}_\gamma(x) \succcurlyeq m_\gamma I_N \quad (3.43)$$

where I_N is a $N \times N$ -dimensional identity matrix.

As $y_{k+1} = x_k - \gamma \nabla \bar{F}_\gamma(x_k)$ and when eqn. (3.43) is substituted into eqn. (3.42), the following inequality (eqn. (3.44)) is formed.

$$\frac{2\|x_1 - x_\gamma^*\|}{\gamma k^2} \geq \bar{F}_\gamma(x) - \bar{F}_\gamma(x_\gamma^*) + \Delta x \nabla \bar{F}_\gamma(x) + \frac{1}{2} \Delta x^T m_\gamma \Delta x \quad (3.44)$$

where

$$\Delta x = -\gamma \nabla \bar{F}_\gamma(x_k) \quad (3.45)$$

The following inequality (eqn. (3.46)) is formed after eqn. (3.45) is substituted into eqn. (3.44).

$$\frac{2\|x_1 - x_\gamma^*\|}{\gamma k^2} \geq \bar{F}_\gamma(x) - \bar{F}_\gamma(x_\gamma^*) - \gamma \nabla \bar{F}_\gamma^2(x) + \frac{\gamma^2 m_\gamma}{2} \nabla \bar{F}_\gamma^2(x) \quad (3.46)$$

Hence, $\bar{\varepsilon}_\gamma$ is bounded as shown in the following inequality (eqn. (3.47)).

$$\frac{2\|x_1 - x_\gamma^*\|}{\gamma k^2} + \frac{\gamma}{2} (2 + \gamma m_\gamma) \nabla \bar{F}_\gamma^2(x) \geq \bar{\varepsilon}_\gamma \quad (3.47)$$

where $\bar{\varepsilon}_\gamma = \bar{F}_\gamma(x) - \bar{F}_\gamma(x_\gamma^*)$

The following inequality (eqn. (3.48)) is formed by combining eqns (3.47), (3.22) and (3.20).

$$\begin{aligned} D_\gamma + \frac{2\|x_1 - x_\gamma^*\|}{\gamma k^2} + \frac{\gamma}{2} [2\nabla \bar{F}_{\gamma'}(x) \nabla \bar{F}_\gamma(x) + \\ (1 + \gamma m_\gamma - \gamma m_{\gamma'}) \nabla \bar{F}_\gamma^2(x) + \|u_0^*(x)\|_c^2 - \|u_\gamma^*(x)\|_c^2] \geq \varepsilon \end{aligned} \quad (3.48)$$

When either γ is nearly equal to 0 or γ changes dynamically and converges to near to 0 within a small iteration number, the left-hand side of eqn. (3.48) tends to be $\frac{2\|x_1 - x_\gamma^*\|}{\gamma k^2}$.

Hence, $F(x)$ converges to $F(x^*)$ with the complexity of $O(\frac{1}{k^2})$ when Nesterov's accelerated gradient descent method and either γ is nearly equal to 0 or γ changes dynamically and converges to near to 0 within a small iteration number. **Lemma 9** is proven.

3.5.4. Discussion of Convergence Complexity of the Proposed Method and several States of the Art Methods

The methods which are included in this discussion are

- the proposed proximal average quasi-Newton (PA-QN) method,
 - the proximal average based accelerated proximal gradient (PA-APG) method [8],
- and

- the adaptive proximal average based proximal gradient (APA-APG) method [1].

Regarding Nesterov's accelerated gradient descent method (see Section 3.5.3), the proof of **Lemma 9** shows that the bound of ε (which is defined in **Lemma 3**) is bounded by $\frac{2\|x_1 - x_\gamma^*\|}{\gamma k^2}$ and $\frac{2\|x_1 - x_\gamma^*\|}{\gamma k^2}$ tends to be large when γ is small. Moreover, when k is large, $x_k \approx \text{Prox}_{\bar{F}_\gamma}^\gamma(x_k)$ (based on fixed point theory [409]). $\|\nabla \bar{F}_\gamma(x_k)\|$ (as $\text{Prox}_{\bar{F}_\gamma}^\gamma(x_k)$ is near to its fixed points and the definition in eqn. (3.4)) is extremely small and hence $\|x_k - x_{k-1}\|$ (as the step depends on gradient for gradient descent methods) is extremely small too. Two issues are faced by Nesterov's accelerated gradient descent method;

- when γ is small, x_k takes a longer time to converge to the minimizer (the bound $\frac{2\|x_1 - x_\gamma^*\|}{\gamma k^2}$ tends to be large), and
- when k is large, the convergence is slowing down ($\|\nabla \bar{F}_\gamma(x_k)\|$ and $\|x_k - x_{k-1}\|$ are extremely small).

These are the same issues faced by PA-APG method as the convergence complexity of PA-APG is guaranteed to be $O(\frac{1}{k^2})$ only when γ is near to 0.

As the proposed method not only consider the descent direction derived from Nesterov's accelerated gradient descent method, hence, the convergence can be accelerated by the descent directions derived from other integrated methods which include not only other first-order methods but also the second-order methods. First-order method's step magnitude is highly dependent on the magnitude of the first-order information. It causes the convergence to be slow when the magnitude of the first-order information is low (the magnitude of the step will also be low). Hence, second-order methods which find zeroes from the approximate objective function and/or approximate second-order information tend to converge quicker than first-order methods in term of steps.

Similarly to APA-APG, **Lemma 7** and **Theorem 1** in [1] states that the bound of its error of objective function $F(x)$ is as below (see eqn. (3.49)).

$$\begin{aligned}
F(x_{k+1}) - F(x^*) &\leq \frac{\tau_k^2}{\gamma_{k+1}} \left\{ \frac{\gamma_1}{\tau_0^2} [F_{\gamma_1}(x_1) - F(x^*)] + \right. \\
&\quad \left. \frac{1}{2} \|x^* - \tilde{x}^1\|^2 + \gamma_1^2 a \left\| \text{Diam}_{\frac{1}{c^2 Q}} \right\|^2 \right\} + \frac{\gamma_{k+1}}{2} \left\| \text{Diam}_{\frac{1}{c^2 Q}} \right\|^2
\end{aligned} \tag{3.49}$$

where

$$\gamma_{k+1} = \min \left(\frac{1}{k+a}, \frac{1}{L_f} \right),$$

$$\tau_k = \frac{1}{k+a},$$

$a > 0$ is a constant, and

the descriptions of notations of $\tilde{x}^1$, L_f and $\text{Diam}_{\frac{1}{c^2 Q}}$ can be found in [1].

From eqn. (3.49), γ_{k+1} needs to be small so that $F(x_{k+1})$ is guaranteed to converge to $F(x^*)$ due to the term of $\frac{\gamma_{k+1}}{2} \left\| \text{Diam}_{\frac{1}{c^2 Q}} \right\|^2$. If γ_{k+1} is small, τ_k^2 need to be small to ensure the term of $\frac{\tau_k^2}{2\gamma_{k+1}} \|x^* - \tilde{x}^1\|^2$ to be small. τ_k^2 is only very small when k is very large. However, when τ_k^2 is very small, x_{k+1} tend to be $\text{Prox}_{\tilde{F}_{\gamma_{k+1}}}^{\gamma_{k+1}}(\hat{x}^k - \gamma_{k+1} \nabla f(\hat{x}^k))$ where the descriptions of notations of $\hat{x}^k$ and $\nabla f(\hat{x}^k)$ can be found in [1]. As the step depends on $\text{Prox}_{\tilde{F}_{\gamma_{k+1}}}^{\gamma_{k+1}}(\hat{x}^k - \gamma_{k+1} \nabla f(\hat{x}^k))$ and $\left\| \text{Prox}_{\tilde{F}_{\gamma_{k+1}}}^{\gamma_{k+1}}(\hat{x}^k - \gamma_{k+1} \nabla f(\hat{x}^k)) - x_{k-1} \right\|$ is extreme small; hence, the convergence is slow when k is large. In overall, the method converges slower especially when k is large.

3.6. Experiments

In this section, experiments on an important problem in machine learning: the sum of multiple sparsities (L_1 -norms) for sparse representation problem are conducted. It has been widely used in face recognition [3][260][4][147], anomaly detection [6][5] and compressed sensing. The example contains a more computationally expensive individual proximal mapping than the ones in problems of the overlapping group Lasso and graph-guided fused Lasso which are used in the previous works [1][277]. Sum of multiple sparsities is more representative for large scale machine learning problem such as those problems with high-resolution images. The problem of the sum of multiple sparsities can be transformed to a linear programming problem. Note that the focus of this chapter is to

propose a method which can generally solve the convex optimization problem (which may not solvable by linear programming) whereby the objective function contains non-smooth composite rather than linear programming problems. Hence, the algorithms for solving linear programming problem are not considered in the experiments. The results of the proposed algorithms, Greedy PA-QN and PA-QN, are compared against the results of PA-APG in [277] and the state-of-the-art algorithms, APA-APG1 and APA-APG2 in [1]. Note that Greedy PA-QN is especially more useful than its alternative PA-QN when parallel computing acceleration is available to be used in computationally expensive problems.

3.6.1. Problem of Sum of Multiple Sparsities

The following problem (see eqn. (3.50)) is the problem of the sum of multiple sparsities to be solved in the following experiments.

$$F^* = \min_{x \in \mathbb{X}} F(x) = \min_{x \in \mathbb{X}} \alpha_1 \|x\|_1 + \alpha_2 \|y - Ax\|_1 \quad (3.50)$$

where

- $\mathbb{X}$ is a finite-dimensional linear space,
- $A \in \mathbb{R}^{m \times n}$ is a dictionary,
- $y \in \mathbb{R}^m$ is a vector of observations (measurements),
- $x \in \mathbb{R}^n$ is a sparse coefficient vector (sparse signal in compressed sensing) of the observation sample y corresponding to the dictionary A ,
- $f_1(x) = \|x\|_1$,
- $f_2(x) = \|y - Ax\|_1$, and
- α_1 and α_2 are two constants.

Note that the $f_1(x)$, $f_2(x)$, α_1 and α_2 in eqn. (3.50) correspond to the $f_i(x)$ and α_i in eqns (3.5)-(3.8).

The purpose of the problem in eqn. (3.50) is to estimate the ground truth sparse coefficient vector x_o (x is used to estimate x_o) of the noisy observation sample y with its corresponding dictionary A .

The proximal mapping of $f_1(x)$ (see eqns (3.50) and (3.52)), $Prox_{f_1}^\gamma(x)$, is simply a shrinkage operator which performs the element-wise operation as shown in eqn. (3.51).

$$shrink(o, p) = (s - p)_+ - (-o - p)_+ = \begin{cases} o_i - p, & o_i \geq p \\ 0, & |o_i| \leq p \\ o_i + p, & o_i \leq -p \end{cases} \quad (3.51)$$

$$Prox_{f_1}^\gamma(x) = shrink(x, \gamma) \quad (3.52)$$

where

$\gamma = \gamma_a = \frac{\bar{\gamma} \cdot \|b\|_1}{2m}$ is used in PA-QN, Greedy PA-QN and PA-APG, in which $\bar{\gamma}$ is a parameter, m is the row number of A and is also the dimension of b which is defined in eqn. (3.50), and

$\gamma = \gamma_a \cdot \frac{a}{(k+a)}$ is used in APA-APG1 and APA-APG2, in which k is the iteration number and a is a parameter.

The proximal mapping of $f_2(x)$ (see eqn. (3.50)), $Prox_{f_2}^\gamma(x)$, is defined in eqn. (3.53).

$$Prox_{f_2}^\gamma(x) = \underset{u \in \mathbb{X}}{\operatorname{argmin}} \|y - Au\|_1 + \frac{1}{2\gamma} \|u - x\|^2 \quad (3.53)$$

Eqn. (3.53) can be reformulated to eqn. (3.54) and it can be solved by the alternating direction method of multipliers (ADMM) [285] with the **Algorithm B.3** in Appendix B.

$$Prox_{f_2}^\gamma(x) = \underset{u \in \mathbb{X}}{\operatorname{argmin}} \|\check{e}\|_1 + \frac{1}{2\gamma} \|u - x\|^2 \quad (3.54)$$

s.t. (such that) $y = Au + \check{e}$

The four parameters, (m, n, k_1, k_2) are used to represent the characteristics of the test sets in the experiments. m and n are the row and column (atom) dimensions of dictionaries while they are also the dimensions of the observation sample y and sparse coefficient vector x of the sample respectively. k_1 and k_2 are the numbers of non-zero elements in the ground truth sparse coefficient and error vectors, x_o and e_o . Three datasets which have the same number of dictionary atoms with different (m, n, k_1, k_2) are used in the experiments. Table 3.1 shows the characteristics of the three datasets.

Table 3.1 Characteristics of experimental datasets

Set	(m, n, k_1, k_2)	Dimensions of sample and dictionary
Set 1	(50, 100, 5, 3)	Low
Set 2	(100, 100, 5, 5)	Medium
Set 3	(1000, 100, 5, 50)	High

The datasets are generated with the procedure shown in Figure 3.1.

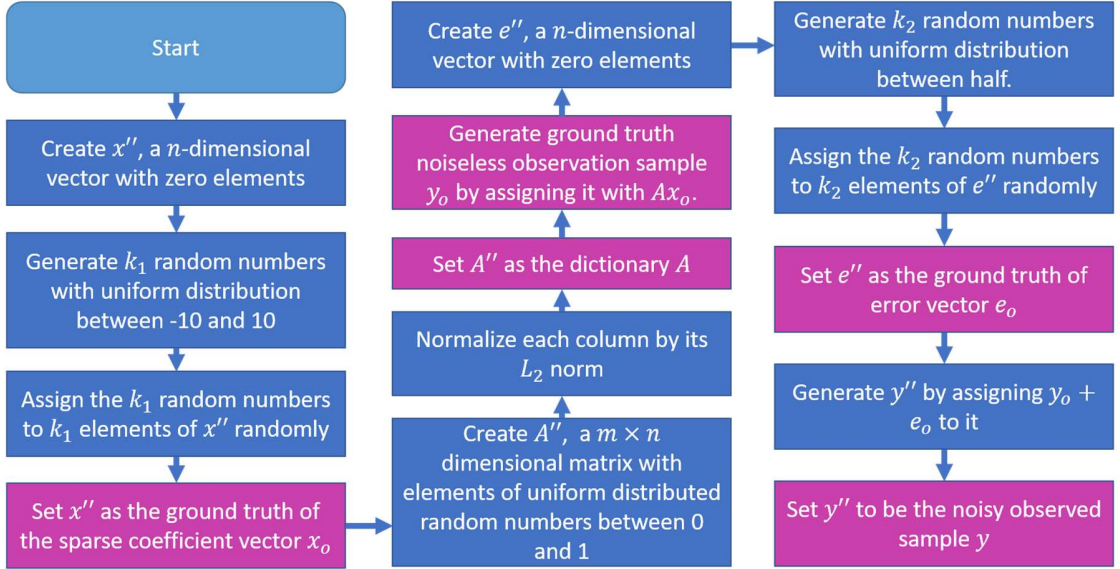


Figure 3.1 Dataset generation procedure

Two sets of α_1 and α_2 (see eqn. (3.50)); namely: 0.5 and 0.5 and $\frac{1}{1+10^5}$ and $\frac{10^5}{1+10^5}$ respectively, are used in the experiments. The first set is used for investigating the performance of each algorithm on the objective function (in eqn. (3.50)) which consists of approximately equally dominated individual convex functions, $\alpha_1 f_1(x)$ and $\alpha_2 f_2(x)$ (see eqn. (3.50)) while the other one is for investigating the algorithm performances when the $\alpha_2 f_2(x)$ is dominating in the objective function.

Note that the purpose of conducting experiments with the two different sets of α_1 and α_2 is to investigate the convergence behaviours of all algorithms on the minimizations for different objective functions with different characteristics. The two objective functions are convex composites described as following:

- the first composite contains two equally-weighted (non-biased) non-smooth functions, and
- the second composite contains two non-equally-weighted non-smooth functions and one function is highly weighed (biased) than another one.

Four evaluation methods are used; namely:

- $F(x_k)$ vs k : the objective function value $F(x_k)$ versus the convergence speed of the precision in term of iteration k ,
- ϵ_p vs k : the precision ϵ_p of the estimated sparse coefficient vector x_k versus the convergence speed of the precision in term of iteration k ,
- $F(x_k)$ vs t : the objective function value $F(x_k)$ versus the convergence speed of the precision in term of time t , and
- ϵ_p vs t : the precision ϵ_p of the estimated sparse coefficient vector x_k versus the convergence speed of the precision in term of time t .

The precision ϵ_p at k^{th} iteration of an algorithm is defined in eqn. (3.55).

$$\epsilon_p = \frac{\|x_k - x_o\|_2}{\|x_o\|_2} \quad (3.55)$$

where

- x_k is the estimated sparse coefficient vector at k^{th} iteration, and
- x_o is the ground truth of the sparse coefficient vector.

Note that the results of the last two performance measures ($F(x_k)$ vs t , and ϵ_p vs t) of all experiments are presented in Appendix B. It is due to the convergence behaviours of all algorithms with the last two performance measures ($F(x_k)$ vs t , and ϵ_p vs t) are similar to the ones with the first two performance ($F(x_k)$ vs k , and ϵ_p vs k). Moreover, the presented convergence behaviour results of all methods are not boosted by parallel computing. All methods are parallelable accelerated Jacobi methods. However, the proposed methods (PA-QN and Greedy PA-QN) can benefit more with parallelable computing acceleration as the acceleration can be applied in the “for-loops” for determining the objective values of many different step size of both methods (See **Algorithms B.1** and **B.2**) too.

The following two subsections show the results of performances of various algorithms which solve the two minimizations (eqn. (3.50)) with the two different parameter sets (the two sets of α_1 and α_2 , 0.5 and 0.5 and $\frac{1}{1+10^5}$ and $\frac{10^5}{1+10^5}$ respectively).

Two proposed methods; namely:

- proposed proximal average quasi-Newton methods (PA-QN), and
- Greedy PA-QN.

and three state-of-the-art Jacobi type methods; namely:

- adaptive proximal average based proximal gradient method 1 (APA-APG1) [1],
- APA-APG2 [1], and
- proximal average based accelerated proximal gradient (PA-APG) method [8].

are used in performance comparisons.

The best parameters set of each algorithm is selected by a particle swarm algorithm³ (PSO) with a specific range of each parameter. The parameter range of each parameter to be selected by the PSO algorithm is listed in Table 3.1.

The definitions of m_1, m_2 in PA-QN can be found in Algorithm A.1. The τ of both PA-QN and Greedy PA-QN can be found in Algorithms A.1 and A.2 respectively. The a in APA-APG1 and APA-APG2 can be found in both eqns (3.49) and (3.52). The $\bar{\gamma}$ in all algorithms can be found in eqn. (3.52). The $\bar{\gamma}_{ADMM}$ in all algorithms is a parameter used in ADMM algorithm for solving $Prox_{f_2}^\gamma(x)$ in eqn. (3.53).

³ The particle swarm function, based on the algorithm described in [459], using modification suggested in [460], [461], in Matlab R2018b is used.

Table 3.2 Parameters to be optimized using particle swarm algorithm and their ranges

Algorithm	Parameter	Optimized Range
PA-QN	m_1	0 – 1
	m_2	0 – 1
	τ	0 – 1
	$\bar{\gamma}$	$10^{-10} - 10$
	$\bar{\gamma}_{ADMM}$	0.9 – 1.1
Greedy PA-QN	τ	0 – 1
	$\bar{\gamma}$	$10^{-10} - 10$
	$\bar{\gamma}_{ADMM}$	0.9 – 1.1
APA-APG1	a	$1 - 10^{-3}$
	$\bar{\gamma}$	$10^{-10} - 10$
	$\bar{\gamma}_{ADMM}$	0.9 – 1.1
APA-APG2	a	$1 - 10^{-3}$
	$\bar{\gamma}$	$10^{-10} - 10$
	$\bar{\gamma}_{ADMM}$	0.9 – 1.1
PA-APG1	$\bar{\gamma}$	$10^{-10} - 10$
	$\bar{\gamma}_{ADMM}$	0.9 – 1.1

3.6.2. Experiments with $\alpha_1 = 0.5$ and $\alpha_2 = 0.5$.

In this subsection, the convergence behaviours of all algorithms on the problem in eqn. (3.50) in which objective function consists of two equally-weighted (non-biased, $\alpha_1 = 0.5$ and $\alpha_2 = 0.5$) non-smooth functions are investigated.

Figure 3.2, Table 3.3 and Table 3.4 show the following performance measures; namely:

- $F(x_k)$ vs k , and
- ϵ_p vs k ,

of the proposed algorithms and the other state-of-the-art algorithms on low, medium and high dimensional datasets with (m, n, k_1, k_2) equal to $(50, 100, 5, 3)$, $(100, 100, 5, 5)$ and $(1000, 100, 5, 50)$ respectively. Both α_1 and α_2 in objective function equal to 0.5.

The top, middle and bottom graphs in Figure 3.2 shows the comparisons when low, medium and high dimensional datasets with $(m, n, k_1, k_2) = (50, 100, 5, 3)$, $(m, n, k_1, k_2) = (100, 100, 5, 5)$ and $(m, n, k_1, k_2) = (1000, 100, 5, 50)$ are used respectively. The first

row of Table 3.3 Objective function value Fxk versus iteration number k on sum of multiple sparsities with $\alpha_1 = 0.5$ and $\alpha_2 = 0.5$

and Table 3.4 shows which dataset (e.g. (m, n, k_1, k_2)) is used, while the second row shows 3 objective values (in Table 3.3 Objective function value Fxk versus iteration number k on sum of multiple sparsities with $\alpha_1 = 0.5$ and $\alpha_2 = 0.5$

) and ϵ_p values (in Table 3.4) which are separated by symbol “|” and the third to seventh rows show the number of iterations k of each method to reach the precision ϵ_p stated in the second row.

Note that the following two performance measures:

- $F(x_k)$ vs t , and
- ϵ_p vs t

of all above-mentioned algorithms and cases are in Figure B.1, Table B.1 and Table B.2 in Appendix B.

Figure 3.2, Table 3.3 Objective function value Fxk versus iteration number k on sum of multiple sparsities with $\alpha_1 = 0.5$ and $\alpha_2 = 0.5$

and Table 3.4 show that the proposed algorithms, PA-QN and Greedy PA-QN, converge faster (much lesser steps) and can reach low objective function values and high precisions in term of sparse coefficient vector x estimation quicker (much lesser steps) in almost all cases.

When low dimensional dataset with $(m, n, k_1, k_2) = (50, 100, 5, 3)$ is used, Figure 3.2 and Table 3.3 show that Greedy PA-QN reaches $F(x_k)$ below 70, 60 and 50 quickest in 1, 6 and 23 iterations and converges quickest around at 200th iteration in $F(x_k)$ among all algorithms. PA-QN reaches $F(x_k)$ both 70, 60 and 50 in 4, 17 and 55 iterations and converges around at 400th iteration in term of $F(x_k)$. APA-APG1 and APA-APG1 reaches $F(x_k)$ below 70, 60 and 50 in 5, 23 and 99 iterations and 5, 20 and 84 iterations respectively. Both algorithms reach these $F(x_k)$ much faster than PA-APG and slower than PA-QN and Greedy PA-QN. PA-APG reaches $F(x_k)$ below 70, 60 and 50 in 73, 157 and 330 iterations

which are much slower than all other algorithms. APA-APG1, APA-APG2 and PA-APG do not converge in 1000 iterations in term of $F(x_k)$. Figure 3.2 and Table 3.4 show that PA-QN reaches the highest precision (the lowest ϵ_p) among all algorithm in 1000 iterations. PA-QN reaches ϵ_p below 10^{-3} , 10^{-4} and 10^{-5} in 419, 459 and 640 iterations. PA-QN converges at around 650th iteration. PA-QN is the second quickest algorithm and Greedy PA-QN is the quickest algorithm in reaching ϵ_p below 10^{-3} and 10^{-4} and in convergence in term of ϵ_p . Greedy PA-QN converge around 400th iteration in term of ϵ_p . Greedy PA-QN, APA-APG1 and APA-APG2 reach ϵ_p below 10^{-3} and 10^{-4} in 185 and 273 iterations, 815 and 952 iterations and 830 and 958 iterations respectively. PA-APG reaches ϵ_p below 10^{-3} in 920 iterations. Greedy PA-QN, APA-APG1, APA-APG2 and PA-APG do not reach ϵ_p below 10^{-5} and PA-APG does not reach ϵ_p below 10^{-4} in 1000 iterations. APA-APG1, APA-APG2 and PA-APG do not converge in 1000 iteration in term of ϵ_p . In conclusion, Greedy PA-QN is the quickest algorithm in terms of convergence time in both the optimization of $F(x_k)$ and the estimation of x_k and only Greedy PA-QN and PA-QN converges in 1000 iteration for the low dimensional case. Moreover, PA-QN has the highest precision in 1000 iteration for the low dimensional case.

When medium dimensional dataset with $(m, n, k_1, k_2) = (100, 100, 5, 5)$ is used, Figure 3.2 and Table 3.3 show that Greedy PA-QN reaches $F(x_k)$ below 50, 40 and 30 quickest in 3, 5 and 12 iterations and converges quickest around at 50th iteration in $F(x_k)$ among all algorithms. PA-QN reaches $F(x_k)$ both 50, 40 and 30 in 8, 11 and 29 iterations and converges around at 100th iteration in term of $F(x_k)$. APA-APG1 and APA-APG2 reaches $F(x_k)$ below 50, 40 and 30 in 50, 85 and 133 iterations and 40, 68 and 106 iterations respectively. Both algorithms reach these $F(x_k)$ much faster than PA-APG but much slower than PA-QN and Greedy PA-QN. PA-APG reaches $F(x_k)$ below 50, 40 and 30 in 235, 307 and 385 iterations which are much slower than all other algorithms. APA-APG1, APA-APG2 and PA-APG do not converge in 1000 iterations in term of $F(x_k)$. Figure 3.2 and Table 3.3 show that Greedy PA-QN reaches the highest precision (the lowest ϵ_p) among all algorithm in 1000 iterations. Greedy PA-QN reaches ϵ_p below 10^{-3} , 10^{-4} and 10^{-5} in 82, 116 and 144 iterations. Greedy PA-QN converges at around 150th iteration. In

conclusion, Greedy PA-QN is the quickest algorithm in reaching ϵ_p below 10^{-3} and 10^{-4} and in convergence in term of ϵ_p while PA-QN is the second quickest algorithm and perform almost equally as Greedy PA-QN. PA-QN converges around at 250th iteration in term of ϵ_p . PA-QN, APA-APG1, APA-APG2 and PA-APG reach ϵ_p below 10^{-3} and 10^{-4} in 91 and 125 iterations, 714 and 965 iterations, 600 and 900 iterations and 891 and 983 iterations respectively. All algorithms do not reach ϵ_p below 10^{-5} in 1000 iterations except Greedy PA-QN. APA-APG1, APA-APG2 and PA-APG do not converge in 1000 iteration in term of ϵ_p . In conclusion, Greedy PA-QN and PA-QN are the quickest and second quickest algorithms in terms of convergence time in both the optimization of $F(x_k)$ and the estimation of x_k and are the only algorithms converges in 1000 iteration for the medium dimensional case. Moreover, Greedy PA-QN has the highest precision in the estimation of x_k while PA-QN has similar precision with Greedy PA-QN in 1000 iterations for the medium dimensional case.

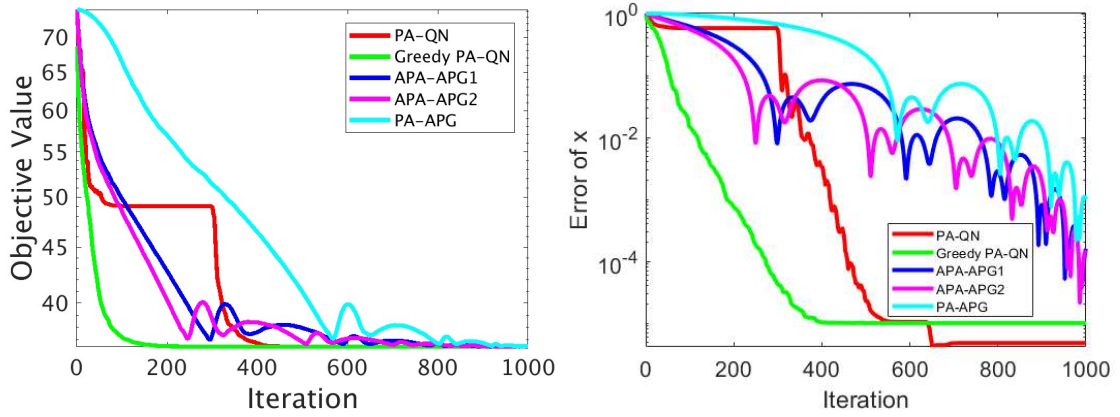
When high dimensional dataset with $(m, n, k_1, k_2) = (1000, 100, 5, 50)$ is used, Figure 3.2 and Table 3.3 show that Greedy PA-QN reaches $F(x_k)$ below 200, 150 and 100 quickest in 2, 3 and 3 iterations and converges quickest in very few iterations in $F(x_k)$ among all algorithms. PA-QN reaches $F(x_k)$ both 200, 150 and 100 in 4, 4 and 6 iterations and also converges around in very few iterations in term of $F(x_k)$. APA-APG1 and APA-APG1 reaches $F(x_k)$ below 200, 150 and 100 in 58, 73 and 90 iterations and 42, 52 and 64 iterations respectively. Both algorithms reach these $F(x_k)$ much faster than PA-APG but much slower than PA-QN and Greedy PA-QN. PA-APG reaches $F(x_k)$ below 200, 150 and 100 in 258, 290 and 322 iterations which are much slower than all other algorithms. APA-APG1, APA-APG1 and PA-APG do not converge in 1000 iterations which are much slower than PA-QN and Greedy PA-QN in term of $F(x_k)$. Figure 3.2 and Table 3.4 show that Greedy PA-QN reaches the highest precision (the lowest ϵ_p) and much higher precision (the final ϵ_p is below 10^{-14} when the ones of all other algorithms are at most near to 10^{-5}) than the other algorithms in 1000 iterations. Greedy PA-QN reaches ϵ_p below 10^{-3} , 10^{-4} and 10^{-14} in 31, 83 and 550 iterations. Greedy PA-QN converges at around 600th iteration. PA-QN is the quickest algorithm in reaching ϵ_p below 10^{-3} and

10^{-4} and in convergence in term of ϵ_p while Greedy PA-QN is the second quickest algorithm. PA-QN converges within 50 iterations in term of ϵ_p . PA-QN, APA-APG1, APA-APG2 and PA-APG reach ϵ_p below 10^{-3} and 10^{-4} in 11 and 15 iterations, 703 and 932 iterations, 623 and 916 iterations and 882 and 990 iterations respectively. All algorithms do not reach ϵ_p below 10^{-6} in 1000 iterations except Greedy PA-QN. APA-APG1, APA-APG2 and PA-APG do not converge in 1000 iteration in term of ϵ_p . In conclusion, Greedy PA-QN and PA-QN are the two quickest algorithms in terms of convergence time in both the optimization of $F(x_k)$ and the estimation of x_k and are the only algorithms converges in 1000 iteration for the high dimensional case. Moreover, Greedy PA-QN has the highest precision in the estimation of x_k which is much better than the other algorithms in 1000 iterations for the high dimensional case.

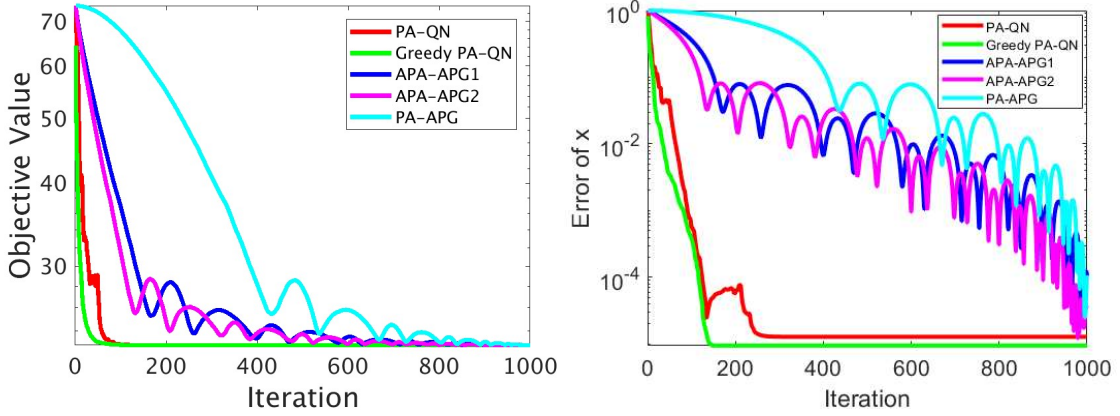
In overall, both the proposed algorithms, PA-QN and Greedy PA-QN converge and reach high precision much quicker than other algorithms in the three cases in solving eqn. (3.50) with $\alpha_1 = 0.5$ and $\alpha_2 = 0.5$. The following summary shows the comparison between PA-QN and Greedy PA-QN:

- When a low dimensional dataset is used, PA-QN converges slower than Greedy PA-QN but have higher final precision.
- When a medium dimensional dataset is used, PA-QN and Greedy PA-QN perform almost the same. Greedy PA-QN converge slightly faster and have slightly high final precision.
- When a high dimensional dataset is used, PA-QN converges faster than Greedy PA-QN but Greedy PA-QN has much higher final precision than PA-QN. It also implies that Greedy PA-QN performs significant better for high dimension case (large m).

When $(m, n, k_1, k_2) = (50, 100, 5, 3)$



When $(m, n, k_1, k_2) = (100, 100, 5, 5)$



When $(m, n, k_1, k_2) = (1000, 100, 5, 50)$

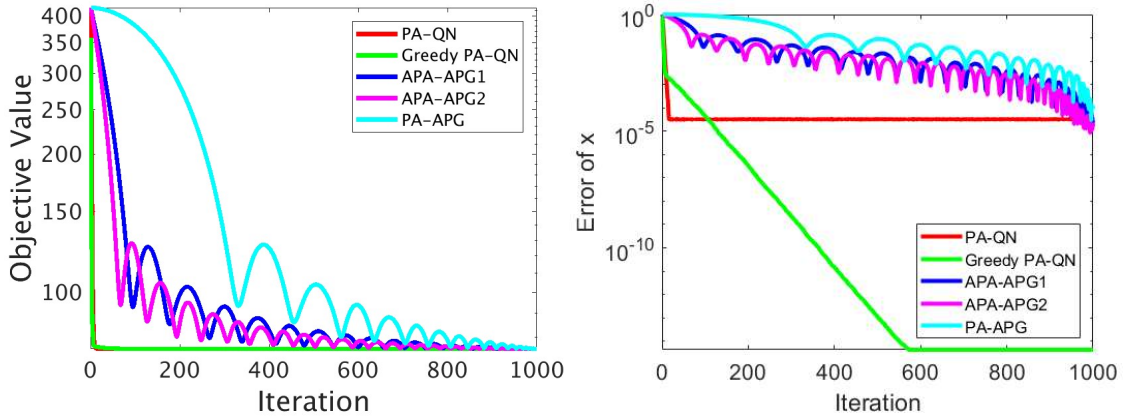


Figure 3.2 Objective function value and error vs iteration on the sum of multiple sparsities with $\alpha_1 = 0.5$ and $\alpha_2 = 0.5$

Table 3.3 Objective function value $F(x_k)$ versus iteration number k on sum of multiple sparsities with $\alpha_1 = 0.5$ and $\alpha_2 = 0.5$

Cases (m, n, k_1, k_2)	(50, 100, 5, 3)	(100, 100, 5, 5)	(1000, 100, 5, 50)
$F(x_k)$	70 60 50	50 40 30	200 150 100
PA-QN	4 17 55	8 11 29	4 4 6
Greedy PA-QN	1 6 23	3 5 12	2 3 3
APA-APG1	5 23 99	50 85 133	58 73 90
APA-APG2	5 20 84	40 68 106	42 52 64
PA-APG	73 157 330	235 307 385	258 290 322

Table 3.4 Precision ϵ_p versus iteration number k on sum of multiple sparsities with $\alpha_1 = 0.5$ and $\alpha_2 = 0.5$

Cases (m, n, k_1, k_2)	(50, 100, 5, 3)	(100, 100, 5, 5)	(1000, 100, 5, 50)
ϵ_p	$10^{-3} 10^{-4} 10^{-5}$	$10^{-3} 10^{-4} 10^{-5}$	$10^{-3} 10^{-4} 10^{-14}$
PA-QN	419 459 640	91 125 -	11 15 -
Greedy PA-QN	185 273 -	82 116 144	31 83 550
APA-APG1	815 952 -	714 965 -	703 932 -
APA-APG2	830 958 -	600 900 -	623 916 -
PA-APG	920 - -	891 983 -	882 990 -

3.6.3. Experiments with $\alpha_1 = \frac{1}{1+10^5}$ and $\alpha_2 = \frac{10^5}{1+10^5}$.

In this subsection, the convergence behaviours of all algorithms on the problem in eqn. (3.50) in which objective function consists of two non-equally-weighted non-smooth functions and one function is highly weighed (biased) than another one. The coefficients α_1 and α_2 in the objective function equal to $\frac{1}{1+10^5}$ and $\frac{10^5}{1+10^5}$ respectively.

Figure 3.3, Table 3.5 and Table 3.6 show the following performance measures; namely:

- $F(x_k)$ vs k , and
- ϵ_p vs k ,

of the proposed algorithms and the other state-of-the-art algorithms on low, medium and high dimensional datasets with (m, n, k_1, k_2) equal to (50, 100, 5, 3), (100, 100, 5, 5) and (1000, 100, 5, 50) respectively. α_1 and α_2 in objective function equal to $\frac{1}{1+10^5}$ and $\frac{10^5}{1+10^5}$ respectively.

Note that the following two performance measures:

- $F(x_k)$ vs t , and
- ϵ_p vs t

of all above-mentioned algorithms and cases are presented in Figure B.2, Table B.3 and Table B.4 in Appendix B.

Figure 3.3, Table 3.5 and Table 3.6 show that the proposed algorithms, PA-QN and Greedy PA-QN, converge faster (much lesser steps) and can reach low objective function values and high precisions in term of sparse coefficient vector x estimation quicker (much lesser steps) on almost all datasets.

When low dimensional dataset with $(m, n, k_1, k_2) = (50, 100, 5, 3)$ is used, Figure 3.3 and Table 3.5 show that all algorithms reach $F(x_k)$ below 100 in 1 iteration. Greedy PA-QN and PA-APG reach $F(x_k)$ below 70 quickest in 1 iteration while PA-QN reach it in 2 iterations. APA-APG1 reach $F(x_k)$ below 70 much slower and in 455 iterations and APA-APG2 does not reach it at all in 1000 iterations. All algorithms do not reach $F(x_k)$ below 50 except PA-QN reach it in 9 iterations. Only PA-QN and Greedy PA-QN converge in few iterations and the other algorithms do not converge in 1000 iterations in term of $F(x_k)$. Figure 3.3 and Table 3.6 show that PA-QN reaches the highest precision which ϵ_p around 0.02 in 16 steps while the other algorithms only able to reach low precisions which ϵ_p are more than 0.1. PA-QN reaches ϵ_p below 0.3 in 10 iterations and APA-APG2 reach it much slower and in 994 iterations while the other algorithms do not reach ϵ_p below 0.3 at all. PA-QN, Greedy PA-QN, APA-APG1, APA-APG2 and PA-APG reach ϵ_p below 0.9 in 2, 1, 2, 5 and 1 iterations respectively. Only PA-QN and Greedy PA-QN converge in very low number of steps and the other algorithms do not converge in 1000 iterations in term of ϵ_p . In conclusion, Greedy PA-QN and PA-QN are the quickest algorithms in terms of convergence time in both the optimization of $F(x_k)$ and the estimation of x_k and are the only algorithms converges in 1000 iteration for the low dimensional case. Moreover, PA-QN has the highest precision in the estimation of x_k which is much better than the other algorithms in 1000 iterations for the low dimensional case.

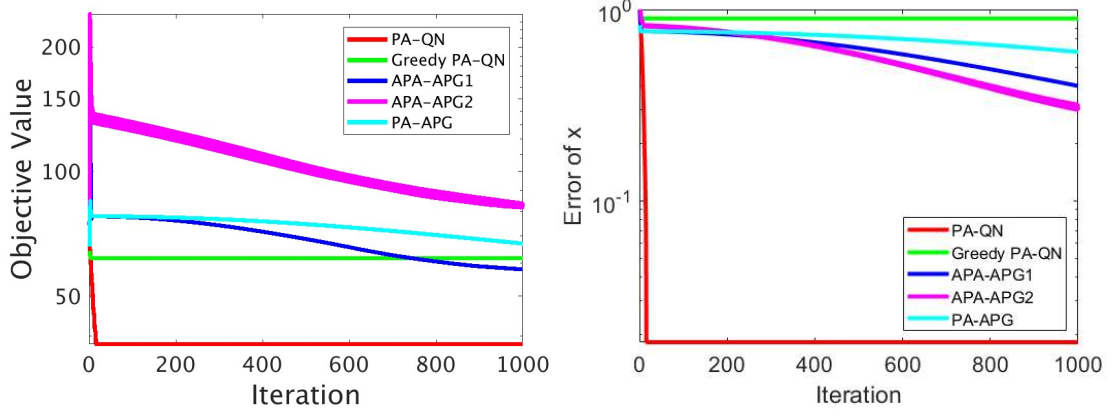
When medium dimensional dataset with $(m, n, k_1, k_2) = (100, 100, 5, 5)$ is used, Figure 3.3 and Table 3.5 show that PA-QN reaches $F(x_k)$ below 35, 30 and 25 quickest in 6, 7 and 11 iterations and converges quickest at around 10th iteration in $F(x_k)$ among all algorithms. PA-QN reaches $F(x_k)$ both 35 and 30 in 18 and 69 iterations and converges around at 100th iteration in term of $F(x_k)$. APA-APG1, APA-APG1 and PA-APG do not reach $F(x_k)$ below 35 at all in 1000 iterations and only stop at $F(x_k)$ above 70 since beginning. Figure 3.3 and Table 3.6 show that PA-QN reaches the highest precision (the lowest ϵ_p) among all algorithm in 1000 iterations. PA-QN reaches ϵ_p below 0.1, 0.09 and 0.02 in 7, 7 and 11 iterations. PA-QN converges at around 10th iteration. PA-QN is the quickest algorithm in reaching ϵ_p below 0.1 and 0.09 and in convergence in term of ϵ_p while Greedy PA-QN is the second quickest algorithm. Greedy PA-QN reaches ϵ_p below 0.1 and 0.09 in 20 and 53 iterations and converges around at 50th iteration in term of ϵ_p . APA-APG1, APA-APG2 and PA-APG only stop at ϵ_p around 1 since beginning. In conclusion, only PA-QN and Greedy PA-QN converge in both the optimization of $F(x_k)$ and the estimation of x_k and PA-QN has the highest precision in the estimation of x_k for the medium dimensional case.

When high dimensional dataset with $(m, n, k_1, k_2) = (1000, 100, 5, 50)$ is used, Figure 3.3 and Table 3.5 show that Greedy PA-QN reaches $F(x_k)$ below 200, 150 and 100 quickest in 1, 1 and 2 iterations and converges quickest in very few iterations in $F(x_k)$ among all algorithms. APA-APG1 and APA-APG1 reaches $F(x_k)$ below 200, 150 and 100 slightly slower than Greedy PA-QN in 3, 3 and 3 iterations and 3, 3 and 6 iterations respectively. PA-QN and PA-APG reaches $F(x_k)$ both 200, 150 and 100 in 10, 17 and 34 iterations and 39, 45 and 70 iterations. APA-APG1 converges in few iterations and PA-QN and PA-APG and APA-APG2 converge at around 100th iterations. Figure 3.3 and Table 3.6 show that Greedy PA-QN reaches the highest precision (the lowest ϵ_p) and much higher precision (the final ϵ_p is below 10^{-14} when the ones of all other algorithms are at most near to 10^{-9}) than the other algorithms in 1000 iterations. Greedy PA-QN reaches ϵ_p below 10^{-5} , 10^{-8} and 10^{-14} in 2, 116 and 408 iterations. Greedy PA-QN converges at around 400th iteration.

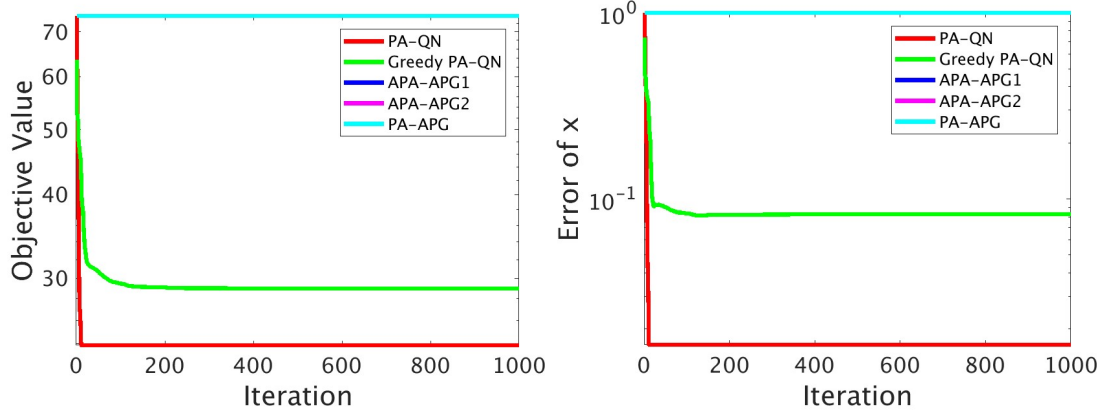
PA-QN is the second quickest algorithm in reaching ϵ_p below 10^{-8} while Greedy PA-QN is the quickest algorithm. PA-QN reaches ϵ_p below 10^{-5} in 112 iterations. PA-QN converges at around 200th iteration in term of ϵ_p . APA-APG1 is the second quickest algorithm in reaching ϵ_p below 10^{-5} while Greedy PA-QN is the quickest algorithm. PA-QN reaches ϵ_p below 10^{-8} in 279 iterations. Both APA-APG1 and APA-APG2 do not converges in 1000 iterations in term of ϵ_p . PA-QN reaches ϵ_p below 10^{-5} and 10^{-8} in 97 and 99 iterations while APA-APG2 reaches ϵ_p below 10^{-5} in 864 and does not reach ϵ_p below 10^{-8} in 1000 iterations. PA-APG converges at around 100th iteration in term of ϵ_p . The following summary shows the comparison among all algorithm in solving the three cases of eqn. (3.50) with $\alpha_1 = \frac{1}{1+10^5}$ and $\alpha_2 = \frac{10^5}{1+10^5}$.

- When a low dimensional dataset is used, both Greedy PA-QN and PA-QN converges quicker than all algorithm in term of $F(x_k)$ in the first case. PA-QN is the only one which has high final precision (the final ϵ_p is below 0.02 while ϵ_p of other algorithms are above 0.1).
- When a medium dimensional dataset is used, both PA-QN and Greedy PA-QN perform better and converge very quick in both $F(x_k)$ and ϵ_p . PA-QN reach particular high final precision (ϵ_p below 0.02). The other algorithm performs poorly since the beginning.
- When a high dimensional dataset is used, PA-QN and PA-APG converge faster than Greedy PA-QN but Greedy PA-QN have much higher final precision than PA-QN and PA-APG in the third case. It implies that Greedy PA-QN performs significant better for high dimension case (large m).

When $(m, n, k_1, k_2) = (50, 100, 5, 3)$



When $(m, n, k_1, k_2) = (100, 100, 5, 5)$



When $(m, n, k_1, k_2) = (1000, 100, 5, 50)$

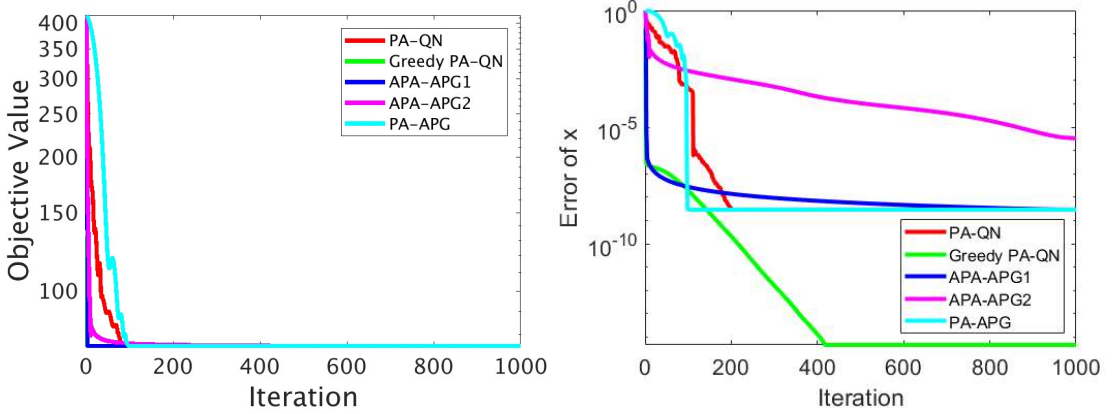


Figure 3.3 Objective function value $F(x_k)$ and error vs iteration k on sum of multiple sparsities with $\alpha_1 = \frac{1}{1+10^5}$ and $\alpha_2 = \frac{10^5}{1+10^5}$

Table 3.5 Objective function value $F(x_k)$ versus iteration number k on sum of multiple sparsities with $\alpha_1 = \frac{1}{1+10^5}$ and $\alpha_2 = \frac{10^5}{1+10^5}$

Cases (m, n, k_1, k_2)	(50, 100, 5, 3)	(100, 100, 5, 5)	(1000, 100, 5, 50)
$F(x_k)$	100 70 50	35 30 25	200 150 100
PA-QN	1 2 9	6 7 11	10 17 34
Greedy PA-QN	1 1 -	18 69 -	1 1 2
APA-APG1	1 455 -	- - -	3 3 3
APA-APG2	1 - -	- - -	3 3 6
PA-APG	1 1 -	- - -	39 45 70

Table 3.6 Precision ϵ_p versus iteration number k on sum of multiple sparsities with $\alpha_1 = \frac{1}{1+10^5}$ and $\alpha_2 = \frac{10^5}{1+10^5}$

Cases (m, n, k_1, k_2)	(50, 100, 5, 3)	(100, 100, 5, 5)	(1000, 100, 5, 50)
ϵ_p	0.9 0.3 0.02	0.1 0.09 0.02	10^{-5} 10^{-8} 10^{-14}
PA-QN	2 10 16	7 7 11	112 185 -
Greedy PA-QN	1 - -	20 53 -	2 116 408
APA-APG1	2 - -	- - -	3 279 -
APA-APG2	5 994 -	- - -	864 - -
PA-APG	1 - -	- - -	97 99 -

3.7. Summary

This chapter presented an approach for an optimization technique issue; namely: nonsmooth composite convex optimization, of sparse representation problem (see Figure 1.1 and Figure 1.2). A parallelable Jacobi iteration type algorithms which utilize both first and second-order information for optimization of a convex composite which contains multiple nonsmooth convex functions are presented. The proposed method is a combinations of the second-order quasi-newton method, multiple first-order methods and parallelable accelerated proximal average and backtracking techniques. The second-order method with BFGS and the first-order methods with the steepest gradient descent for L_2 and L_1 -norms and Nesterov's accelerated gradient descent methods are integrated into the proposed method. Using the proposed method, the convergence rate can be superlinear and can have a lower bound of $\mathcal{O}(\frac{1}{k^2})$. Two algorithms of the proposed method, PA-QN and Greedy PA-QN are introduced. Two problems of the sum of multiple sparsities with three cases each are used in the benchmarking. The first problem is a minimization with objective function which consists of two equally-weighted (non-biased) non-smooth functions. The

second problem's objective function consists of two non-equally-weighted non-smooth functions and one function is highly weighed (biased) than the other one. The experimental results show that the proposed algorithms, PA-QN and Greedy PA-QN, perform better (in term of minimization's objective function value and the precision of sparse coefficient vector estimation) than the other first-order state-of-the-art parallelable Jacobi iteration type methods, APA-APG [1], in overall. Greedy PA-QN performs significantly better in sparse coefficient vector estimation in large dimensional (large m) problems. In the next chapter, a theoretical work which shows the performance will be presented; namely: the lower bound of the successfully ground truth sparse representation and error recovery probability, of the recovery problem design using eqn. (1.2) (see Figure 1.1 and Figure 1.2). It is an extension of the previous works in [4]

Chapter 4

Representation Recovery via L_1 -norm Minimization with Corrupted Data

“Looking at the past must only be a means of understanding more clearly, what and who they are so that they can more wisely build the future.”

- Paulo Freire

In the previous chapter, a novel optimization technique for solving nonsmooth optimization problem which a lot sparse recovery problem involves is presented. In this chapter, we present the theoretical derivation of successful sparse recovery probability bound using an L_1 -norm minimization.

This chapter presents the performance, the successfully ground truth sparse representation and error recovery probability lower bound, of a recovery problem design (eqn. (1.2)), please refer to Figure 1.1 and Figure 1.2. It is an extension of the previous works in [4] which studies a problem of recovering a coefficient vector $x \in \mathbb{R}^n$ and an uncorrupted data Ax from a highly corrupted data y by assuming the equality, $y = Ax + e \in \mathbb{R}^m$

where

$A \in \mathbb{R}^{m \times n}$ is a dictionary, and

$e \in \mathbb{R}^m$ is an unknown error vector whose nonzero entries may be unbounded,

is valid. This is motivated by the face recognition problem in computer vision,

the works in [4] consider all atoms of dictionary A fulfilling a bouquet model⁴. It theoretically derives the bound of recovery probability (rp) from a sparse representation recovery using an L_1 -norm minimization (eqn. (1.1)) as follows.

$$\begin{aligned} \min_{x,e} & \|x\|_1 + \|e\|_1 \\ \text{s. t. } & y = Ax + e \end{aligned} \quad \begin{array}{l} \text{Recap} \\ (1.1) \end{array}$$

The L_1 -norm minimization (eqn. (1.1)) has been successfully applied in face recognition [3]. However, the assumption is only valid when all atoms of dictionary A can be represented by the same Gaussian vector such as in the face recognition and background subtraction with only one background. The assumption is inappropriate for problems in which atoms from different classes cannot be represented by only one Gaussian vector such as viewpoint classification⁵ [5], [6] and background subtraction with possible multiple backgrounds. In these works, a more generalized and more appropriate assumption is introduced. The atoms of the dictionary A are assumed to fulfil a multi-bouquet model⁶ instead of a bouquet model. Under the new assumption, each identity's faces and images of each viewpoint in face recognition and viewpoint classification can be represented by class-dependent Gaussian vectors (see the explanation of multi-bouquet model in section 4.2). Note that independent identically distributed (i.i.d.) Gaussian vectors which have same means and same standard deviations are used to represent all faces of all identities in the previous works in [4]. Moreover, the

⁴ All vectors fulfill a bouquet model when all vectors are highly correlated (and possible overcomplete) to each other and can be represented by Gaussian vectors which have the same means and the standard deviation. See section 4.2 for the definition of the bouquet model.

⁵ Viewpoint classification is the classification of viewpoints used in taking the input photos.

⁶ All vectors fulfill a multi-bouquet model when only vectors belong to the same class can be represented by (class-dependent) Gaussian vectors which have the same means and the standard deviations while vectors from different classes cannot be. See section 4.2 for the definition of the multi-bouquet model.

L_1 -norm minimization (eqn. (1.1)) is extended with the introduction of an additional trade-off parameter λ into the minimization as shown in eqn. (1.2) as follows:

$$\begin{aligned} \min_{x,e} & \|x\|_1 + \lambda \|e\|_1 \\ \text{s. t. } & y = Ax + e \end{aligned} \quad \begin{array}{l} \text{Recap} \\ (1.2) \end{array}$$

where λ is a tradeoff parameter.

λ can be tuned to improve the rp of the sparse representation and the original data recoveries. L_1 -norm minimization (eqn. (1.2)) has been successfully used for background subtraction [153] and viewpoint classification for anomaly detection [5], signal denoising and error correction [147]–[151], super-resolution reconstruction [152], [420], background subtraction [153] and dictionary learning [153]. Similar to the works in [4], the theoretical work presented has shown that the L_1 -norm minimization (eqn. (1.2)) can recover all representations with almost all sign-and -support patterns of e when m is large under the new assumption. The recovery can also be successful even when nearly 100% of measurements in y are corrupted. Simulations and experiments have empirically substantiated the theoretical derivations of the works.

Due to the flow in the derivation and objective of these works are the generalized version of the works in [4], similar notations and definitions are used in the following contents.

In this chapter, the notations are presented in Section 4.1. Section 4.2 introduces a multi bouquet model and the main assumptions. Section 4.3 presents the main results (successful recovery probabilistic bound). Section

4.4 elaborates the roadmap of the proof. Section 4.5 shows the optimal conditions to achieve successfully data recovery. Section 4.6. introduces an iterative construction of separator to achieve the optimal conditions presented in Section 4.6. Section 4.7 presents the probabilistic analysis of a potential candidate in achieving the optimal conditions and proves a difficult case of the main results. Section 4.8 finalises the proof of the main results. Section 4.9 presents the experimental results. Section 4.10 summarizes the chapter.

4.1. Notations

In this chapter, $x_o \in \mathbb{R}^n$, and $e_o \in \mathbb{R}^m$ are the annotations of the original/ground truth sparse coefficient vector and corrupting error vector such that x_o and e_o follow eqn. (4.1).

$$y = Ax_o + e_o = Ax^* + e^* \quad (4.1)$$

where x^* and e^* are the estimated sparse coefficient vector and corrupting error from L_1 -norm minimization in eqn. (1.2).

- For any $n, k \in \mathbb{Z}_+$,
 - $[n]$ denotes the set $\{1, \dots, n\}$.
 - $\binom{[n]}{k}$ denotes the set of subsets $S \subset [n]$ of size k .
 - $I = \text{supp}(x_o) \subset [n]$,
 - $J = \text{supp}(e_o) \subset [m]$,
 - $k_1 = |I|$ be the support size of the sparse coefficient vector x_o ,
 - $k_2 = |J|$ the support size of the error e_o ,
 - $\sigma_x = \text{sign}(x_o(I)) \in \{\pm 1\}^{k_1}$, and
 - $\sigma = \text{sign}(e_o(J)) \in \{\pm 1\}^{k_2}$.
- Let each element of x belongs to one of the N classes and N_i is the total number of elements which belongs to class i ,
 - $\tilde{x}_i \in \mathbb{R}^{N_i}$ is a vector which contains all x elements which belong to class i ,
 - x_j^i is the j^{th} element of $\tilde{x}_i$.
- For $M \in \mathbb{R}^{m \times n}$, $I, J \subseteq [n]$ and $K \subseteq [N_i]$

- $\tilde{M}_i$ will denote the i^{th} row of M ,
- M_J will denote the submatrix of M containing the rows indexed by J .
- $M_{I,J} \in \mathbb{R}^{|I| \times |J|}$ will denote the submatrix of M indexed by I and J .
- $M_i \in \mathbb{R}^{n \times N_i}$ denotes the submatrix of M which consists of all rows of M but only columns which belong to class i , where each column of M belongs to a class and N_i is also the total number of columns which belongs to class i ;
- $M_J^i \in \mathbb{R}^{|J| \times N_i}$ will denote the submatrix of M_i containing the rows indexed by J , and
- $M_{J,K}^i \in \mathbb{R}^{|J| \times |K|}$ will denote the submatrix of M_i indexed by J and K .
- To reduce confusion between the index set I and the identity matrix, I will be used to denote the latter.

Several norms on vectors and matrices will be used:

- For $x \in \mathbb{R}^m$,
 - $\|x\|_1 = \sum_i |x_i|$ is the L_1 -norm,
 - $\|x\|$ is the L_2 -norm.
- For matrices M ,
 - $\|M\| = \sigma_{\max}(M)$ denotes the L_2 operator norm,
 - $\|M\|_F = \sqrt{\text{tr}[M^* M]}$ denotes the Frobenius norm, and
 - $\|M\|_{2,\infty} = \max_i \|M_i\|$ denotes the $L_2 \rightarrow L_\infty$ operator norm.
- For an $m \times n$ matrix M with ordered singular values $\sigma_1 \geq \sigma_2 \geq \dots$,
 - $\sigma_{\min}(M)$ will be used to denote $\sigma_{\min\{m,n\}}(M)$, i.e., the smallest singular value.
- For a linear subspace $V \subseteq \mathbb{R}^n$,
 - $P_V \in \mathbb{R}^{n \times n}$ will denote the projection operator onto V .

The shrinkage operator S with threshold $\frac{1}{2}$, defined for scalars t as

$$S[t] = \text{sign}(t) \max \left\{ t - \frac{1}{2}, 0 \right\} \quad (4.2)$$

and extended to vectors by applying it elementwise.

4.2. Multi-bouquet Models and Main Assumptions

In this section, a multi-bouquet model is introduced. It is the central assumption which affects most of the proofs in this chapter. It is an extended model of the bouquet model which was introduced in [4]. Multi-bouquet model is better than the bouquet model in simulating most of the practical dictionaries. Bouquet model can only be used to explain the dictionaries which all atoms are highly similar. However, the characteristics of most of the dictionaries used in various problems do not follow this assumption. The characteristics of the multi-bouquet model are more similar to many dictionaries used in practical problem especially the dictionary which atoms are belongs to multi-classes and intra-class atoms are similar and intra-class atoms are very different. Two other assumptions on signs and support of x_o and e_o and proportional growth of some parameters are introduced in the later part of this section.

Assumption 1 (The Multi-bouquet Models)

Bouquet model is an assumption (**Assumption 1**) in [4] that the atoms (columns) of dictionary A are independent identically distributed (i.i.d.) samples from a Gaussian distribution such that $A = [\tilde{A}_1, \dots, \tilde{A}_n] \in \mathbb{R}^{m \times n}$, with A_i fulfills the eqn. (4.3)

$$\tilde{A}_i \sim_{i.i.d.} \mathcal{N}\left(\mu, \frac{\nu^2}{m} I_m\right), \|\mu\|_2 = 1, \|\mu\|_\infty \leq C_\mu m^{-1/2} \quad (4.3)$$

where

i	is the index of the atom,
$\mathcal{N}\left(\mu, \frac{\nu^2}{m} I_m\right)$	is a Gaussian vector with a mean of μ and a standard deviation of $\frac{\nu^2}{m} I_m$ whose ν is a parameter, m is the row number A , of and $I_m \in \mathbb{R}^{m \times m}$ is an identity matrix, and
C_μ	is a constant such that some constant $C_\mu \geq 1$.

More detailed information on the bouquet model and its geometrical interpretation can be found in [4].

In this work, an extension of the bouquet model; namely: multi-bouquet model, is proposed. The dictionary A is assumed to fulfil multi-bouquet model. *A dictionary A which fulfil multi-bouquet model is constructed using atoms belonging to multiple classes such that it follows the eqns (4.4)-(4.6):*

$$A = [B_1 | \dots | B_N] = [A_1, \dots, A_n] \in \mathbb{R}^{m \times n} \quad (4.4)$$

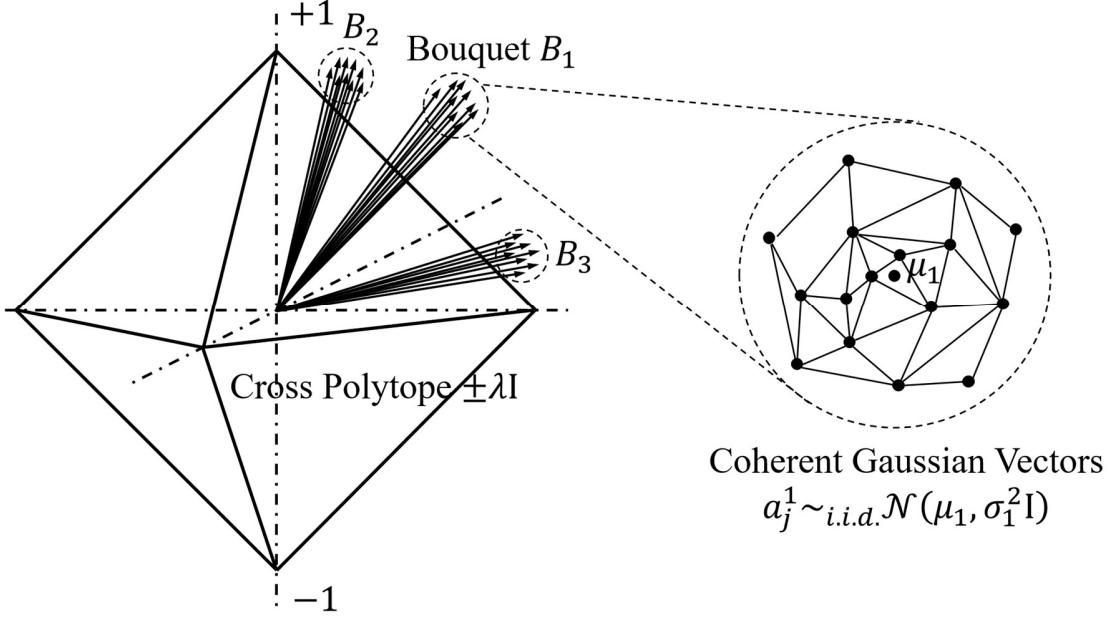
$$B_i = [a_1^i, \dots, a_{N_i}^i] \in \mathbb{R}^{m \times N_i} \quad (4.5)$$

$$a_j^i \sim \text{i.i.d.} \mathcal{N} \left(\mu_i, \frac{v_i^2}{m} I_m \right), \|\mu_i\|_2 = 1, \|\mu_i\|_\infty \leq C_\mu m^{-1/2} \quad (4.6)$$

where

- i is the index of a class,
- B_i is a submatrix of A which consists of all atoms, a_j^i , belong to class i and fulfil eqn. (4.6),
- N and N_i are the total number of classes and the total number of atoms belonging to Class i ,
- $\mathcal{N} \left(\mu_i, \frac{v_i^2}{m} I_m \right)$ is a Gaussian vector with a class-dependent mean of μ_i and a standard deviation of $\frac{v_i^2}{m} I_m$ in which v_i is a class-dependent parameter, m is the row number of A , of and $I_m \in \mathbb{R}^{m \times m}$ is an identity matrix, and
- C_μ is a constant such that for some constant $C_\mu \geq 1$.

Similar to the claims in [4], both the conditions on the variance, $\frac{v_i^2}{m} I_m$, and the mean, μ_i , of the distribution of each bouquet force those atoms that fulfill the same bouquet model to remain both tight and incoherent with a standard basis (of “cross”) when the dimension m increases [4]. An example of a multi bouquet model is shown in Figure 4.1. The left figure shows a 3-bouquet model and a cross polytope spanned by the matrix $\pm \lambda I$. The right figure shows the magnified version of the top of the bouquet B_1 ; it is a group of i.i.d. Gaussian vectors with a variance σ_1^2 and a common mean vector μ_1 .



Note that, the bouquet model of **Assumption 1** in [4] is a special case of the multi-bouquet model with $N = 1$.

Assumption 2 (Signs and Support of x_o and e_o)

This assumption is the same as **Assumption 2** in [4]. *The signs σ_x and support I of x_o are arbitrary. The support J of e_o is randomly and uniformly chosen, and the signs σ of e_o are random (i.i.d. symmetric ± 1 on J) too.*

In this work, the L_1 -norm minimization in eqn. (1.2) and dictionary A which fulfils the multi-bouquet model in **Assumption 1** are used. Similarly to [4], for x_o and e_o to be recoverable from the L_1 -norm minimization in eqn. (1.2), the vertices from each bouquet fulfilled by A needs to “see” through the cross polytope $\pm\lambda I$, where I is an identity matrix with the corresponding dimension, to the vertices from the corresponding “negative bouquet” which are nearly antipodal to the corresponding bouquet geometrically in $-A$. Note that, there is no tradeoff parameter λ in the problem (the problem fixes $\lambda = 1$) in [4]., Hence, the successful recovery probability of the problem in [4] cannot be maximized by adjusting the parameter λ . The novelty of using L_1 -norm minimization in eqn. (1.2) in data recovery rather than using the L_1 -norm minimization in eqn. (1.1) is that the parameter λ

in eqn. (1.2) can be optimized to suit the characteristics of the dataset and, hence the successful recovery probability (rp) can be maximized.

Similar to [4], the analysis will show that the impossible event occurs with probability approaching one as long as x_o is sufficiently sparse enough and the data has very high-dimensional spaces (m).

Assumption 3 (Proportional Growth)

Similar to **Assumption 3** in [4], *the sequence of signal-error problem exhibits proportional growth with parameter $\delta > 0$, $\rho \in (0,1)$, $\alpha > 0$ if*

$$n = \lfloor \delta m \rfloor, \|e_o\|_0 = \lfloor \rho m \rfloor, \|x_o\|_0 = \lfloor \alpha m \rfloor, \quad (4.7)$$

Furthermore, the support size of the error vector e_o can be any fraction $\rho < 1$ of the dimension m . The support size of the sparse coefficient vector x_o is only a very small fraction of m : $\|x_o\|_0 = \alpha m$ for some small constant $\alpha > 0$ (which may depend on δ and ρ). Similarly, the following theoretical derivation will show why the support density α of the sparse coefficient vector x must be very small when ρ is close to 1 (in fact, a small fraction of $1 - \rho$). It is reasonable to assume small support density α of the sparse coefficient vector x_o , as in many practical problems such as in face recognition and viewpoint classification, the support size of x_o is bounded by a constant and by the numbers of images per subject and viewpoint [3], [5], [6], [147].

Under **Assumptions 1–3**, the theorem in the next section can be formed.

4.3. The Main Result

In this section, a theorem which is the main result of all the theoretical proofs in this chapter and a remark which elaborates the critical constants used in the theorem and the functions used in expressing the constants are presented.

Theorem 1 (Sparse Representation Recovery with Multi-Bouquet Model)

Suppose that

- $\delta > 0$, $\rho < 1$ (δ and ρ are defined in **Assumption 3**),
- A is distributed according to eqns (4.4)-(4.6) with all λv_i (λ and v_i are defined in eqns (4.8) and (4.6) respectively) sufficiently small,
- $J \subset [m]$ (m is defined in eqn. (4.1)) is a uniform random subset of size ρm ,
- $\sigma \in \{\pm 1\}$ is i.i.d. symmetric (independent of J).

When

- m is sufficiently large $\lambda > \frac{16}{42-v'}$, $v'^4 \leq 2^{77}$,
- all x_o with $\|x_o\|_0 \leq \alpha_{\delta,\rho,v_{\max}}^* m$, any e_o with $\text{supp}(e_o) = J$, $\text{sign}(e_o(J)) = \sigma$, and
- $A_{J^c,I}$ has full rank k_1 where $k_1 = |I|$ and $\text{supp}(x_o) = I$,

the minimizer of eqn. (4.8) is uniquely defined with a probability of at least $1 - (35 + 5N) \exp(-\varepsilon_{\delta,\rho,v_{\max}}^* m)$.

$$\begin{aligned} (x_o, e_o) &= \argmin \|x\|_1 + \lambda \|e\|_1 \\ \text{s.t.} \quad Ax + e &= Ax_o + e_o \end{aligned} \tag{4.8}$$

where

- λ is a tradeoff parameter
- $A \in \mathbb{R}^{m \times n}$ is the dictionary which fulfils the multi-bouquet model described in **Assumption 1** in Section (4.2),
- $x_o \in \mathbb{R}^n$ is the original/ground truth sparse coefficient vector,
- $e_o \in \mathbb{R}^m$ is the original/ground truth corrupting error vector, and
- x_o and e_o fulfill eqn. (4.1),
- $v_{\max}$ is a constant such that $v_{\max} = \max_i v_i$,
- $v_{\min}$ is a constant such that $v_{\min} = \min_i v_i$, and
- v' is a constant equal to $\frac{v_{\max}}{v_{\min}}$,

By “all λv_i sufficiently small” and “ m sufficiently large” means that

- $0 < \lambda v_{\min} \leq \lambda v_{\max} \leq \lambda v_{\delta,\rho}^*$, and
- $m \geq m_{\delta,\rho,v_{\max},v_{\min}}^*$

where

$$v_{\delta,\rho}^* > 0,$$

$m_{\delta,\rho,v_{\max},v_{\min}}^* > 0$ are constant w.r.t. m , and

$v_{\delta,\rho}^*$, $\varepsilon_{\delta,\rho,v_{\max}}^*$, $\alpha_{\delta,\rho,v_{\max}}^*$ and $m_{\delta,\rho,v_{\max},v_{\min}}^*$ are positive constants [with respect to (w.r.t.) m], which depend on δ , ρ , $v_{\min}$ and $v_{\max}$ (see the definitions below).

Two constants which play important roles in the following proof are defined as follows,

$$\tau = \exp(-48^{-2}\delta^{-1}v_{\max}^{-2}) \quad (4.9)$$

$$C_s = Y[2^{-17}C_\mu^{-4}\delta^{-1}(1-\rho)^2] \quad (4.10)$$

where

$v_{\max}$ is a constant such that $v_{\max} = \max_i v_i$,

δ and ρ are defined in **Assumption 3** such that $\delta > 0$ and $\rho < 1$,

v_i and C_μ are bouquet properties which are defined in eqn. (4.6), and

Y is a function which is introduced in the following paragraph and eqn. (4.12).

A special function and its inverse function which are used for the definition of the constants used in **Theorem 1** and eqns (4.9) and (4.10) are defined as follows.

$$f(x) = x \log\left(\frac{1}{x}\right) \quad (4.11)$$

$$Y[t] = f^{-1}(t) \quad (4.12)$$

where $f: (0, \frac{1}{e}) \rightarrow (0, \frac{1}{e})$ and $Y: (0, \frac{1}{e}) \rightarrow (0, \frac{1}{e})$. Note that Y can be expressed in terms of the Lambert W function [421]. Both function f and Y are strictly increasing. As $\forall x \in (0, \frac{1}{e})$, $f(x) \geq x$

$$Y[t] \leq t \quad \forall t \in (0, \frac{1}{e}) \quad (4.13)$$

Furthermore, for any fixed a and $t < \frac{a}{e}$

$$x \log\left(\frac{a}{x}\right) \leq t \Leftrightarrow x \leq aY\left[\frac{t}{x}\right] \quad (4.14)$$

this property will be useful for computing the union bounds in the latter parts.

Moreover, the constants in **Theorem 1** are expressed as follows:

$$\left\{ \begin{array}{l} v_{\delta,\rho}^* = 2^{-12} \lambda^{-1} \sqrt{\frac{C_s}{\delta}} \\ \varepsilon_{\delta,\rho,v_{\max}}^* = \frac{\lambda^2 v_{\max}^2 (1-\rho) \tau^2}{2} \\ \alpha_{\delta,\rho,v_{\max}}^* = \lambda^2 v_{\max}^2 \Upsilon \left[\frac{\varepsilon^*}{\delta} \right] \\ m_{\delta,\rho,v_{\max},v_{\min}}^* = \max \left(\frac{2^{17}}{(1-\rho)C_s}, \frac{2^{16} \times 3}{\lambda^8 v_{\min}^8 \delta^2 (1-\rho)^3} \right) \end{array} \right. \quad (4.15)$$

where

$v_{\min}$ is a constant such that $v_{\min} = \min_i v_i$,

λ is a tradeoff parameter which is defined in eqns (1.2) and (4.8)

The subscripts of each constant are used to emphasize the dependency between the constants with the quantities related to the multi-bouquet $(\delta, v_{\max}, v_{\min})$ and the density of the error ρ . In the following proof, the subscripts are occasionally dropped for compactness.

E.g. $\alpha_{\delta,\rho,v_{\max}}^*$ and α^* are the same.

Similar to the claims in [4], all given constants are extremely small while m^* is extremely large. The best constants which is obtained from the following analysis can be pessimistic. Linear algebraic manipulations are applied to reduce the quantity of interest to a function of sparse singular values and standard concentration results are then used to argue that they are well behaved for all sparse vectors in the analysis. The previous and hence this analysis is inspired by the restricted isometry property (RIP) analysis of the standard compressed sensing setup. The analysis manages to derive the guarantees for sparse recovery that are order optimal in a simple and clean manner. However, it is important to emphasize and to be reminded that compared to those results obtained with tools from the geometric functional analysis [422] or combinatorics [423] in a finite- m setting, RIP results are pessimistic.

It should be noted again that if the support of x_o is sublinear [i.e., $\|x_o\|_0 = o(m)$ as defined in eqn. (4.7)] and m is sufficiently large, exact recovery of both x and e holds. It means

that, the asymptotically L_1 -norm minimization in eqns (1.2) and (4.8) can recover any sufficiently sparse coefficient vector x from almost any e as long as each bouquet is sufficiently tight. Similar to [4], it is worth to note that there is no assumption made on sparse coefficient vector x , except that its support is very small. Note that, the results in previous [4] and this work hold for all sparse coefficient vector x , and almost all errors e , in terms of signs and support. While the results in the other existing sparse representation literature typically either belongs to all sparse coefficient vectors x under the strong threshold of [68] or almost all sparse coefficient vectors x , in terms of the weak threshold of [68].

4.4. Roadmap of the Proof

In this section, the roadmap of the proof of **Theorem 1** is outlined. Two key lemmas, **Lemma 1** (in Section 4.5) and **Lemma 2** (in Section 4.6) are introduced in Section 4.5 and Section 4.6 respectively. **Lemma 1** shows that the existence of a certificate q under certain conditions guarantees the uniqueness and optimality of (x_o, e_o) in the L_1 -minimization in eqn. (4.8). **Lemma 2** which is borrowed from [4] shows that an iterative procedure can construct the certificate of optimality q when a set of sufficient conditions are fulfilled. **Lemmas 1** and **2** show that if a solution q_0 to an equation (eqn.(4.17)) in **Lemma 1** can be found and it satisfies a condition (eqn. (4.18)) in **Lemma 1** and two conditions (eqn. (4.20)<1 and (4.21)) in **Lemma 2**, given that x_o, e_o is also the optimal solution to the L_1 -norm minimization in eqn. (4.8) and the minimizer is uniquely defined.

In the subsequent parts, the probabilistic bound of a candidate of q_0 , the minimum L_2 -norm solution to the eqn. (C.60) in Appendix C, fulfilling the **Lemmas 1** and **2** for difficult cases are analysed in **Lemma 3** (in Section 4.7.1) and **Lemma 4** (in Section 4.7.2). Three separated bounds on three quantities guarantees that the conditions in **Lemma 2** will be fulfilled. Hence, a candidate of q_0 which corresponding three quantities fulfilling the three bounds guarantees that the certificate q in **Lemma 1** exists and the L_1 -minimization in eqn. (4.8) can successfully recover the (x_o, e_o) and (x_o, e_o) is also the unique minimizer. **Lemma 3** analyses the successful probabilistic bound of one of the three corresponding bounds of the above-mentioned candidate of q_0 and **Lemma 4** analyses the other two. In

both **Lemmas 3** and **4**, the assumptions of an underdetermined A and dense errors are assumed for the difficult cases. Finally, the lowest successful recovery probability bound for **Theorem 1** to be valid for the difficult cases is concluded by **Lemmas 3** and **4** when the candidate of q_0 is used (in Section 4.8).

In the final part, the analysis for the difficult cases is extended to cover all cases which fulfill the assumptions of **Theorem 1** and concludes **Theorem 1** (in Section 4.8).

Note that only the main **Lemmas 1** to **4** and the derivations of the lowest successful recovery probability bound for L_1 -minimization in eqn. (4.8) recovers the (x_o, e_o) in both difficult cases and all cases under the assumptions of **Theorem 1** are shown in the following sections. The detailed proofs of the Lemmas and the other supporting proofs are presented in the Appendix C.

4.5. Optimality Conditions

In this section, **Lemma 1** shows that the existence of a certificate q under certain conditions guarantees the uniqueness and optimality of (x_o, e_o) in the L_1 -minimization in eqn. (4.8) is presented. Before **Lemma 1** is elaborated, an expression of dictionary A which will assist in the understanding of the following contents is outlined.

The matrix A is distributed as in following eqn. (4.16).

$$\begin{aligned}
 A &= [B_1 | \dots | B_N] \\
 &= [a_1^1, \dots, a_{N_1}^1 | \dots | a_1^N, \dots, a_{N_N}^N] \\
 &= [\mu_1 + v_1 Z_1^1, \dots, \mu_1 + v_1 Z_{N_1}^1 | \dots | \mu_N + v_N Z_1^N, \dots, \mu_N + v_N Z_{N_N}^N] \\
 &= [\mu_1 1_1^* | \dots | \mu_N 1_N^*] + [v_1 Z_1 | \dots | v_N Z_N]
 \end{aligned} \tag{4.16}$$

where

v_i is a deviation amplifier,

$Z_i \in \mathbb{R}^{m \times N_i}$ is a matrix which has columns of i.i.d. $N\left(0, \frac{1}{m}\right)$ and $v_i Z_i$ represents the deviations of the all elements a_j^i of i^{th} bouquet which have the common mean μ_i and the deviation amplifier v_i , and

$1_i^* \in \mathbb{R}^{m \times N_i}$ is a matrix with all elements are 1.

The following lemma shows the optimality conditions for (x_o, e_o) in terms of properties of μ_i and submatrices of Z_i .

Lemma 1

Suppose that the original/ground truth sparse coefficient vector is

$$x_o = [\tilde{x}_1, \dots, \tilde{x}_t, \dots, \tilde{x}_N] \in \mathbb{R}^n$$

where

$$\tilde{x}_i = [x_1^i, \dots, x_{N_i}^i]^T \in \mathbb{R}^{N_i},$$

t^{th} class is the only class the elements in $\tilde{x}_t$ belongs to,

$\tilde{x}_t$ is the only $\tilde{x}_i$ containing the support of x_o ,

N is the total number of class,

N_i is the total number of elements of $\tilde{x}_i$,

the original/ground truth corrupting error vector is

$$e_o \in \mathbb{R}^m$$

and each bouquet mean is

$$\mu_i = [\mu_1^i, \dots, \mu_m^i]^T \in \mathbb{R}^m$$

(defined in eqn. (4.6)). Let $I = \text{supp}(x_o)$, $I_t = \text{supp}(\tilde{x}_t)$, the bouquet mean and deviation amplifier of class t are μ_t and v_t (defined in eqn. (4.6)), $\sigma_x = \text{sign}(x_o(I))$, $J = \text{supp}(e_o)$ and $\sigma = \text{sign}(e_o(J))$. If the following system of equations in eqn. (4.17) has a solution $q \in \mathbb{R}^{m+n-k_1-k_2}$ with $\|q\|_\infty < 1$:

$$\begin{cases} \begin{bmatrix} v_1(Z_{j^c}^1)^* & \frac{1}{\lambda} I \\ \vdots & \\ v_N(Z_{j^c}^N)^* & \\ v_t(Z_{j^c, I_t}^t)^* & 0 \end{bmatrix} q = \begin{bmatrix} -v_1(Z_j^1)^* \sigma \\ \vdots \\ -v_N(Z_j^N)^* \sigma \\ \frac{1}{\lambda} \sigma_x - v_t(Z_{j, I_t}^t)^* \sigma \end{bmatrix} \\ \langle \mu_{j^c}^1, q_{1, \dots, m-k_2} \rangle = -\langle \mu_j^1, \sigma \rangle \\ \vdots \\ \langle \mu_{j^c}^N, q_{1, \dots, m-k_2} \rangle = -\langle \mu_j^N, \sigma \rangle \end{cases} \quad (4.17)$$

where k_1 and k_2 are defined in Section 4.1 and the eqn. (4.18) is fulfilled,

$$A_{J^c, I} \text{ has full rank } k_1 \quad (4.18)$$

then, (x_o, e_o) is the unique optimal solution to the following L_1 -norm minimization (eqn. (4.19)),

$$\begin{aligned} \min_{x, e} & \|x\|_1 + \lambda \|e\|_1 \\ \text{s. t.} \quad & Ax + e = Ax_o + e_o \end{aligned} \quad (4.19)$$

where λ is a tradeoff parameter.

The proof of the **Lemma 1** is presented in Section C.3.1 in Appendix C.

Note that the eqn. (C.60) in the proof of **Lemma 1** in the Appendix C is equivalent to eqn. (4.17). Eqn. (C.60) will be used in following lemmas which will be used for proving **Theorem 1** in the latter parts.

In general, eqn. (4.17) has many solutions. Only one solution of q^* to eqn. (4.17) with L_∞ -norm smaller than 1 is needed to certify the optimality. In the later part, a close guess at such a certificate q using least squares can be constructed is shown as Φ (defined in (C.59)) is very likely to be well-conditioned. An iterative method of separator construction which is proposed in Section II.C in [4] shows the sufficiency of such a guess in the next section (Section 4.6).

4.6. Iterative Construction of Separator

In this Section, **Lemma 2** which shows that an iterative procedure can construct the certificate of optimality q when a set of sufficient conditions are fulfilled is presented. The lemma is borrowed from Section II.C in [4]. The lemma allows the requirements on q to be relaxed and hence a solution of small L_2 -norm that does not too severely violate the L_∞ constraint is all that is needed rather than directly looking for a solution with $\|q\|_\infty < 1$. The relaxed solution suffices to construct a solution with $\|S[q]\|_2$ sufficiently small, where S is the soft-thresholding operator with threshold $\frac{1}{2}$, defined in eqn. (4.2). For simplicity, $S[q]$ is often written as Sq .

Lemma 2 (Lemma 2 in [4])

For an underdetermined system of equations $\Phi q = w$, where Φ is defined in eqn. (C.59). P_Φ and $\xi_s(\Phi)$ denote the projection operator onto the range(Φ^*) and the norm P_Φ is restricted to sparse vectors as shown in eqn. (4.20) respectively.

$$\xi_s(\Phi) = \sup_{\substack{\|x\|_0 \leq s, \\ \|x\|_2 \leq 1}} \|P_\Phi x\|_2 \quad (4.20)$$

If $\xi_s(\Phi) < 1$ and there is a solution q_0 with $\Phi q_0 = w$ satisfying eqn. (4.21), then a solution q^* satisfying $\|q^*\|_\infty \leq \frac{1}{2}$ exists.

$$\|q_0\| + \frac{\xi_s(\Phi)}{1 - \xi_s(\Phi)} \|Sq_0\| \leq \frac{\sqrt{s}}{2} \quad (4.21)$$

In the following section, the probability bounds of the three bounds of the following quantities

- the restricted operator norm $\xi_s(\Phi)$,
- the norm of the initial guess $\|q_0\|_2$, and
- the norm of the initial violations $\|Sq_0\|_2$

in eqn. (4.21) when a candidate q_0 is applied are shown. These three bounds guarantee that eqn. (4.21) and; hence the candidate q_0 is one of the solutions of eqn. (4.17) in **Lemma 1**.

4.7. Probabilistic Analysis and Final Proof

Similar to [4], **Lemmas 1** and **2** have shown that if a solution q_0 to eqn. (4.17) can be found and it satisfies eqn. (4.21) [and also further verify that eqn. (4.18) holds], for a given x_o , e_o is also the optimal solution to the L_1 -norm minimization eqn. (4.19). A candidate, the minimum L_2 -norm solution to eqn. (4.17), will be shown to satisfy eqn. (4.21) with vast probability. The quantities in eqn. (4.21) are:

- the restricted operator norm $\xi_s(\Phi)$,
- the norm of the initial guess $\|q_0\|_2$, and
- the norm of the initial violations $\|Sq_0\|_2$

have to be bounded. In this section, the probabilistic bounds which hold simultaneously on these three quantities for sufficiently large m with vast probability over all sufficiently sparse signals x_0 are presented (Section 4.7.1 for quantity 1 and Section 4.7.2 for quantities 2 and 3). The probabilistic bounds of the bounds of the three quantities will be used for deriving the lowest successful rp of x_0 and e_o using the L_1 -norm minimization in eqn. (4.19) and; hence, for proving **Theorem 1**.

The tools and the supporting proofs for the probability analyses are presented in Section C.1 in Appendix C.

4.7.1. Projection of Sparse Vectors

In this subsection, **Lemma 3** which derives the probability bound (eqn. (4.24)) of a bound of $\xi_s(\Phi)$ is presented (see eqn. (4.21) and Φ is defined in eqn. (C.59)).

Similar to [4], the sparse vectors cannot be too aligned with the range of Φ^* is shown in this section. In **Lemma 3** (and also the lemmas in Subsection 4.7.2), the following assumptions which may make the sparse representation and uncorrupted data recovery problem more difficult are used.

- $\delta \geq 1$ is assumed; i.e., the dictionary A is either a square or a wide matrix.
- Large fraction of errors is assumed ($\rho \geq \frac{15}{16}$).

where

δ and ρ are defined in eqn. (4.7), and

A is defined in **Assumption 1**.

Regarding the proof of **Theorem 1** (Subsection 4.8), the result of **Theorem 1** for this difficult case with above-mentioned assumptions on δ and ρ is proven first, and then the result of **Theorem 1** for arbitrary $\delta > 0$, $\rho < 1$ is shown thereafter.

Lemma 3

Suppose for any $\delta \geq 1$, $\rho \geq \frac{15}{16}$, $v_{\max} \leq v^*$, $m \leq m^*$, any support $I \subset [n]$ of size is bounded by the inequality in eqn. (4.22), the bound of L_0 -norm of x , s , follow eqn. (4.23) and $C_s > 0$ is given in (4.10), and does not depend on m or v_i .

$$|I| \leq \alpha^* m \quad (4.22)$$

$$s = C_s m \quad (4.23)$$

where

δ and ρ are defined in **Assumption 3**,
 $v_{\max}$ is defined in eqn. (4.9),
 m , and n are defined in eqn. (4.1),
 v^* , m^* and α^* are the short form of $v_{\delta, \rho}^*$, $m_{\rho, \delta, v_{\max}, v_{\min}}^*$ and $\alpha_{\rho, \delta, v_{\max}}^*$ and are defined in eqn. (4.15),
 $v_{\min}$ is defined in eqn. (4.15)
 I is defined in Section 4.1,
 C_s is defined in eqn. (4.10), and
 s is the bound on the number of elements of $\|Sq\|$ such that $\|Sq\|_0 \leq s$.

Then, eqn. (4.24) is satisfied with probability at least $1 - (8 + N)\exp(-2\varepsilon^* m)$.

$$1 - \xi_s(\Phi) \geq \frac{\lambda^4 v_{\min}^4 \delta (1 - \rho)}{512} \quad (4.24)$$

where

ε^* is the short form of $\varepsilon_{\delta, \rho, v_{\max}}^*$ and is defined in eqn. (4.15),
 Φ is defined in eqn. (C.59),
 $\xi_s(\Phi)$ is defined in eqn. (4.20),
 A and N are defined in **Assumption 1**,
 J is defined in Section 4.1, and

In order to avoid confusion, it is note that the quantity s in eqn. (4.23) in **Lemma 3** represents the bound on the number of elements of $\|Sq\|$ (i.e., after applying soft-thresholding operator on q), and not the sparsity of the sparse coefficient vector x .

The detailed proof of **Lemma 3** is presented in Section C.3.2 in Appendix C.

4.7.2. Initial Certificate

In this section, the minimum L_2 -norm solution to the eqn. (C.60) is used as a candidate q_0 and is analyzed for the certificate of optimality. $\|q_0\|_2$ and $\|Sq_0\|_2$ (the operator S is defined in eqn. (4.2)) are upper bounded. q_0 can be refined to a q^* which satisfies $\|q^*\|_\infty \leq \frac{1}{2}$ with these two bounds (see **Lemma 2** for the definition of q^*) and the bound in eqn. (4.24) which is the proven result of **Lemma 3**. Hence, the two bounds contribute to the second half of the conditions required in the **Lemma 2**.

Lemma 4

Suppose any $\delta \geq 1$, $\rho < 1$, $v_{\max} \leq v^*$, $m \geq m^*$ and any subset of size $I \subset [n]$ of size

$$|I| \leq \alpha^* m \quad (4.25)$$

where

m and n	are defined in Assumption 1 ,
δ and ρ	are defined in Assumption 3 ,
$v_{\max}$	is defined in eqn. (4.9),
v^* , m^* and α^*	are the short forms of $v_{\delta,\rho}^*$, $m_{\delta,\rho,v_{\max},v_{\min}}^*$ and $\alpha_{\delta,\rho,v_{\max}}^*$ and are defined in eqn. (4.15), and
I	is defined in Section 4.1.

Eqns (4.26) and (4.27) are satisfied by the minimum L_2 -norm solution q_0 to the system of eqn. (C.60) simultaneously for all x_o with $\text{supp}(x_o) \subseteq I$, giving a probability at least $1 - (30 + 4N)\exp(-2\varepsilon_{\delta,\rho,v_{\max}}^* m)$ in Z, J, σ .

$$\|q_0\|_2 \leq (128 + 176\lambda + 8\lambda^2 v') v_{\max} \sqrt{\delta m} + O(1) \quad (4.26)$$

$$\|Sq_0\|_2 \leq (128 + 176\lambda + 8\lambda^2 v') v_{\max} \sqrt{\delta m} \exp(-C\delta^{-1} v_{\max}^{-2}) + O(1) \quad (4.27)$$

where

- N is defined in **Assumption 1**,
 ε^* is the short form of $\varepsilon_{\delta, \rho, v_{\max}}^*$ and is defined in eqn. (4.15),
 λ is defined in eqn. (4.8),
 $v_{\min}$ is defined in eqn. (4.15),
 v' is a constant equal to $\frac{v_{\max}}{v_{\min}}$, and
 C is a numerical constant. In particular, the result holds $C = \frac{1}{48^2}$. With these choices, the $O(1)$ term is upper bounded by $48(1 - \rho)^{-\frac{1}{2}}$.

The detailed proof of **Lemma 4** is presented in Section C.3.3 in Appendix C.

4.8. Proof of Theorem 1

Firstly, the result with the assumption that the matrix is wide ($\delta \geq 1$), the error e_o is not too sparse ($\rho \geq \frac{15}{16}$) and the ratio of the largest and the smallest not extremely too large ($v'^4 \leq 2^{77}$) is proven in Section 4.7.2. Lastly, the result implied by this more difficult case for arbitrary δ, ρ will be shown at the later part of this section.

Note:

- δ and ρ are defined in **Assumption 3**,
 e_o is defined in eqn. (4.1),
 $v_{\max}$ is defined in eqn. (4.9),
 $v_{\min}$ is defined in eqn. (4.15), and
 v' is a constant equal to $\frac{v_{\max}}{v_{\min}}$.

Underdetermined A (dictionary defined in Assumption 1), dense errors e_o (defined in eqn. (4.1)).

Let any $\delta \geq 1$, $\rho \geq \frac{15}{16}$, $\lambda > \frac{16}{42-v'}$, $v'^4 \leq 2^{77}$, $v_{\min} \leq v_{\max} \leq v^*$, $m \geq m^*$, $k_1 = \lfloor \alpha^* m \rfloor$, $C_s > 0$ according to eqn. (4.10) and $s = \lfloor C_s m \rfloor$ in where

- δ and ρ are defined in **Assumption 3**,
 λ is defined in eqn. (4.8),
 $v_{\max}$ is defined in eqn. (4.9),

v_{min}	is defined in eqn. (4.15),
v'	is a constant equal to $\frac{v_{max}}{v_{min}}$,
v^*, α^* and m^*	are short forms of $v_{\delta, \rho}^*$, $\alpha_{\delta, \rho, v_{max}}^*$ and $m_{\delta, \rho, v_{max}, v_{min}}^*$ and are specified in eqn. (4.15) , and
m	is defined in Assumption 1 ,

Note that, the setting above is consistent with the assumptions and the conclusions of **Lemmas 3** and **4**.

For each $I \in \binom{[n]}{k_1}$, let $\mathcal{E}_I$ be the event that:

- **Lemma 3** shows the bound of the matrix Φ has $\xi_s(\Phi)$;
- **Lemma 4** bounds the $\|q_0\|$ and $\|Sq_0\|$ in where q_0 is the the minimum L_2 -norm solution of eqn. (C.60) for all x_o with $supp(x_o) \subseteq I$
- $\text{rank}(A_{J^c, I}) = k_1$

in where

n	is defined in Assumption 1 ,
x_o	is defined in eqn. (4.1),
Φ	is defined in eqn. (C.59),
$\mathcal{E}_\Phi$	denotes the event in eqn. (C.101), and
S	is defined in eqn. (4.2).

Hence, $A_{J^c, I}$ has full rank k_1 with probability one. By **Lemmas 3** and **4**, the probability bound in eqn. (4.28) holds.

$$\forall I \in \binom{[n]}{k_1}, \mathbb{P}[\mathcal{E}_I^c] \leq (38 + 5N) \exp(-2\varepsilon^* m) \quad (4.28)$$

where

$A_{J^c, I}$	is defined in Lemma 1 ,
k_1	is defined in Section 4.1,
ε^*	is the short forms of $\varepsilon_{\delta, \rho, v_{max}}^*$ and is defined in eqn. (4.15), and
m and N	are defined in Assumption 1 .

On $\mathcal{E}_I$, for all x_o supported on I , q_o satisfies eqn. (4.29).

$$\begin{aligned} & \|q_o\| + \frac{\|Sq_o\|}{1 - \xi_s(\Phi)} \\ & \leq (128 + 176\lambda + 8\lambda v')v_{\max}\sqrt{\delta m} \left(1 + \frac{512\tau}{\lambda^4 v_{\min}^4 \delta(1 - \rho)}\right) \\ & + \frac{48}{\sqrt{1 - \rho}} \left(1 + \frac{512}{\lambda^4 v_{\min}^4 \delta(1 - \rho)}\right) \end{aligned} \quad (4.29)$$

where

x_o	is defined in eqn. (4.1),
q_o	is defined in Section 4.7.2,
S	is defined in eqn. (4.2),
$\xi_s(\Phi)$	is defined in eqn. (4.20),
λ	is defined in eqn. (4.8),
δ and ρ	are defined in Assumption 3 ,
$v_{\max}$ and τ	are defined in eqn. (4.9),
$v_{\min}$	is defined in eqn. (4.15), and
v'	is a constant equal to $\frac{v_{\max}}{v_{\min}}$.

Three bounds in eqns (4.30)-(4.32) are showed below for any $v_{\min} \leq v_{\max} \leq v^*$

$$\text{(Bound 1)} \quad \frac{512\tau}{\lambda^4 v_{\min}^4 \delta(1 - \rho)} \leq 1 \quad (4.30)$$

$$\text{(Bound 2)} \quad 2 \cdot (128 + 176\lambda + 8\lambda v')v_{\max}\sqrt{\delta m} < \frac{\sqrt{C_s m}}{4} \quad (4.31)$$

$$\text{(Bound 3)} \quad \frac{48}{\sqrt{1 - \rho}} \left(1 + \frac{512}{\lambda^4 v_{\min}^4 \delta(1 - \rho)}\right) < \frac{\sqrt{C_s m}}{4} \quad (4.32)$$

With the three bounds in eqns (4.30)-(4.32) above, on $\mathcal{E}_I$, for all x_o supported on I

$$\|q_o\| + \frac{\|Sq_o\|}{1 - \xi_s(\Phi)} < \frac{\sqrt{C_s m}}{2} = \frac{\sqrt{s}}{2} \quad (4.33)$$

Hence, there is a solution q^* to eqn. (4.17) with $\|q^*\|_\infty \leq \frac{1}{2} < 1$ by **Lemma 2**. **Lemma 1** then implies that on $\mathcal{E}_I$, for all with x_o is the unique solution to eqn. (4.19)

Furthermore, by using α^* in eqn. (4.15), the following bound in eqn. (4.16) holds. By eqn. (4.14), the bound in eqn. (4.35) and the probability bound in eqn. (4.36) hold.

$$\alpha^* < \Upsilon \left\lceil \frac{\varepsilon^*}{\delta} \right\rceil \leq \delta \cdot \Upsilon \left\lceil \frac{\varepsilon^*}{\delta} \right\rceil \quad (4.34)$$

$$\alpha^* \log \left(\frac{\delta}{\alpha^*} \right) \leq \varepsilon^* \quad (4.35)$$

$$\begin{aligned} \mathbb{P}[\cap_I \mathcal{E}_I] &\geq 1 - \sum_I \mathbb{P}[\mathcal{E}_I^c] \\ &\geq 1 - \# \binom{[n]}{k_1} \times (38 + 5N) \exp(-2\varepsilon^* m) \\ &\geq 1 - (38 + 5N) \exp \left(\alpha^* \log \left(\frac{\delta}{\alpha^*} \right) m - 2\varepsilon^* m \right) \\ &\geq 1 - (38 + 5N) \exp(-\varepsilon^* m) \end{aligned} \quad (4.36)$$

where $\# \binom{[n]}{k_1}$ denotes total number of possible combinations of $\binom{[n]}{k_1}$.

Lastly, only eqns (4.30)-(4.32) need to be proven to establish the main result for $\delta \geq 1$ and $\rho \in \left[\frac{15}{16}, 1 \right)$.

(1). (Bound 1) in eqn. (4.30)

(Bound 1) is established by proving an intermediate bound in eqn. (4.37) below holds whenever $v_{\min} \leq v_{\max} \leq v^*$ in which v^* is the short form of $v_{\delta, \rho}^*$ and is as specified in eqn. (4.15).

$$e^8 \tau \leq \lambda^8 v_{\max}^8 \quad (4.37)$$

where

$v_{\max}$ and τ are defined in eqn. (4.9), and
 λ is defined in eqn. (4.8).

By eqn. (4.9), eqn. (4.38) forms.

$$\tau^{\frac{1}{8}} = \exp(-8^{-1} 48^{-2} \delta^{-1} v_{\max}^{-2}) \quad (4.38)$$

By eqn. (4.15), the bound in eqn. (4.39) holds for $v_{max} \leq v^*$.

$$\lambda v_{max} \leq 2^{-12} \sqrt{\frac{C_s}{\delta}} < 2^{-20} \delta^{-1} < 8^{-1} 48^{-2} \delta^{-1} \quad (4.39)$$

where

δ is defined in **Assumption 3**, and

C_s is defined in eqn. (4.10).

and the bound in eqn. (4.40) is used (by the fact in eqn. (4.13) that $Y[t] \leq t$) in the second inequality.

$$C_s \leq 2^{-17} C_\mu^{-2} \delta^{-2} (1 - \rho) \ll 2^{-16} \delta^{-1} \quad (4.40)$$

where

C_μ is defined in **Assumption 1**, and

ρ is defined in **Assumption 3**.

Hence eqns (4.41) and (4.48) proving (4.37).

$$\tau^{\frac{1}{8}} < \exp(-\lambda^{-1} v_{max}^{-1}) \quad (4.41)$$

$$\lambda^{-1} v_{max}^{-1} \tau^{\frac{1}{8}} < \lambda^{-1} v_{max}^{-1} \exp(-\lambda^{-1} v_{max}^{-1}) \leq \frac{1}{e} \quad (4.42)$$

Above, eqn. (4.51) has been used.

$$\max_x x e^{-x} = e^{-1} \quad (4.43)$$

By eqn. (4.37), the following bound in eqn. (4.44) which is equivalent to eqn. (4.30) holds.

$$\begin{aligned} \frac{512\tau}{\lambda^4 v_{min}^4 \delta (1 - \rho)} &\leq \frac{512\lambda^4 v_{max}^4}{e^8 \delta (1 - \rho)} v'^4 \\ &\leq \frac{C_s^2}{e^8 2^{39} \delta^3 (1 - \rho)} v'^4 \\ &\leq \frac{(1 - \rho)}{e^8 2^{73} C_\mu^4 \delta^7} v'^4 \\ &\leq \frac{1}{e^8 2^{77} C_\mu^4 \delta^7} v'^4 \\ &\leq \frac{1}{e^8 C_\mu^4 \delta^7} \ll 1 \end{aligned} \quad (4.44)$$

where $v_{max} \leq v^*$, $C_s \leq 2^{-17} C_\mu^{-2} \delta^{-2} (1 - \rho)$, $\rho \geq \frac{15}{16}$, $v'^4 \leq 2^{77}$ are used in the second, third, fourth and fifth inequalities respectively in where v' is a constant equal to $\frac{v_{max}}{v_{min}}$.

(2). (Bound 2) in eqn. (4.31)

To establish (Bound 2), the condition which makes eqn. (4.31) hold is analyzed. If the following condition in eqn. (4.45) holds, eqn. (4.31) holds where $v \leq v^*$ and eqn. (4.15) is used in where

$$\frac{2^{-11}}{\lambda} \cdot (128 + 176\lambda + 8\lambda v') \sqrt{C_s m} < \frac{\sqrt{C_s m}}{4} \quad (4.45)$$

where

- λ is defined in eqn. (4.8),
- v_{max} is defined in eqn. (4.9),
- v_{min} is defined in eqn. (4.15),
- v' is a constant equal to $\frac{v_{max}}{v_{min}}$,
- C_s is defined in eqn. (4.10), and
- m is defined in **Assumption 1**.

By rearranging eqn. (4.45), the following simplified inequality in eqn. forms.

$$v' \lambda - 42\lambda + 16 < 0 \quad (4.46)$$

Hence, when eqn. (4.53) holds, eqn. (4.31) holds.

$$\lambda > \frac{16}{42 - v'} \quad (4.47)$$

(3). (Bound 3) in eqn. (4.32)

Before establishing eqn. (4.32), the following inequality in eqn. (4.48) is being proven.

$$\frac{512}{\lambda^4 v_{min}^4 \delta (1 - \rho)} > 1 \quad (4.48)$$

where

- λ is defined in eqn. (4.8),
- v_{min} is defined in eqn. (4.15), and
- δ and ρ are defined in eqn. (4.7)

For $v_{max} \leq v^*$, the following bound in eqn. (4.49) holds.

$$\begin{aligned}
 \frac{512}{\lambda^4 v_{min}^4 \delta (1 - \rho)} &> \frac{512 \delta}{2^{-48} C_s^2 (1 - \rho)} \\
 &> \frac{2^{91} C_\mu^8 \delta^3}{(1 - \rho)^3} \\
 &> 2^{103} C_\mu^8 \delta^3 \gg 1
 \end{aligned} \tag{4.49}$$

where

C_s is defined in eqn. (4.10), and

C_μ is defined in **Assumption 1**.

and $C_s \leq 2^{-17} C_\mu^{-4} \delta^{-1} (1 - \rho)^2$ (in third inequality), $\rho \in \left[\frac{15}{16}, 1\right)$ (in fourth inequality), $C_\mu \geq 1$ (in fifth inequality) and $\delta \geq 1$ (in fifth inequality) are used. Hence, eqn. (4.48) is established.

By eqn. (4.48), the following bound in eqn. (4.50) holds.

$$2 \cdot \frac{512}{\lambda^4 v_{min}^4 \delta (1 - \rho)} > 1 + \frac{512}{\lambda^4 v_{min}^4 \delta (1 - \rho)} \tag{4.50}$$

Eqn. (4.51) holds for $m \geq m^*$ with eqn. (4.15).

$$\frac{\sqrt{C_s m}}{4} \geq \frac{1}{4} \sqrt{\frac{2^{32} \times 3^2}{\lambda^8 v_{min}^8 \delta^2 (1 - \rho)^3}} = 2 \cdot \frac{48}{\sqrt{1 - \rho}} \cdot \frac{512}{\lambda^4 v_{min}^4 \delta (1 - \rho)} \tag{4.51}$$

Eqn. (4.32) is established when the bounds in eqns (4.50) and (4.51) hold.

By eqn. (4.30)-(4.32), **Theorem 1** is established for $\delta \geq 1$, $\rho \in \left[\frac{15}{16}, 1\right)$, $\lambda > \frac{16}{42 - v'}$ and $v'^4 \leq 2^{77}$ where

δ and ρ are defined in eqn. (4.7),

λ is defined in eqn. (4.8), and

v' is a constant equal to $\frac{v_{max}}{v_{min}}$.

Extension to arbitrary

$\delta > 0, \rho \in (0,1)$ in where δ and ρ are defined in eqn. (4.7). Let any $\delta > 0$. This extension mimics the extension for the bouquet model in the previous work [4].

Replace δ in all the above choice of constants with $\max(\delta, 1)$. According to the multi-bouquet model, a dictionary in eqn. (4.52) is formed.

$$\tilde{A} = [\mu_1 1_1^* | \dots | \mu_N 1_N^* | \mu_k \tilde{1}^*] + [v_1 Z_1 | \dots | v_N Z_N | v_k \tilde{Z}] \in \mathbb{R}^{m \times m} \quad (4.52)$$

where

μ_i	is defined in Assumption 1 ,
v_i	is a deviation amplifier and is defined in Assumption 1 ,
k	is an arbitrary number between 1 to N
Z_i and 1_i^*	are defined in eqn. (4.16), and
$\tilde{Z} \in \mathbb{R}^{m \times (m - \delta m)}$	is a matrix which has elements i.i.d. $N\left(0, \frac{1}{m}\right)$, and
$\tilde{1}^* \in \mathbb{R}^{m \times (m - \delta m)}$	is a matrix with all elements are 1.

With vast probability in $\tilde{Z}, J, \sigma$ for any e_o with signs and support J, σ and all sufficiently sparse $\tilde{x}_o$, the minimizer in eqn. (4.53) is uniquely defined.

$$\begin{aligned} (\tilde{x}_o, e_o) &= \operatorname{argmin} \|\tilde{x}\|_1 + \lambda \|e\|_1 \\ \text{s.t. } \tilde{A}\tilde{x} + e &= \tilde{A}\tilde{x}_o + e_o \end{aligned} \quad (4.53)$$

where e_o is defined in eqn. (4.1).

Getting $A \in \mathbb{R}^{m \times \delta m}$ by choosing the first δm columns of $\tilde{A}$. Obviously, A is distributed according to the multi-bouquet model defined in **Assumption 1**. **Lemma 12** (in Appendix C) presents that if $[x_o^*, 0 \dots 0]^*$, e_o can be exactly recovered by (4.53), and (x_o, e_o) is the unique optimal solution to eqn. (4.54) for any $x_o \in \mathbb{R}^{\delta m}$ which is defined in eqn. (4.1). Hence, with at least the same large probability, all sparse x_o are recovered by L_1 -minimization in the reduced system $Ax + e = y$ from a random error e_o .

$$\min \|x\|_1 + \lambda \|e\|_1 \text{ s.t. } Ax + e = \tilde{A}\tilde{x}_o + e_o \quad (4.54)$$

Lastly, **Lemma 13** (in Appendix C) directly shows that if L_1 -minimization uniquely recovers $(x_o, \tilde{e}_o)$, then for any e_o with $J = \text{supp}(e_o) \subseteq \text{supp}(\tilde{e}_o)$ and $\text{sign}(e_o(J)) = \text{sign}(\tilde{e}_o(J))$, then L_1 -minimization uniquely recovers (x_o, e_o) . Let any $\rho \in (0,1)$. The above choice of constants is chosen, but when ρ is replaced by $\max\{\rho, \frac{15}{16}\}$. For $\rho < \frac{15}{16}$, the signs and support are generated as follows: generate a random subset $\tilde{J} \subset [m]$ of size $\frac{15m}{16}$, and a random sign pattern $\tilde{\sigma}$ with $\tilde{\sigma}(\tilde{J}^c) = 0$, and i.i.d. Rademacher. Then, L_1 -minimization uniquely recovers $(x_o, \tilde{e}_o)$ all sufficiently sparse x_o with vast probability. Let J be a random subset of $\tilde{J}$ of size ρm , and $\sigma = \tilde{\sigma}(J)$. By **Lemma 13**, L_1 -minimization also uniquely recovers (x_o, e_o) for all sufficiently sparse x_o with the same overwhelmingly large probability and; hence, **Theorem 1** is proven.

This extension (for multi bouquet model) mimics the extension (for the bouquet model) in the previous work [4].

4.9. Experiments

This section compares the error correction (EC) abilities of the following four sparse recovery algorithms:

- both L_1 -minimizations in eqn. (1.1) and (1.2),
- a state-of-the-art sparse recovery L_1 -regularized Least Absolute (LqLA) algorithm [424] ($\lambda = 1$) using generalized nonconvex regularization, and
- a proposed LqLA (with adaptive λ) extension which λ is learnt by the proposed same procedure for eqn. (1.2) as described below.

$q = 0.5$ is used for the L_q regularizers in the algorithms of both LqLA and the proposed LqLA extension. The comparison investigates the effect of various model parameters on the four algorithms. Simulations with artificial data and the data in practical problems were conducted. Primal Augmented Lagrange Multiplier (PALM) algorithm was used for the L_1 -minimizations. The maximum iteration number was set to 100 and 100 simulated data were used in all experiments. Bicubic interpolation was used to resize the original image at high resolution to an image at a lower target resolution before being vectorized to being

low dimensional data or atoms of dictionary in the practical problem. The low dimension data or atom of the dictionary was generated as the following procedure:

- the image was resized to the targeted resolution (height and width), and
- the resized image was vectorized by stacking its columns or row into a low dimensional image vector.

The sparse coefficient vector and error recovery for all simulations depended on the signs and support of the ground truth sparse coefficient vector and error vector (x_o, e_o) . In all simulations, both $x_o(i)$ and $e_o(i) \in \{0,1\}$. An output $(\hat{x}, \hat{e})$ was determined as correct if the correctness condition in eqn. (4.54) was fulfilled.

$$\max(\|\hat{x} - x_o\|_\infty, \|\hat{e} - e_o\|_\infty) < 0.01 \quad (4.55)$$

Note that the correctness of output depends on two factors:

- (Factor 1) Recovery probability: Is the optimal solution of the L_1 -minimization equal to the ground truth sparse coefficient vector and error vector (x_o, e_o) ?
- (Factor 2) The convergence rate of the algorithm: Does the optimization converges to the optimal solution in the maximum number of iterations?

The maximum number of iterations of the algorithm was fixed so that the optimization was computed in feasible time. As the convergence rate of a high dimensional problem takes longer than in low dimensional problem; hence the output at 100 steps is incorrect if the eqn. (4.54) is not fulfilled even when Factor 1 is fulfilled.

The procedures of dictionary generation for simulations with artificial and practical datasets were as follows.

*Generation procedure of the dictionary A for the artificial original datasets follows the multi-bouquet model (defined in **Assumption 1** in Section 4.2) is shown in Figure 4.2.*

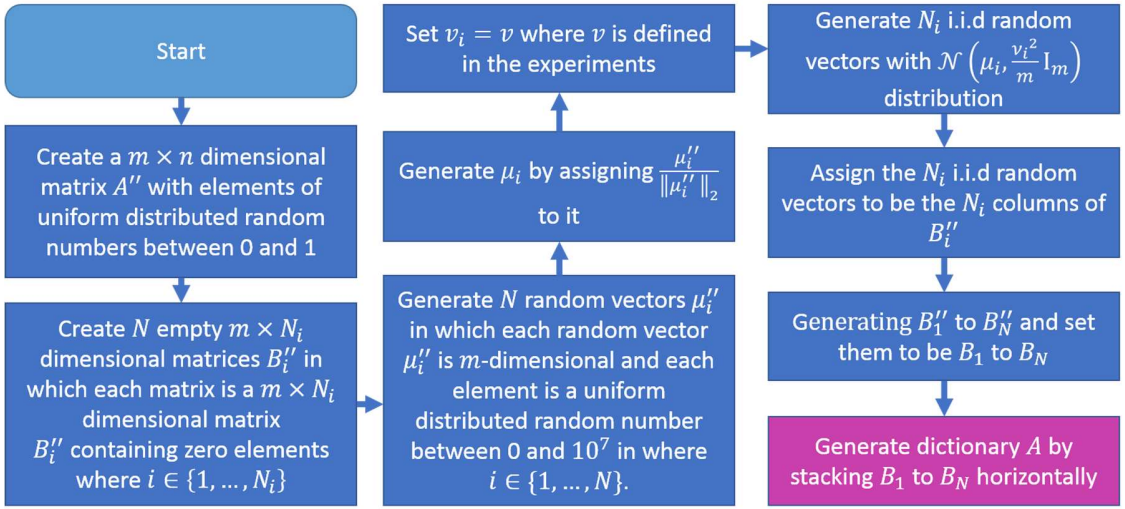


Figure 4.2 Dictionary generation procedure for the artificial dataset

Generation procedure of the dictionary A for the practical datasets is shown in Figure 4.3.

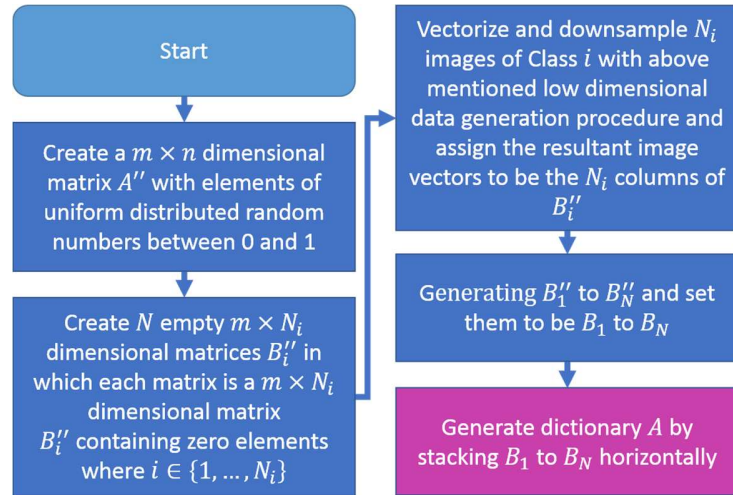


Figure 4.3 Dictionary generation procedure for the practical dataset

The procedure for the generation of the ground truth of the sparse coefficient vector x_o , the ground truth of the error vector e_o and the original input/observed sample y is shown in Figure 4.4

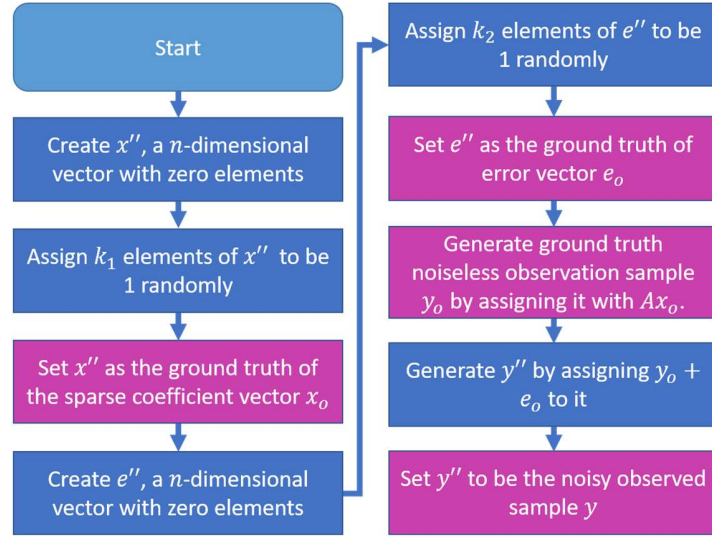


Figure 4.4 Data generation procedure for the artificial dataset

The procedure was the same for both simulations using artificial datasets and simulations using practical datasets.

Note that the “Adaptive λ ” and “ $\lambda = 1$ ” in the legends in the figures in the following subsections refer to the L_1 -minimizations in eqn. (1.1) and (1.2) respectively. In all figures below, the fraction of unsuccessful trials versus portions of corrupted elements (the elements which element values have ground truth residual of 1) in the original data ρ (from 0 to 1) are presented. Moreover, the adaptive λ in each experiment was selected by a particle swarm algorithm (PSO) within the range of $10^{-2} - 10^2$

Section 4.9.1 to Section 4.9.4 present the simulation results with the following data:

- artificial data which fulfils the bouquet model (multi-bouquet model with $N = 1$) (Section 4.9.1),
- artificial data which fulfils multi-bouquet model with $N > 1$ (Section 4.9.2),
- practical data used in face recognition (Section 4.9.3), and
- practical data used in viewpoint classification (Section 4.9.4).

4.9.1. Simulations using Artificial Data fulfilling the Bouquet Model

In this subsection, the comparisons of the EC abilities of the two L_1 -minimizations in eqns (1.1) and (1.2) and the algorithms of LqLA and the proposed LqLA extension are presented. Furthermore, the investigation of the effect of various model parameters on the four algorithms on the simulations with artificial data which fulfils the bouquet model (a special case of the multi-bouquet model with $N = 1$) are outlined. The sequence of the model parameters being investigated is as follows:

- the dimension of observed data m (see **Assumption 1**),
- the parameter δ which determines the dimension of the sparse coefficient vector x (see **Assumption 3**),
- the support size of the sparse coefficient vector x (see Section 4.1), and
- the parameter v which determines all v_i (see **Assumption 1**).

Figure 4.5 shows the EC abilities of both L_1 -minimizations in eqns (1.1) and (1.2), and the algorithms of LqLA and the proposed LqLA extension under changing dimension m and model parameter ρ . One of the two dimensions, $m = 200$ and $m = 400$, was used in each experimental case. Model parameters $\delta = 0.2$, $k_1 = 1$ and $v = 1$ were used in all cases. Results show that L_1 -minimization in eqn. (1.2) had zero unsuccessful trials for all ρ while the performance of L_1 -minimization in eqn. (1.1) dropped significantly since $\rho = 0.6$ when maximum iteration of 100 steps was applied. For the results of LqLA and the proposed LqLA extension, both algorithms surprisingly get 0 successful trials. It may be due to reason of the slow convergence of the algorithms as stated in [424] to reach the correctness condition in eqn. (4.55).

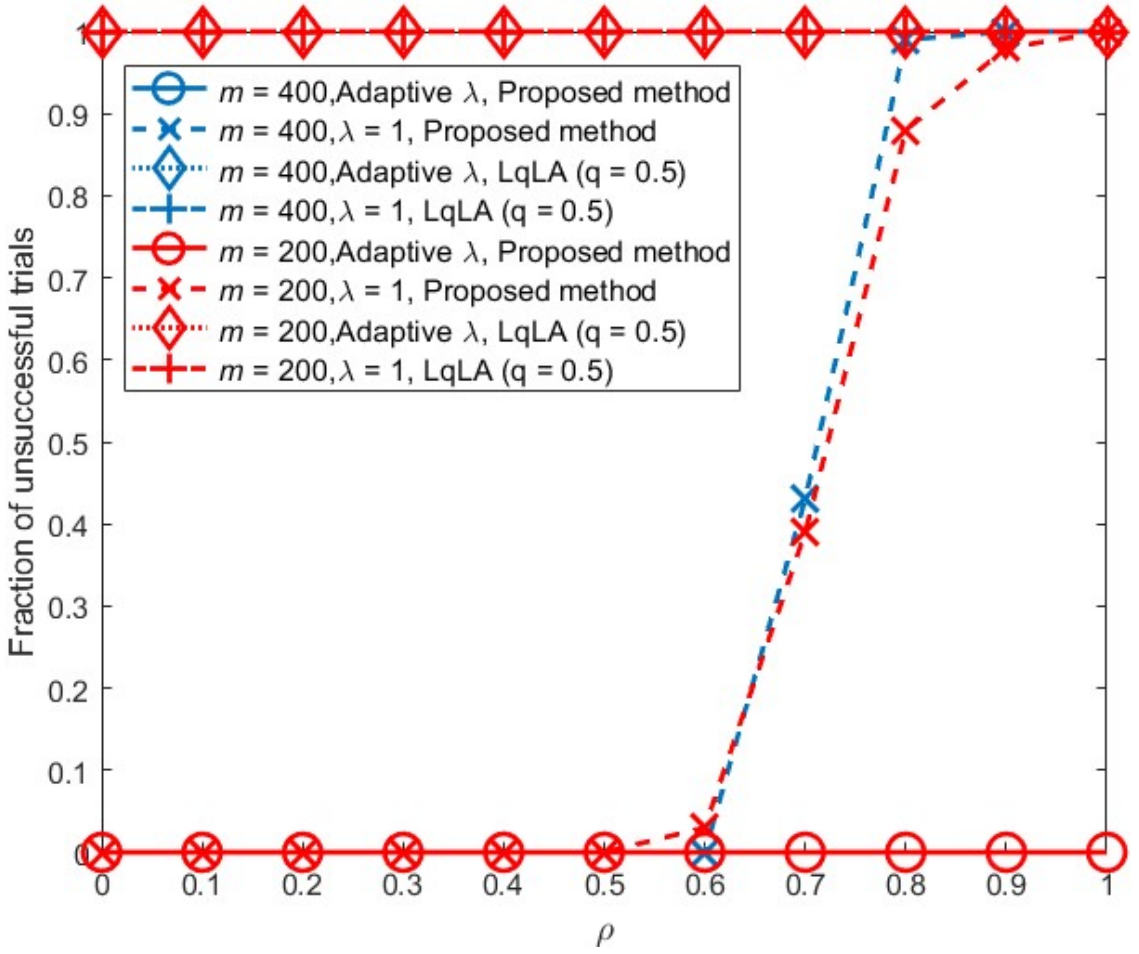
Figure 4.5 ECs under changing dimension m and model parameter ρ

Figure 4.6 shows the EC abilities of both L_1 -minimizations in eqns (1.1) and (1.2), and the algorithms of LqLA and the proposed LqLA extension under changing model parameters δ and ρ . The dimension of $m = 400$ and model parameters $k_1 = 1$, and $v = 1$, were used in all cases. Different model parameter δ 0.2, 0.5, 1 and 1.2, were used in different cases. Similar to the results in Figure 4.5, L_1 -minimization in eqn. (1.2) had significant good performance in all cases except when $\delta = 1.2$ and $\rho = 1$ while the performance of L_1 -minimization in eqn. (1.1) dropped significantly since $\rho \geq 0.6$. In general, the performances of both L_1 -minimizations decreased with the increase of δ . It can be understood as the larger the value of δ , the more atoms in the dictionary A and, hence, the higher the dimension of the dictionary A is. Overall, L_1 -minimization in eqn. (1.2) outperformed L_1 -minimization in eqn. (1.1) significantly when maximum iteration of 100 steps was applied. Same as the results of LqLA and the proposed LqLA extension in

previous graph, both algorithms surprisingly get 0 successful trials. It may due to reason of the slow convergence of the algorithms as stated in [424] to reach the correctness condition in eqn. (4.55).

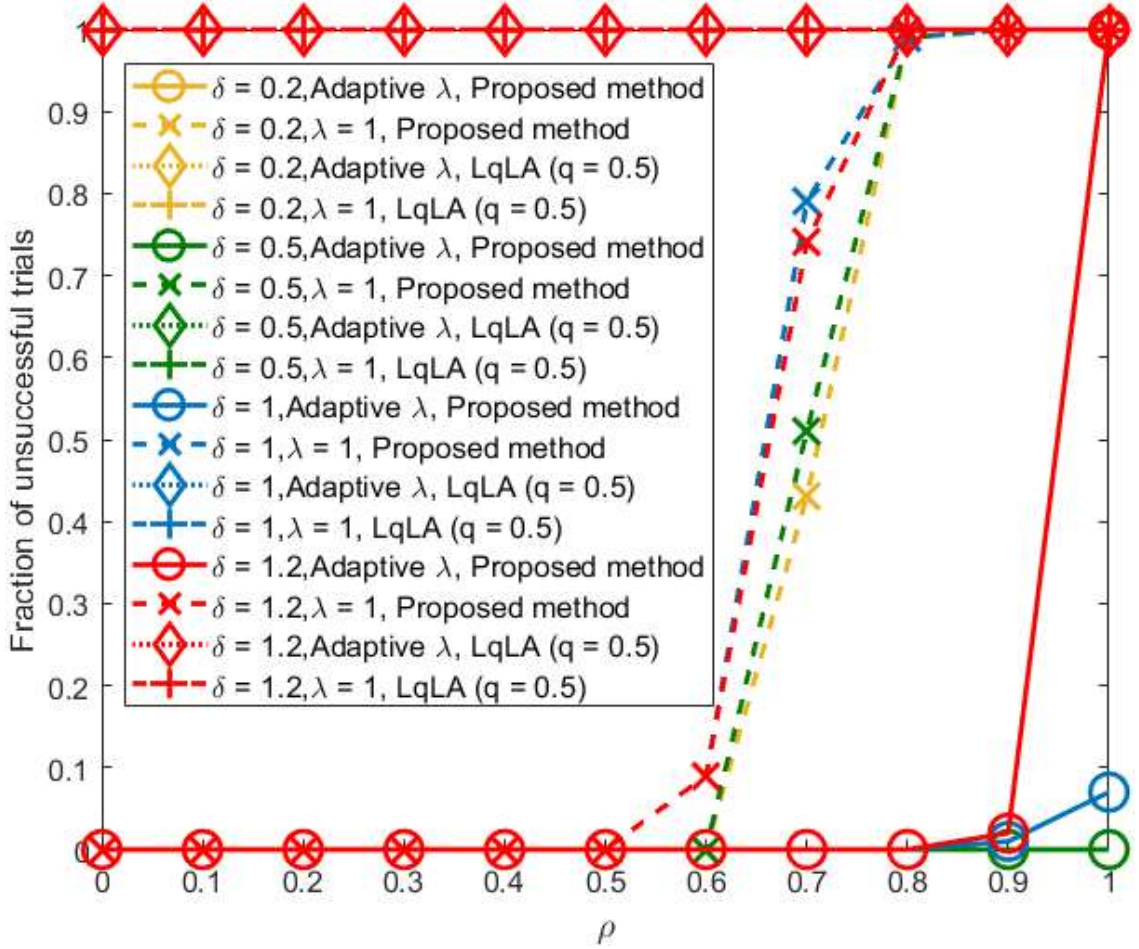


Figure 4.6 ECs under changing model parameters δ and ρ

Figure 4.7 shows the EC abilities of both L_1 -minimizations in eqns (1.1) and (1.2), and the algorithms of LqLA and the proposed LqLA extension under changing model parameters k_1 and ρ . The dimension of $m = 400$ and model parameters $\delta = 0.2$, and $v = 1$, were used in all cases. Different model parameter k_1 range from 1 to 3 were used in different cases. L_1 -minimization in eqn. (1.2) had significantly good performance in all cases except when $\rho = 1$ and $k_1 = 1$ or 2. L_1 -minimization in eqn. (1.1) performed badly since $\rho \geq 0.3$, $\rho \geq 0.5$ and $\rho \geq 0.7$ when in cases when $k_1 = 1$, $k_1 = 2$ and $k_1 = 3$ respectively. In general, both performances of L_1 -minimizations decreased with the increase of k_1 . It is understandable as the larger the value of k_1 , the longer the time to converge and the harder

the minimization is. Overall, L_1 -minimization in eqn. (1.2) outperformed L_1 -minimization in eqn. (1.1) significantly when maximum iteration of 100 steps was applied. Both algorithms of LqLA and the proposed LqLA extension surprisingly get 0 successful trials. It may due to reason of the slow convergence of the algorithms as stated in [424] to reach the correctness condition in eqn. (4.55).

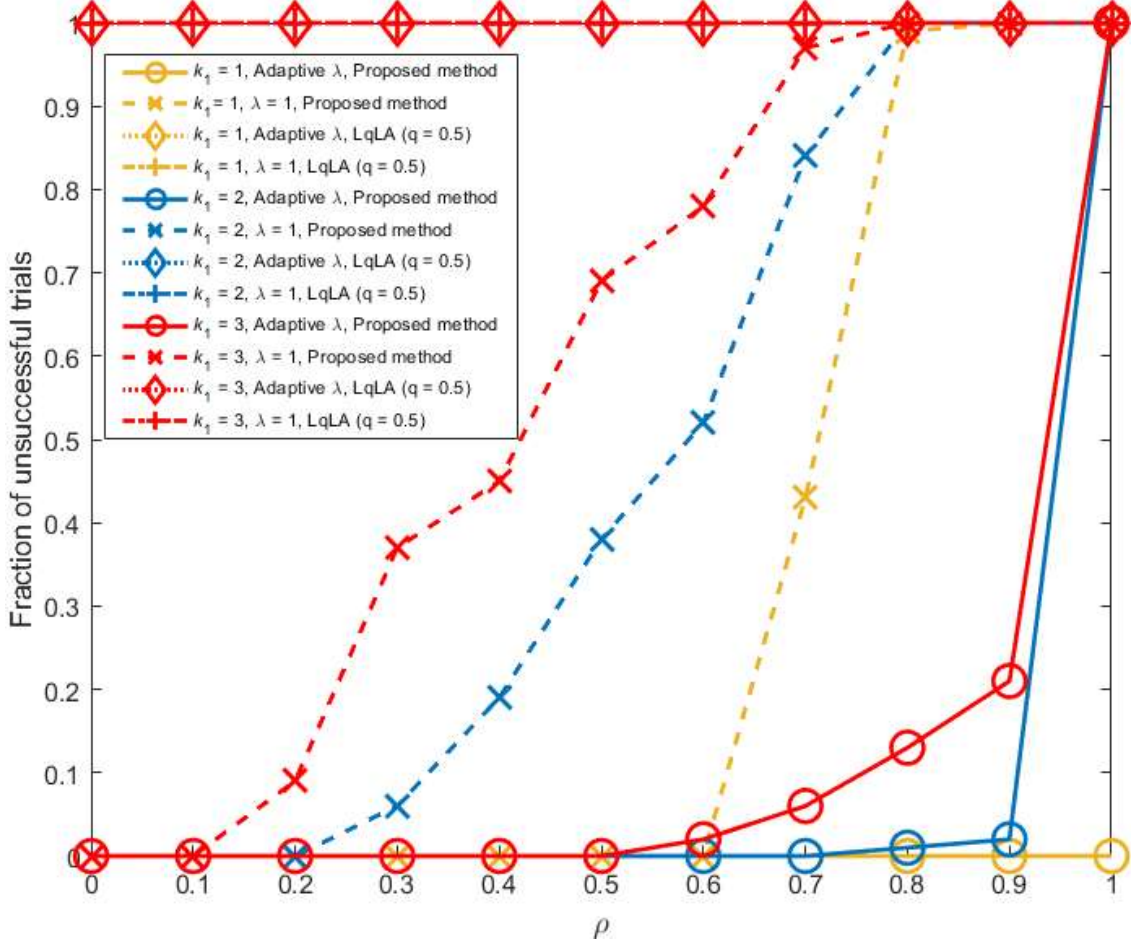


Figure 4.7 ECs under changing model parameter k_1 and ρ

Figure 4.8 shows the EC abilities of both L_1 -minimizations in eqns (1.1) and (1.2), and the algorithms of LqLA and the proposed LqLA extension under changing model parameters v and ρ . The dimension of $m = 400$ and model parameters $\delta = 0.2$, and $k_1 = 1$, were used in all cases. Different model parameter v , 0.1, 0.2, 0.5, 0.7, 1 and 2, were used in different cases. Both L_1 -minimization in eqns (1.1) and (1.2) performed poorly in cases which $v \leq 0.2$. L_1 -minimization in eqn. (1.2) only performed well when $\rho \leq 0.8$, $\rho \leq 0.9$ and $\rho \leq 0.9$ for cases which $v = 0.7$, $v = 0.5$ and $v = 1$ respectively and for all ρ for

cases which $v = 2$. L_1 -minimization in eqn. (1.1) only performed well when $\rho \leq 0.5$ for cases which $v = 0.5$ and when $\rho \leq 0.6$, for cases which either $v = 0.7$, $v = 0.1$ or $v = 2$. In general, the performances of both L_1 -minimizations increased with the increase of v . It is understandable as the larger the value of v , the larger the number of the discriminative atoms in the dictionary A , hence, the easier the minimization is and the faster the convergence is. Overall, L_1 -minimization in eqn. (1.2) outperformed L_1 -minimization in eqn. (1.1) significantly when maximum iteration of 100 steps was applied. Both algorithms of LqLA and the proposed LqLA extension surprisingly get 0 successful trials. It may due to reason of the slow convergence of the algorithms as stated in [424] to reach the correctness condition in eqn. (4.55).

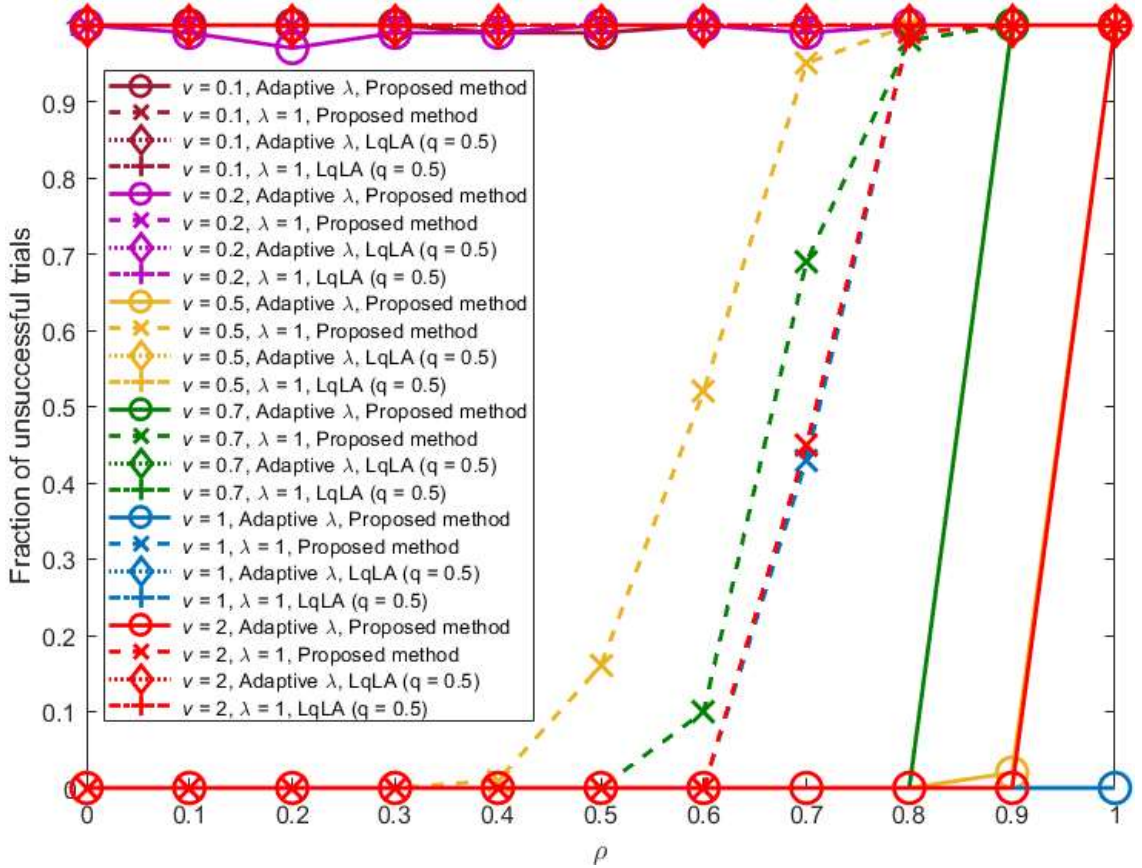


Figure 4.8 ECs under changing model parameters v and ρ

Overall, L_1 -minimization in eqn. (1.2) performed much better than L_1 -minimization in eqn. (1.1), and the algorithms of LqLA and the proposed LqLA extension in the simulations with artificial data which fulfils the bouquet model.

Note that all the sudden jump in the fraction of unsuccessful trials when L_1 -minimization in eqn. (1.2) was used may due to either the failure of PSO to find the best λ or the best λ was not in the search range.

4.9.2. Simulations using Artificial Data fulfilling the Multi-Bouquet Model

In this subsection, the comparisons of the EC abilities of the two L_1 -minimizations in eqns (1.1) and (1.2) and the algorithms of LqLA and the proposed LqLA extension are presented. Furthermore, the investigation of the effect of various model parameters, m , δ , k_1 and v on the four algorithms on the simulations with artificial data which fulfills multi-bouquet model with $N = 3$ are outlined.

Figure 4.9 shows the EC abilities of both L_1 -minimizations in eqns (1.1) and (1.2), and the algorithms of LqLA and the proposed LqLA extension under changing dimension m and model parameter ρ . One of three dimensions m , 50, 100 and 200, was used in each experimental case. Model parameters $\delta = 0.2$, $k_1 = 1$ and $v = 1$ were used in all cases. L_1 -minimization in eqn. (1.2) performed well in all cases while L_1 -minimizations in eqn. (1.1) only performed well for cases which $\rho \leq 0.6$ when maximum iteration of 100 steps was applied. Both algorithms of LqLA and the proposed LqLA extension surprisingly get 0 successful trials. It may due to reason of the slow convergence of the algorithms as stated in [424] to reach the correctness condition in eqn. (4.55).

Figure 4.10 shows the EC abilities of both L_1 -minimizations in eqn. (1.1) and (1.2), and the algorithms of LqLA and the proposed LqLA extension under changing model parameters δ and ρ . The dimension of $m = 200$ and model parameters $k_1 = 1$, and $v = 1$, were used in all cases. Different model parameter δ 0.2, 0.5, 1 and 1.2, were used in different cases. L_1 -minimization in eqn. (1.2) performed well in all cases while L_1 -minimization in eqn. (1.1) only performed well for cases which $\rho \leq 0.6$ when maximum iteration of 100 steps was applied. Both algorithms of LqLA and the proposed LqLA extension surprisingly get 0 successful trials. It may due to reason of the slow convergence of the algorithms as stated in [424] to reach the correctness condition in eqn. (4.55).

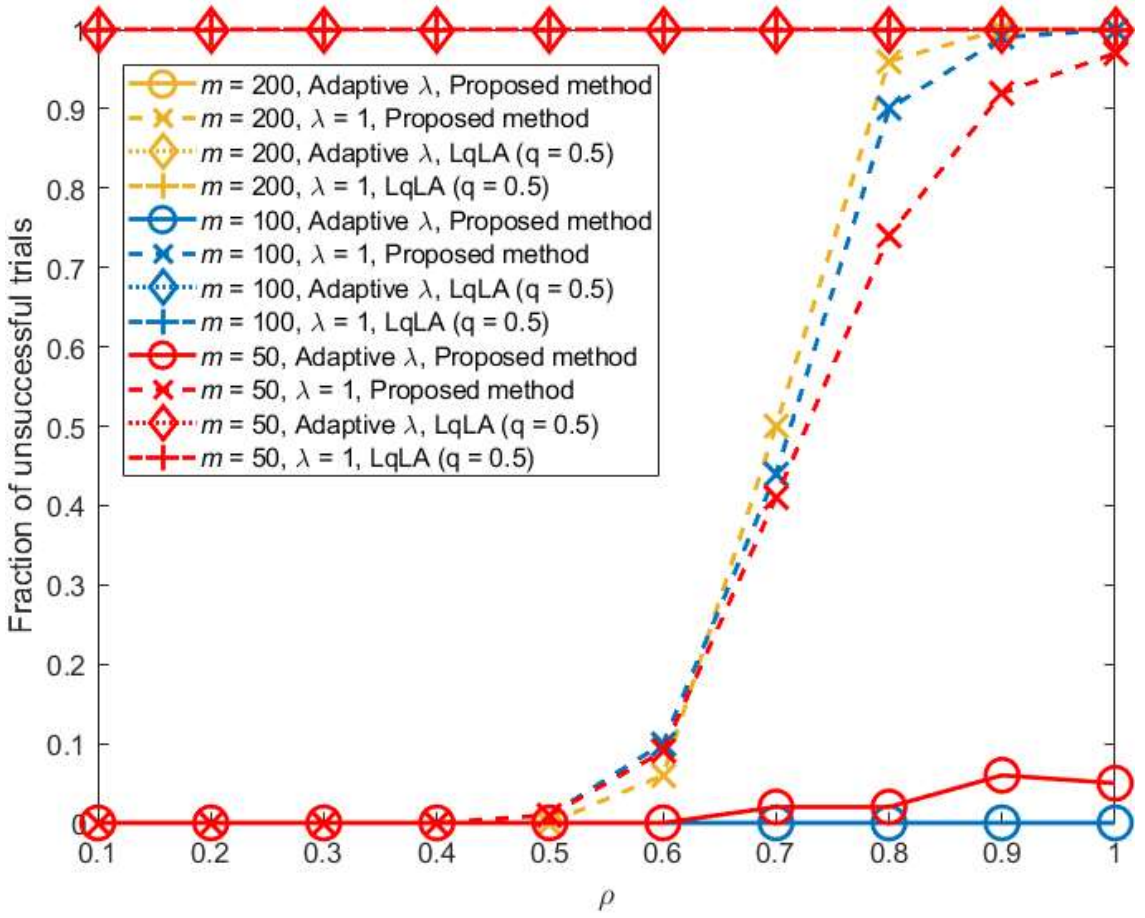
Figure 4.9 ECs under changing dimension m and model parameter ρ

Figure 4.11 shows the EC abilities of both L_1 -minimizations in eqn. (1.1) and (1.2), and the algorithms of LqLA and the proposed LqLA extension under changing model parameters k_1 and ρ . The dimension of $m = 200$ and model parameters $\delta = 0.2$, and $v = 1$, were used in all cases. Different model parameter k_1 range from 1 to 3 were used in different cases. L_1 -minimization in eqn. (1.2) performed well in all cases while L_1 -minimization in eqn. (1.1) only performed well for cases which $\rho \leq 0.6$ when maximum iteration of 100 steps was applied. Both algorithms of LqLA and the proposed LqLA extension surprisingly get 0 successful trials. It may due to reason of the slow convergence of the algorithms as stated in [424] to reach the correctness condition in eqn. (4.55).

Figure 4.12 shows the EC abilities of both L_1 -minimizations in eqn. (1.1) and (1.2), and the algorithms of LqLA and the proposed LqLA extension under changing model

parameters v and ρ . The dimension of $m = 200$ and model parameters $\delta = 0.2$, and $k_1 = 1$, were used in all cases. Different model parameter v 0.1, 0.2, 0.5, 0.7, 1 and 2, were used in different cases. Both L_1 -minimization in eqns (1.1) and (1.2) performed poorly in cases which $v \leq 0.2$. However, L_1 -minimization in eqn. (1.2) performed better than L_1 -minimization in eqn. (1.1) in all these cases. L_1 -minimization in eqn. (1.2) performed well in all cases when $v \geq 0.2$ except the cases which $\rho = 1$ while L_1 -minimization in eqn. (1.1) only performed well for cases which $\rho \leq 0.6$ when $v \geq 0.7$. Overall, L_1 -minimization in eqn. (1.2) outperformed L_1 -minimization in eqn. (1.1) significantly when maximum iteration of 100 steps was applied. Both algorithms of LqLA and the proposed LqLA extension surprisingly get 0 successful trials. It may due to reason of the slow convergence of the algorithms as stated in [424] to reach the correctness condition in eqn. (4.55).

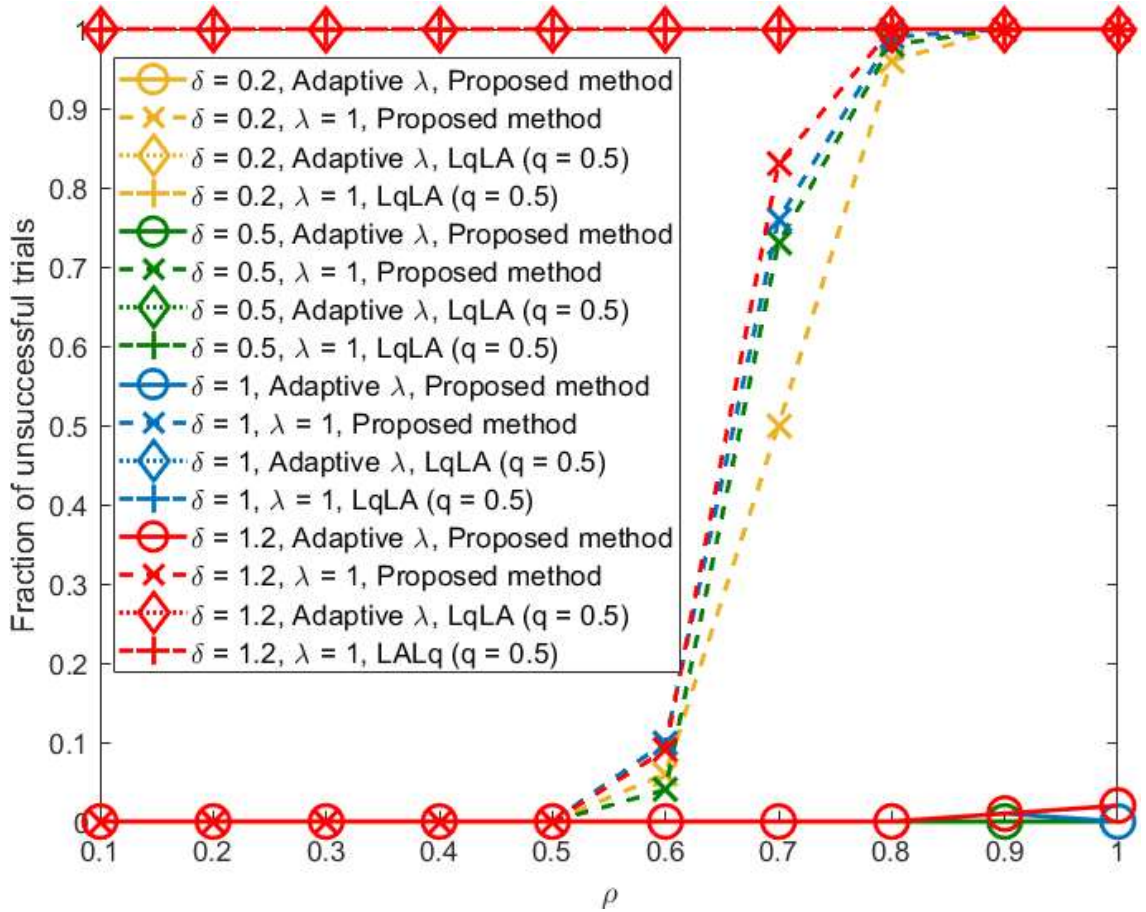
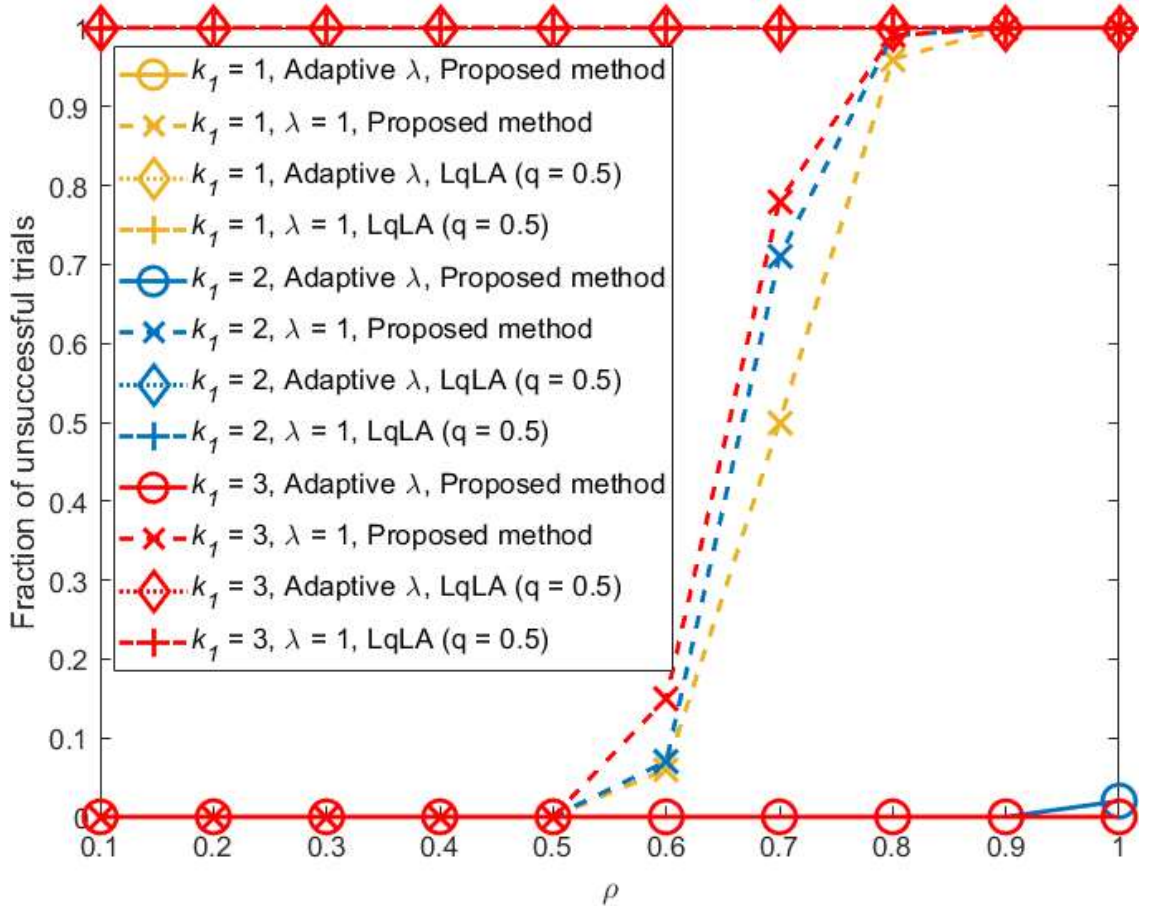
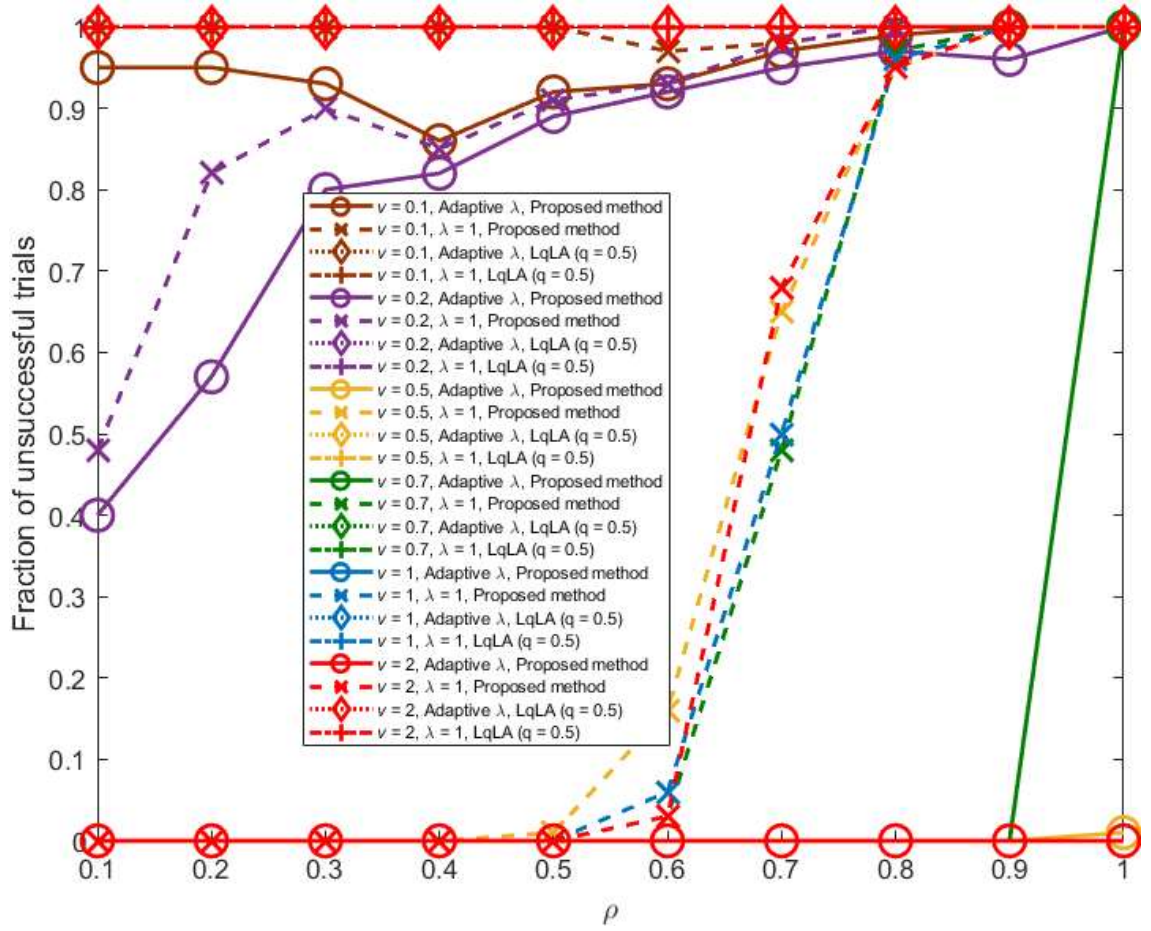


Figure 4.10 ECs under changing model parameters δ and ρ

Figure 4.11 ECs under changing model parameter k_1 and ρ

Overall, L_1 -minimization in eqn. (1.2) performed much better than L_1 -minimization in eqn. (1.1), and the algorithms of LqLA and the proposed LqLA extension in the simulations with artificial data which fulfils the multi-bouquet model.

Note again that all the sudden jumps in the fraction of unsuccessful trials when L_1 -minimization in eqn. (1.2) was used may due to either the failure of PSO to find the best λ or the best λ was not in the search range.

Figure 4.12 ECs under changing model parameters ν and ρ

4.9.3. Simulations using Practical Data used in Face Recognition

In this subsection, the comparisons of the EC abilities of the two L_1 -minimizations in eqns (1.1) and (1.2) and the algorithms of LqLA and the proposed LqLA extension are presented. Furthermore, the investigation of the effect of various model parameters on the two L_1 -minimizations on the simulations with practical data which was used in face recognition and fulfils multi-bouquet model with $N > 1$ are outlined. The data from Extended Yale B face database [425] was used. Figure 4.13 shows the exemplar images of 2 subjects from the Extended Yale B face database. The N represents the subject number. 19 images of each subject were used to construct the dictionary A .



Figure 4.13 Exemplar Images from Extended Yale B face database [425]

Figure 4.14 shows the EC abilities of both L_1 -minimizations in eqns (1.1) and (1.2), and the algorithms of LqLA and the proposed LqLA extension under changing dimension m , N and model parameter ρ . Three pairs of m and N , as follows:

- (1). $m = 30 \times 34$ and $N = 4$
- (2). $m = 42 \times 48$ and $N = 9$
- (3). $m = 60 \times 68$ and $N = 19$

were used in the experimental case while the integers stated for m is the product of height and width of the downsampled images before being vectorized. Model parameter $k_1 = 7$ was used in all cases. The results show that L_1 -minimizations in eqn. (1.2) always outperformed and performed equally than L_1 -minimization in eqn. (1.1) in cases which $\rho \leq 0.6$ and $\rho \geq 0.7$ respectively. Smaller the ρ produced much better performance of L_1 -minimization in eqn. (1.2) than L_1 -minimizations in eqn. (1.1). The gap of the fractions of unsuccessful trials between the two L_1 -minimizations started from above 0.2 and diminished to 0 when ρ increased from 0 to 1 in general. L_1 -minimizations in eqn. (1.1) had fractions of unsuccessful trials above 0.8 since $\rho = 0.6$ while L_1 -minimization in eqn. (1.2) had them only since $\rho = 0.7$. Both L_1 -minimizations had fractions of unsuccessful trials almost or equal to 1 when $\rho \geq 0.8$. Overall, L_1 -minimization in eqn. (1.2) performed better than L_1 -minimization in eqn. (1.1) in the face recognition simulations which data used fulfills multi-bouquet model. Both algorithms of LqLA and the proposed LqLA

extension surprisingly get 0 successful trials. It may due to reason of the slow convergence of the algorithms as stated in [424] to reach the correctness condition in eqn. (4.55).

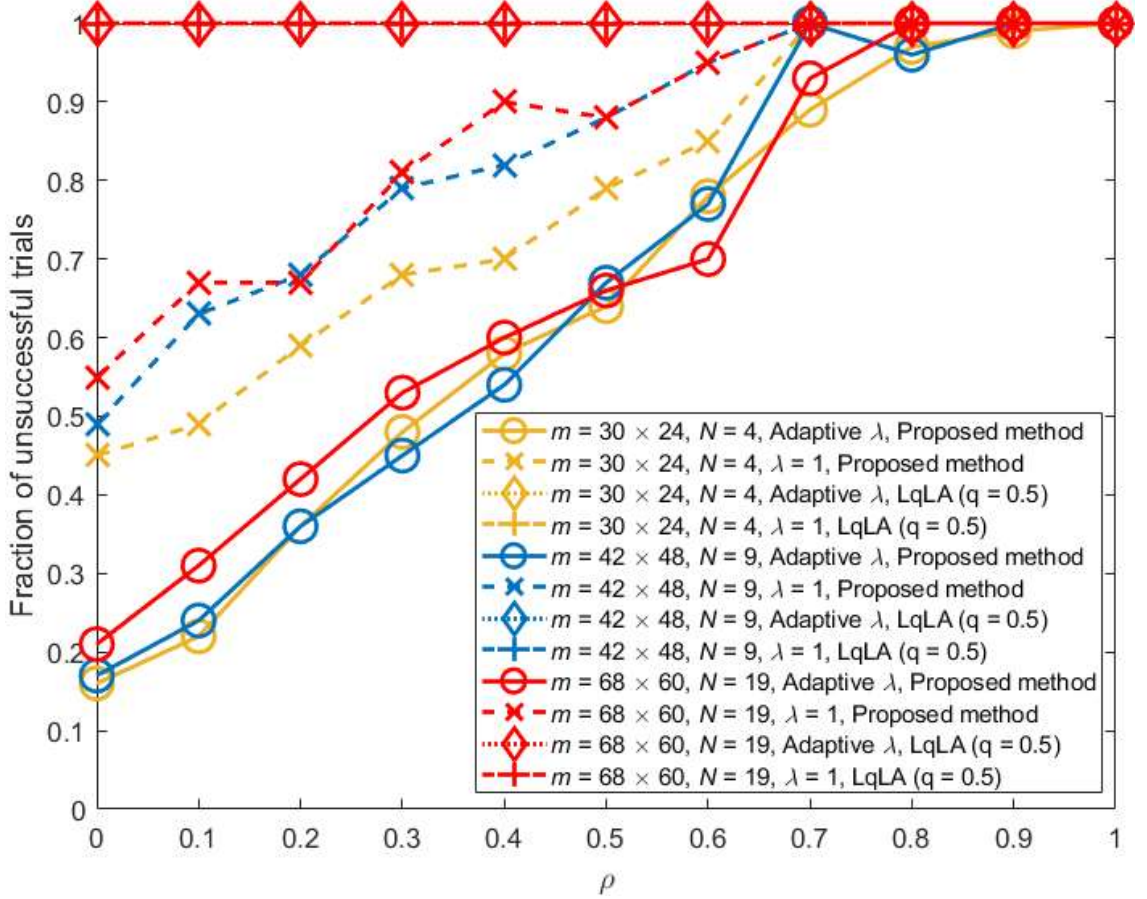


Figure 4.14 ECs under changing dimension m and model parameters N and ρ

4.9.4. Simulations using Practical Data used in Viewpoint Classification

In this subsection, the comparisons of the EC abilities of the two L_1 -minimizations in eqns (1.1) and (1.2) and the algorithms of LqLA and the proposed LqLA extension are presented. Furthermore, the investigation of the effect of various model parameters on the two L_1 -minimizations on the simulations with practical data from Rolls-Royce dataset which was used in viewpoint classification and fulfills multi-bouquet model with $N > 1$ are outlined. The data from Rolls-Royce database [5], [6] was used. Figure 4.15 shows the exemplar images of 2 viewpoints from Rolls-Royce database. The N represents the subject number. 19 images of each viewpoint were used to construct the dictionary A .

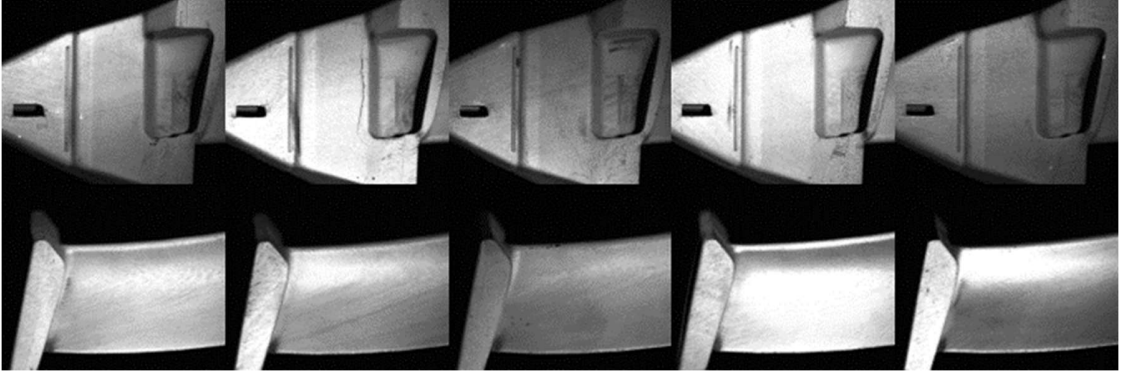


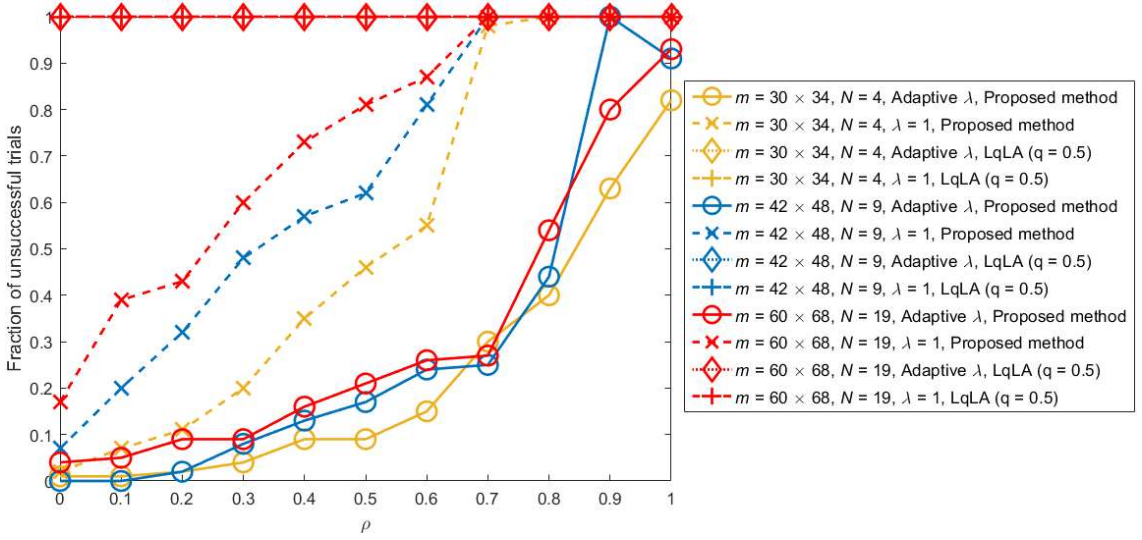
Figure 4.15 Exemplar Images from Rolls-Royce database [5], [6]

Figure 4.14 shows the EC abilities of both L_1 -minimizations in eqns (1.1) and (1.2), and the algorithms of LqLA and the proposed LqLA extension under changing dimension m , N and model parameter ρ . Three pairs of m and N , as follows:

- (1). $m = 30 \times 34$ and $N = 4$
- (2). $m = 42 \times 48$ and $N = 9$
- (3). $m = 60 \times 68$ and $N = 19$

were used in the experimental case while the integers stated for m is the product of height and width of the downsampled images before being vectorized. Model parameter $k_1 = 7$ was used in all cases. The results show that L_1 -minimizations in eqn. (1.2) outperformed L_1 -minimization in eqn. (1.1) in almost all cases. The gap of the fractions of unsuccessful trials between the two L_1 -minimizations started from small to large and large to small again. The largest gap happened when $\rho = 0.7$ and the gap was above 0.7. The gaps were smaller than 0.2 when $\rho = 0$ and $\rho = 1$. Overall, L_1 -minimization in eqn. (1.2) performed much better than L_1 -minimization in eqn. (1.1) in the viewpoint classification simulations which data used fulfills multi-bouquet model. Both algorithms of LqLA and the proposed LqLA extension surprisingly get 0 successful trials. It may due to reason of the slow convergence of the algorithms as stated in [424] to reach the correctness condition in eqn. (4.55).

Note again that all the sudden jump in the fraction of unsuccessful trials when L_1 -minimization in eqn. (1.2) was used may due to either the failure of PSO to find the best λ or the best λ was not in the search range.

Figure 4.16 ECs under changing dimension m and model parameters N and ρ

4.10. Summary

This chapter presented the performance of a recovery problem design (see Figure 1.1 and Figure 1.2), L_1 -minimization in eqn. (1.2) and the algorithms of LqLA and the proposed LqLA extension in recovering the sparse coefficient vector and the corrupting error vector of a corrupted data. The theoretical successful recovery probability lower bound is presented. It is an extension works of the main results in [4] which studies the problem of recovering the sparse coefficient vector and the corrupting error vector of a corrupted data using L_1 -minimization in eqn. (1.1). The results from [4] only can be applied with the assumption that the dictionary fulfils the bouquet model. The derivation of the works can be applied to not only the sparse representation problem in which dictionary fulfils bouquet model but also the problem in which the dictionary fulfils the multi-bouquet model. The main results state that the successful recovery probability bound was substantially high even nearly 100% of the measurements in the data y were corrupted when m was sufficiently large. Experimental results also showed that the L_1 -minimization in eqn. (1.2) outperformed L_1 -minimization in eqn. (1.1) and the algorithms of LqLA and the proposed LqLA extension in the benchmarks using both artificial and practical data. In the next chapter, an approach which solves a feasibility problem when computational resources are limited when L_1 -minimization in eqn. (1.2) is used as a sparse representation problem design.

Chapter 5

Feature Selection for Dictionary Transfer

“What is important in knowledge is not quantity, but quality. It is important to know what knowledge is significant, what is less so, and what is trivial.”

- Leo Tolstoy

In the previous two chapters, two contributions:

- *a novel optimization technique for solving nonsmooth optimization problem which a lot sparse recovery problem involves, and*
- *the theoretical derivation of successful sparse recovery probability bound using an L_1 -norm minimization*

for sparse recovery problems are presented.

This chapter focus on presenting a problem reformulation method which can make computationally non feasible dictionary-based problem due to the high dimensional dictionary to be feasible without retraining the dictionary. A lot of sparse recovery problems are formulated into a dictionary-based problem. A lot of dictionary-based problems are non feasible under constraints of computational power and resources due to the high dimensional dictionary being used. We use eqn. (1.2), a sparse representation problem design, please refer to Figure 1.1 and Figure 1.2, as the sparse recovery/dictionary-based example and present a framework of applying unsupervised feature selection to solve the feasibility problem. Meanwhile, the performances of several state-of-the-art unsupervised feature selections are investigated.

In dictionary-based artificial intelligent approaches, the dictionaries which are constructed or learnt during the training stage are frequently reused in the future. The reuse of these dictionaries may happen in other machines which may have lower computational power or memory capacity. Dictionaries with large dimension may become impractical in these machines. Moreover, retraining of new lower-dimensional dictionaries can be very time-consuming especially in a large-scale problem or/and with a large number of training data. Hence, a dictionary reconstruction approach with a simple proposed unsupervised feature selection technique is proposed to resolve the above-mentioned problem without retraining the dictionaries. Two following assumptions are made:

- *the underlying relationship between each feature in the dictionaries are unknown, and*
- *the training process of the dictionaries is unknown.*

Both clustering and nearest neighbour algorithm are utilized in the proposed approach. The objective of the new technique is to generate new lower-dimensional dictionaries which make the problem feasible and preserve the optimized outputs the same as or as nearly possible as the outputs obtained using the original high dimensional dictionaries. The promising experimental results show that the proposed technique is superior to the naïve down-sampling techniques which are frequently used for dimensional reduction, especially in image-based problems and two state-of-the-art unsupervised feature selection methods.

Section 5.1 presents some related works in unsupervised feature selection. Section 5.2 introduces the motivation of the proposed approach. Section 5.3 presents the main notations used in the chapter and the assumptions of the

proposed technique. Section 5.4 elaborates the details of the proposed approach. It includes

- *feature grouping with unsupervised clustering (Section 5.4.1),*
- *feature selection with nearest neighbour approach (Section 5.4.2), and*
- *the numbers of the cluster and nearest neighbours to be selected (Section 5.4.3).*

Section 5.5 presents the experimental results by comparing the proposed approach with the naïve downsampling technique and two state-of-the-art unsupervised feature selection method.

5.1. Related Works

Nowadays, high-dimensional data widely appears in various application fields such as image, video and multimedia processing and analysis, sound, time series, document and gene expression analysis, etc.

Feature selection which reduces the original feature dimensions to simplify machine learning, data mining and knowledge discovery problem and reducing overfitting problem plays a critical role in removing less important features. According to the availability of the labels in the feature selection learning process, the processes are generally categorized to supervised, semi-supervised and unsupervised feature selection. Supervised feature selection [426]–[428] aims to select features which are likely to have higher discriminative powers for the label information. Semi-supervised feature selection [429]–[431] selects the discriminative features from a data pool in which a small portion of it is labelled. Unsupervised feature selection [432]–[434] learns the significant representative features from unlabelled data.

Unsupervised feature selection is significantly being interested in the scientific community recently. Unsupervised feature learning has been widely used. It has shown promising results in various field such as data clustering [435]–[440], document analysis [441], [442],

and image and video analysis [443], [444]. Unsupervised feature selection advantages have two-fold: (1) they are unbiased when the prior knowledge is not available, and (2) they are less likely causing data overfitting than supervised feature selection methods which results may not work well when new classes are introduced into the task. Unsupervised supervised learning generally can be categorized into 3 categories: (1) Filter methods [445], (2) Wrapper methods [446], and (3) Embedding methods [447]. Recently, the manifold structure and sparse learning mechanism draw a lot of attention from the community. Some recent contributions are mentioned as follows. A feature selection approach which preserves the multi-cluster structure of data was proposed by Cai et al. [448]. An approach which applies a local discriminative score to reflect structure information with an $L_{2,1}$ regularizer was presented by Yang et al. [119]. An approach which utilized local discriminative information, manifold structures and feature correlations simultaneously were introduced by Li et al. [449]. An approach which jointly performs robust label learning and robust feature learning was proposed by Qian and Zhai [450]. An approach which performs simultaneous orthogonal basis clustering and feature selection by estimating latent cluster centres for the projected data were presented by Han and Kim [451]. A projection-free sparse learning approach which performs feature selection for clustering was introduced by Wang et al. [452]. An approach adaptively learning a similarity matrix to determine the structure of selected features were proposed by Nie et al. [453]. A consensus clustering-based method was introduced by Liu et al. [454] for pseudo-labelling and feature selection.

Unsupervised feature learning has been widely used. It has shown promising results in various field such as data clustering [17–22], document analysis [23,24], and image and video analysis [25,26]. However, its ability has not been exploited much in sparse recovery. In the following sections, we investigate how the unsupervised feature selection can solve an unfeasible problem and how it can improve the performance of sparse recovery/dictionary-based problem. A simple new feature selection method which is a combination of clustering and k-nearest neighbour in series is proposed and its performance in improving sparse recovery is investigated with several the state-of-the-art unsupervised learning methods.

5.2. Motivation

In general, dictionary-based approaches solve one or multiple optimization problems to derive the optimized arguments which can be used to achieve the problems' objectives. In most cases, the optimized arguments are vectors such as the sparse representation [3]–[6]. The L_1 -norm minimization in eqns (1.1) and (1.2) are the examples to learn the sparse coefficient vector $x \in \mathbb{R}^n$ and the error vector $e \in \mathbb{R}^m$. In the problems, the dictionary $A \in \mathbb{R}^{m \times n}$ may be learnt by a dictionary learning algorithm or which atoms are constructed by sample-based technique such as the atoms can be normalized, vectorized and resized training data. In this chapter, both the L_1 -norm minimizations in eqn. (1.2) is used for the demonstration purpose and represents all dictionary-based problems. The proposed method can be used on any single or multiple dictionaries and is not limited to the dictionary which fulfils the equalities stated in eqn. (1.2).

$$\begin{aligned} & \min_{x,e} \|x\|_1 + \lambda \|e\|_1 \\ & s. t. \quad y = Ax + e \end{aligned} \quad \begin{array}{l} \text{Recap} \\ (1.2) \end{array}$$

where

- λ is a tradeoff parameter,
- $y \in \mathbb{R}^m$ is the input/observed data, and
- $A \in \mathbb{R}^{m \times n}$ is the dictionary.

Both L_1 -norm minimization in eqns (1.1) and (1.2) have been rigorously used in face recognition [3], error correction [4], anomaly detection [5], [6] and background subtraction [153]. Eqn. (1.1) is the eqn. (1.2) with $\lambda=1$. The feasible size of dictionary A which can be stored by the memory and the reasonable time of convergence are different for different computational units. Hence, dimension reduction of the dictionary is necessary to make the problem feasible for a lower end computational unit. Moreover, the results obtained by solving the problem with the lower dimensional dictionary should be similar to the results expected from the original problem.

5.3. Main Notations and Assumptions

The following two important notations:

- the input is a high dimensional dictionary $A \in \mathbb{R}^{m \times n}$, and
- the output is a low dimensional dictionary $A' \in \mathbb{R}^{m' \times n}$ which is a submatrix of A where $m \gg m'$.

and two assumptions:

- the underlying relationship between each feature in the dictionaries are unknown, and
- the training process of the dictionaries is unknown.

are noted in the problem.

Due to the two assumptions, conventional feature selection approaches such as the image-based super-resolution feature selection approach which utilizes the information of the location of each feature (pixel) to construct the clusters of all features on the image cannot be used.

5.4. Feature Selection for Dictionary Transfer

In this section, the details of the proposed approach which consists of the combination of clustering and nearest neighbor techniques is elaborated.

There are three ways to reduce the dimension of the dictionaries:

- reduce the feature number (row) in the dictionaries,
- reduce the atom number (column) in the dictionaries, or
- reduce both the feature and atom numbers in the dictionaries.

As the objective of the problem is to get the optimized arguments (x and e) as nearly the same as the ones derived from optimization using the original dictionaries, the second and third ways are eliminated. The dimensional reduction through the first way is a feature selection manner.

Moreover, **Theorem 1** in Chapter 4 indicates that the higher the v_{max} (please refer to its definitions in the theorem in Section 4.3), the higher the lower bound of the successful recovery probability of sparse coefficient vector x and error vector e under the assumptions made in the theorem. It implies that the more discriminative the feature in the dictionary A' is the higher the lower bound of the successful recovery probability will be.

Based on the above-mentioned implications, an approach which contains the following two steps is proposed for the first way of the dimension reduction:

- feature grouping, and
- discriminative feature selection

The two steps are done by unsupervised clustering and nearest neighbour approach respectively. Unsupervised clustering approach is used to partition all features into clusters when the relationships such as locations or correlations between features in the existing dictionary are unknown. Similar features are grouped into the same cluster. To preserve the optimized argument values (values of x and e) to be as near as possible to the one learnt from the optimization with original dictionary, nearest neighbor technique is used to select the discriminative features to be reserved in the new dictionary without any feature transformation.

Note that the proposed approach is not limited to the problem which only consists of one dictionary but can be also applied on any single or multiple existing dictionaries in any dictionary-based problems.

A new L_1 -norm minimization in eqn. (5.1) is used to estimate the sparse coefficient vector $\hat{x}$.

$$\begin{aligned} & \min_{x, e'} \|x\|_1 + \lambda \|e'\|_1 \\ \text{s.t. } & y' = A'x + e' \end{aligned} \quad (5.1)$$

where

- λ is a tradeoff parameter,
- $A' \in \mathbb{R}^{m' \times n}$ is the lower dimensional dictionary which is the output of conducting the dimension reduction technique (e.g. the proposed method) on the original dictionary $A \in \mathbb{R}^{m \times n}$,
- $y' \in \mathbb{R}^{m'}$ is a m' -dimensional data and is the data containing the features selected by the dimensional reduction technique from the original/ground truth input/observed data $y \in \mathbb{R}^m$, and
- $e' \in \mathbb{R}^{m'}$ is a m' -dimensional corrupting error vector.

Figure 5.1 presents the pipeline of the proposed approach. In the example in Figure 5.1, each feature (a.k.a. a feature vector and a row of A e.g. F_1) of all data is to be clustered with all other feature vectors. The feature vectors which are nearest to the cluster centres (the small dots which connect to different features with lines) are chosen to be kept in the new lower-dimensional dictionary A' . In the example, $m = 9$, $m' = 6$, $N_c = 3$ and $N_{nn} = 2$ are used.

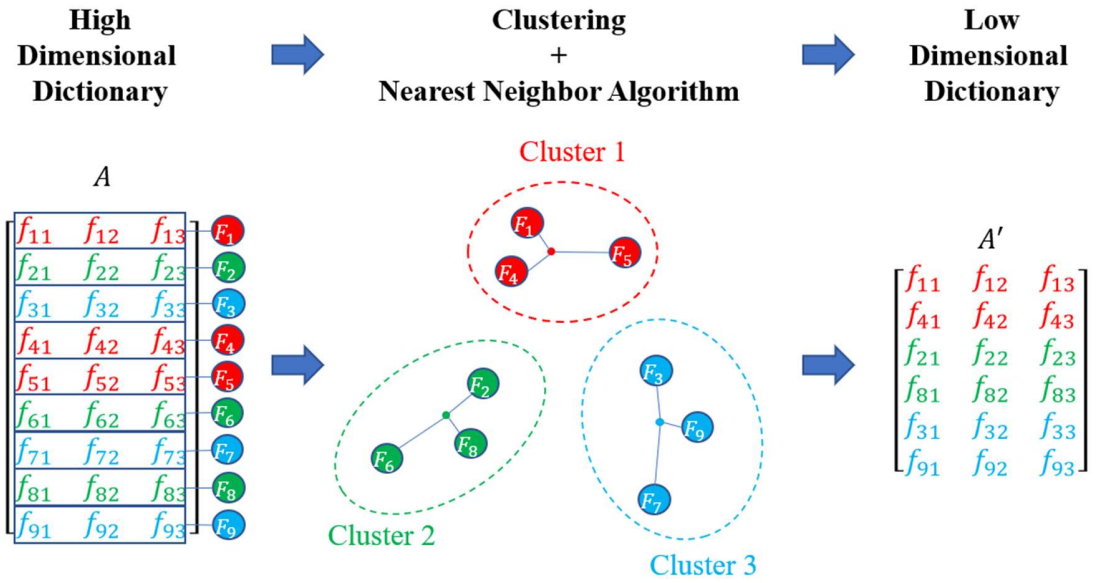


Figure 5.1 Proposed Method Pipeline

In the following Sections 5.4.1 and 5.4.2, the details of the feature grouping with unsupervised clustering and the following discriminative feature selection with the nearest neighbor technique are presented respectively. Section 5.4.3 elaborates on the idea of choosing a reasonable cluster number N_c and the nearest neighbor number N_{nn} to the cluster centers to be selected for the formation of the new dictionary A' . The detailed algorithms of the proposed approach can be found in Appendix D.

5.4.1. Feature Grouping with Unsupervised Clustering

Assuming that the underlying relationship between the features in the existing dictionary A is unknown, hence, the supervised clustering techniques cannot be used. In order to

remove redundant features and, to have discriminative atoms at a later stage, unsupervised clustering which is good in grouping similar features is used. For those lower end computational units which have the memory capacity issue, sequential or batch-based clustering techniques can be used. For the demonstration purpose, the k-mean clustering which uses a small number of features to sequentially learn the clustering centroids at a time is used in this chapter.

Suppose dictionary A is expressed as in eqn. (5.2).

$$A = \begin{bmatrix} f_{11} & \cdots & f_{1n} \\ \vdots & \ddots & \vdots \\ f_{m1} & \cdots & f_{mn} \end{bmatrix} = \begin{bmatrix} F_1^t \\ \vdots \\ F_m^t \end{bmatrix} \quad (5.2)$$

where F_i is equal to $\begin{bmatrix} f_{i1} \\ \vdots \\ f_{in} \end{bmatrix}$ and is the i^{th} feature vector in dictionary A . The m feature vectors,

F_1 to F_m , are to be clustered into N_c clusters where $N_c < m$ using a clustering algorithm. Each feature vector is assigned to a cluster. The outputs of the clustering are passed to the next step which is elaborated in Section 5.4.2. It further selects the discriminative feature vectors to construct the new dictionary A' , where $A' \in \mathbb{R}^{m' \times n}$ is a low dimensional dictionary and is a submatrix of A and $m \gg m'$. Note that $N_c = 3$ (3 clusters) is used in the example in Figure 5.1.

5.4.2. Feature Selection with Nearest Neighbor Approach

In the previous subsection, all feature vectors, $F_i, i = 1 \dots m$, are grouped into different clusters. Each cluster contains feature vectors which are discriminative from the feature vectors in other clusters. In order to preserve the values of the optimized arguments as closely as possible to the ones obtained from the optimization using the original dictionary A , a subset of all original feature vectors is chosen to be kept in the new dictionary A' without transforming them. Nearest neighbor algorithm is used to select a number, either $N_{nn} < m$ or $N_{nn} - 1 < m$, of feature vectors which have the nearest distance (e.g. root mean square) to its corresponding cluster center. These feature vectors are to be kept in the new lower-dimensional dictionary A' . In Figure 5.1, $N_{nn} = 2$ is used.

5.4.3. Numbers of Cluster and Nearest Neighbors to be selected

The cluster number and the number of the nearest neighbors to the cluster centers to be selected to form the new dictionary A' are dependent on the maximum feasible size of A' . The maximum feasible size of A' can be approximated by the following 2 factors:

- memory capacity, and
- reasonable time to derive/compute accurate optimization results.

Suppose only one lower-dimensional dictionary, A' , is to be used in the new optimization in a lower end computational unit and $m'_{max} \geq m'$ is the maximum feasible row number of A' after considering the 2 factors above. Let N_{nn} features be selected from $N_c - \text{mod}(m'_{max}, N_c)$ clusters and $N_{nn} - 1$ features be selected from the other $\text{mod}(m'_{max}, N_c)$ clusters. Eqns (5.3) and (5.4) are established.

$$m' = N_c \times N_{nn} - \text{mod}(m'_{max}, N_c) \quad (5.3)$$

$$m'_{max} \geq N_c \times N_{nn} - \text{mod}(m'_{max}, N_c) \quad (5.4)$$

where $\text{mod}(m'_{max}, N_c)$ is $m'_{max} \bmod N_c$ and mod is the modulus operator. In the experiment in this chapter, $N_{nn} = 1$ is used and N_c is arbitrarily fixed. For those optimizations with more than one dictionary, the values of N_c and N_{nn} for each dictionary can be selected by extending the above-mentioned idea. Note that the numbers of selected feature in all clusters and the number of clusters are not the focus of the research. The number selection method described in this subsection is only one of the many ways of selecting the numbers. The numbers can be adjusted based on the clustering method or prior knowledge.

5.5. Experiments

This section compares the error correction (EC) ability of the following four algorithms on the original input/observed data and dictionary.

- L_1 -minimizations in eqn. (5.1) using the proposed dimension reduction method,
- Naïve downsampling technique, and
- two state-of-the-art unsupervised feature selection algorithms:
 - Unsupervised Feature Selection with Ordinal Locality (UFSOL) [455], and
 - Unsupervised Feature Selection with Multi-Subspace Randomization and Collaboration (SRCFS) [456]

Naïve downsampling technique downsamples the original input/observed data and all original dictionary's atoms to the targeted dimensions m' by extracting the vector's element periodically with the largest feasible downsampling rate. Both UFSOL and SRCFS similarly select features unsupervised based on the feature relationships learnt with specific objective functions as our proposed method do. Simulations with artificial data and the data in practical problems were conducted. Primal Augmented Lagrange Multiplier (PALM) algorithm was used for the L_1 -minimizations. Moreover, the maximum iteration number was set to 100 and 100 simulated data were used in all experiments. λ in each experiment was selected using a particle swarm algorithm (PSO) within the range of $10^{-2} - 10^2$. Bicubic interpolation was used to resize the original image at high resolution to an image at a lower target resolution before being vectorized to the so-called original (m -dimensional) data and the atoms of the original dictionary in the practical problem. Dimension reduction technique based on either the proposed method or using Naïve downsampling technique was conducted on the original data and the atoms of the original dictionary afterward. The original data or an atom of the original dictionary was generated as shown in the following procedure:

- the image was resized to the targeted resolution (height and width), and
- the resized image was vectorized by stacking its columns or row into a low dimension image vector.

The sparse coefficient vector and error recovery for all simulations depended on the signs and support of the ground truth sparse coefficient vector and error vector (x_o, e_o) . In all simulations, both $x_o(i)$ and $e_o(i) \in \{0,1\}$. An output $(\hat{x}, \hat{e})$ was determined as correct if the correctness condition in eqn. (5.5) was fulfilled.

$$\max(\|\hat{x} - x_o\|_\infty, \|\hat{e} - e_o\|_\infty) < 0.01 \quad (5.5)$$

Note that the correctness of an output depends on two factors:

- (Factor 1) Recovery probability: Is the optimal solution of the L_1 -minimization equal to the ground truth sparse coefficient vector and error vector (x_o, e_o) ?
- (Factor 2) The convergence rate of the algorithm: Does the optimization converges to the optimal solution in the maximum number of iterations?

The maximum number of iterations of the algorithm was fixed so that the optimization was computed in feasible time. As the convergence rate of high dimensional problem takes

longer than in low dimensional problem; hence the output at 100 steps is incorrect if the eqn. (4.54) is not fulfilled even when Factor 1 is fulfilled.

The procedures of the original dictionary generation for simulations with artificial and practical datasets were as follows.

*Generation procedure of the original dictionary A for the artificial original datasets follows the multi-bouquet model (defined in **Assumption 1** in Section 4.2) is shown in Figure 5.2.*

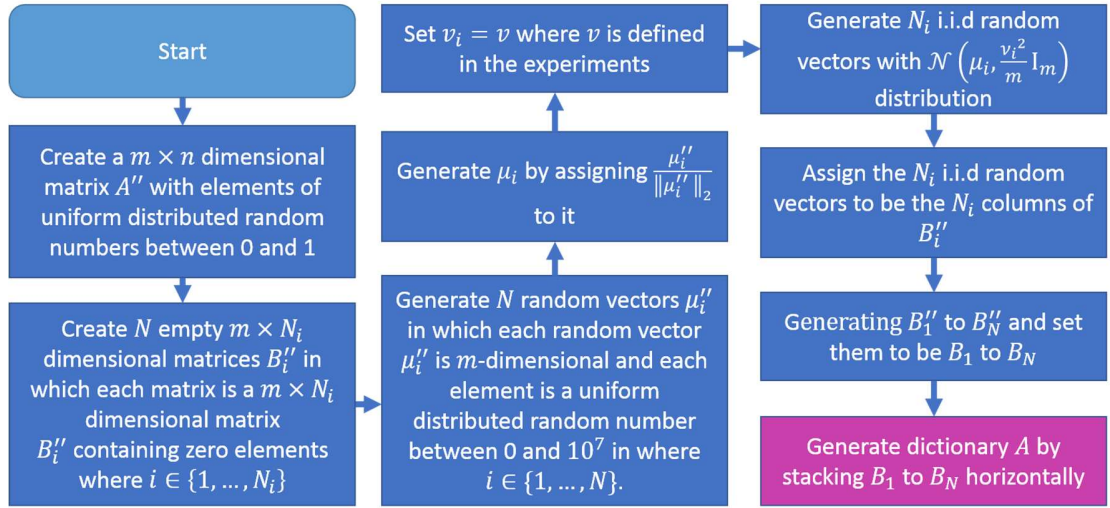


Figure 5.2 Dictionary generation procedure for artificial dataset

Generation procedure of the original dictionary A for the practical datasets is shown in Figure 5.3.

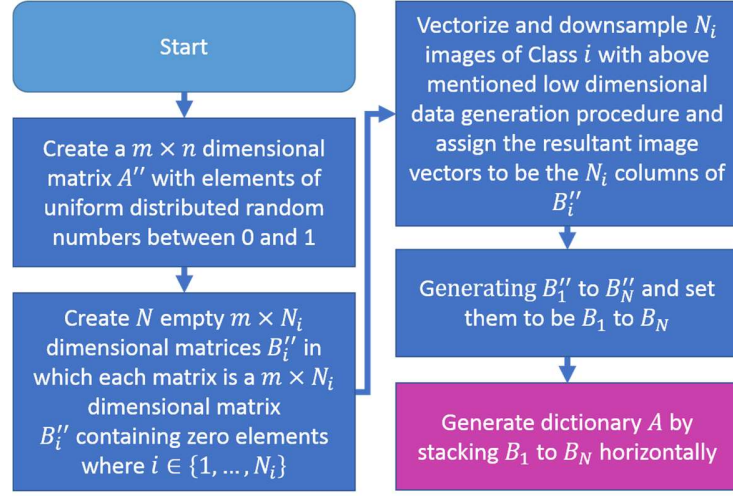


Figure 5.3 Dictionary generation procedure for practical dataset

The procedure for the generation of the ground truth of the sparse coefficient vector x_o , the ground truth of error vector e_o and the original input/observed sample y is shown in Figure 5.4.

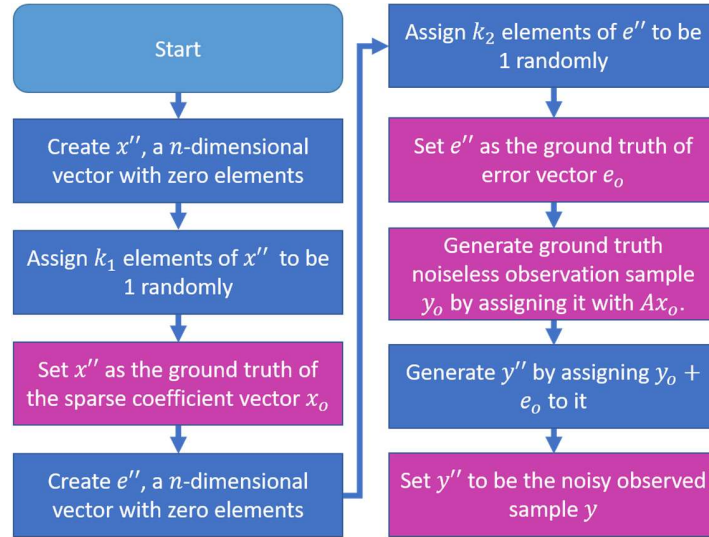


Figure 5.4 Data generation procedure for both artificial and practical datasets

The procedure was the same for both simulations using artificial datasets and simulations using practical datasets.

Note that the fraction of unsuccessful trials versus portions of corrupted elements (the elements which element values has ground truth residual of 1) in the original data ρ (from 0 to 1) are presented in all figures below.

Section 5.5.1 to 5.5.3 present the simulation results using the following data:

- artificial data which fulfills the multi-bouquet model defined in Section 4.2 (Section 5.5.1),
- practical data used in face recognition (Section 5.5.2), and
- practical data used in viewpoint classification (Section 5.5.3).

5.5.1. Simulations using Artificial Data fulfilling the Multi Bouquet Model

In this subsection, the comparisons of the EC ability of L_1 -minimizations in eqn. (5.1) using the four following unsupervised feature selection methods on the original input/observed data and dictionary are presented:

- the proposed dimension reduction method,
- Naïve downsampling technique and
- the two state-of-the-art techniques, UFSOL, and SRCFS.

UFSOL with the numbers of the nearest neighbors equal to 5 and cluster number equal to ceiling of $\frac{n}{3}$ is used. SRCFS with the numbers of the nearest neighbors equal to 3, random subspaces in each basic feature partition and basic feature partitions of SRCFS equal to ceiling of $\frac{m'}{10}$ are used.

The dataset follows the multi-bouquet model and the model parameters defined in **Assumption 1** and **Assumption 3** in Section 4.2. The two datasets with dimensions m equal to 200 and 400 respectively were used. In both datasets, $\delta = 0.2$, $v = 1$ and $\|x_o\|_0 = 1$ were defined in which $n = \lfloor \delta m \rfloor$.

Figure 5.5 shows the EC ability of L_1 -minimizations in eqn. (5.1) using the proposed dimension reduction method, Naïve downsampling technique, UFSOL and SRCFS on the original input/observed data and dictionary with dimension $m = 200$. m' equals to 80, 60, 40, 20, 10 and 5. The ECs under changing portions of corrupted elements in the original

data ρ are shown. The figure shows that almost all ECs using the proposed method outperformed the ECs using the Naïve downsampling technique, UFSOL, and SRCFS in all cases. The difference in the ECs increased with the decrease of the final dimension m' . All ECs were similar when m' was very large. All ECs dropped very much when ρ was very large. Both ECs dropped with the decrease of m' . The ECs using the proposed method outperformed the ECs using Naïve downsampling, UFSOL and SRCFS in all the cases when $m' = 5$. The ECs using SRCFS is slightly better than the ECs using the proposed method when $m' = 20$ and 10. The ECs using the proposed method outperformed the ECs using Naïve downsampling, UFSOL, and SRCFS in general.

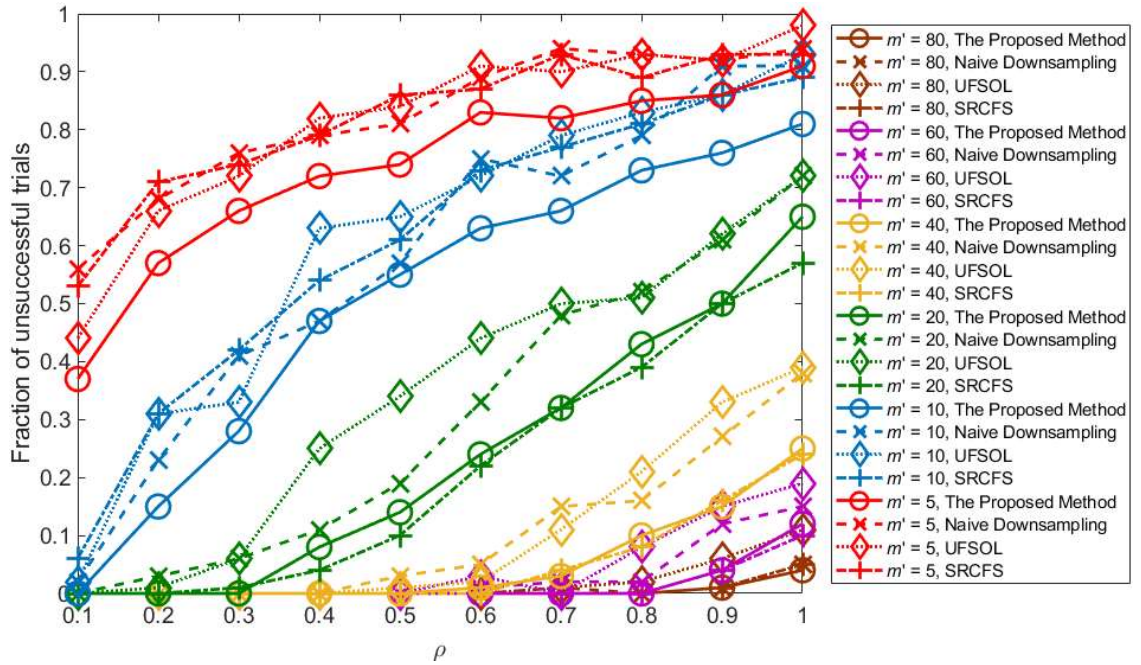


Figure 5.5 ECs under changing dimension m' and model parameter ρ

when $m = 200$

Figure 5.6 shows the EC ability of L_1 -minimizations in eqn. (5.1) using the proposed dimension reduction method, Naïve downsampling technique, UFSOL and SRCFS on the original input/observed data and dictionary which dimension $m = 400$. Almost all ECs using the proposed method outperformed the ECs using the Naïve downsampling technique, UFSOL, and SRCFS in all cases. The difference in the ECs increased with the decrease of the final dimension m' and converged to a constant when m' was small. All ECs dropped very much when ρ was very large. All ECs dropped with the decrease of m' . The ECs using

the proposed method outperformed the ECs using Naïve downsampling in all the cases when $m' = 5$, UFSOL, and SRCFS. The ECs using SRCFS is slightly better than the ECs using the proposed method when $m' = 20$ and 10. The ECs using the proposed method outperformed the ECs using Naïve downsampling, UFSOL, and SRCFS in general.

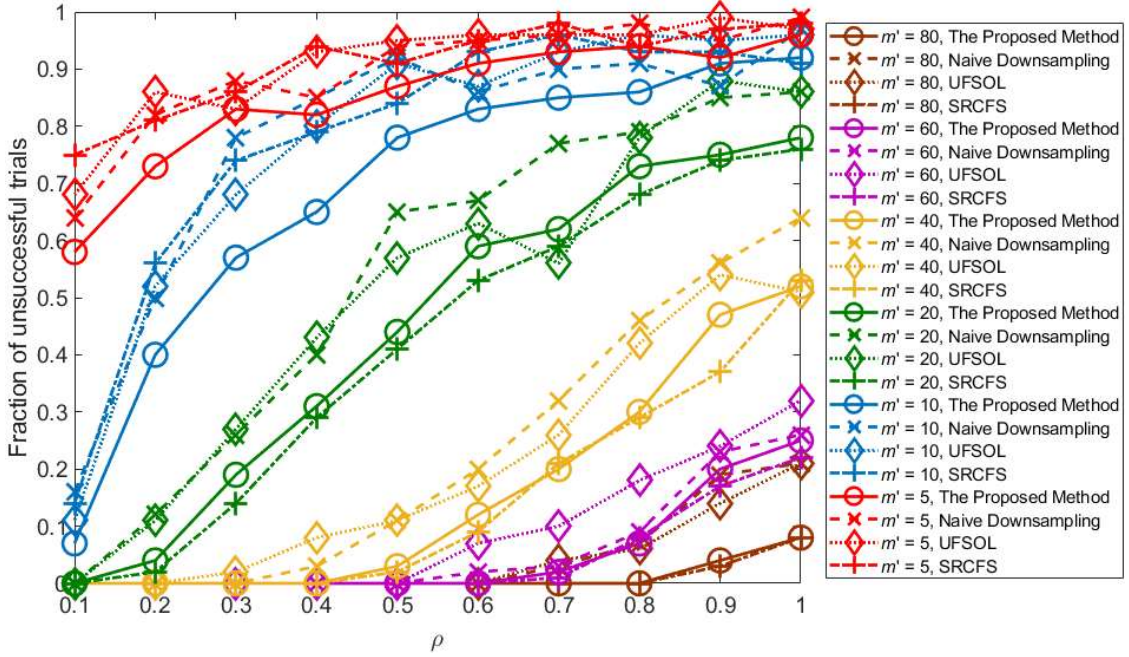


Figure 5.6 ECs under changing dimension m' and model parameter ρ

when $m = 400$

From all the above-mentioned results, it can be concluded that the overall EC ability using the proposed method is superior to the overall EC ability using Naïve downsampling technique in the simulations with artificial original data which fulfills the multi-bouquet model.

5.5.2. Simulations using Practical Data used in Face Recognition

In this subsection, the comparisons of the EC ability of L_1 -minimizations in eqn. (5.1) using the four following unsupervised feature selection methods on the original input/observed data and dictionary are presented:

- the proposed dimension reduction method,
- Naïve downsampling technique and

- the two state-of-the-art techniques, UFSOL, and SRCFS.

The data used was derived from the Extended Yale B face database [425] which is the same dataset used in Section 4.9.3. More details of the database can be found in Section 4.9.3. In the following results, 19 images of each chosen identities were used to construct the dictionary A .

Two pairs of m and N , as follows:

- $m = 30 \times 34$ and $N = 4$
- $m = 42 \times 48$ and $N = 9$

were used in the two experimental cases respectively in which

m is the product of height and width of the downsampled images before being vectorized, and

N is the subject number which is used to construct the dictionary A

$k_1 = 7$ (7 random elements of the ground truth sparse coefficient vector x_o are set to 1) was used in all cases.

UFSOL with the numbers of the nearest neighbors equal to 3 and cluster number equal to 10 (for the first pair of m and N) 20 (for the second pairs of m and N) are used. SRCFS with the numbers of the nearest neighbors equal to 3 and both the numbers of random subspaces in each basic feature partition and basic feature partitions equal to ceiling of 10 (for the first pair of m and N) 20 (for the second pairs of m and N) are used.

Figure 5.7 shows the EC ability of L_1 -minimizations in eqn. (5.1) using the proposed dimension reduction method, Naïve downsampling technique, UFSOL and SRCFS. The original input/observed data and dictionary were constructed with $m = 30 \times 24$ and $N = 4$. m' equals to 800, 400, 200 and 100 were used respectively. The ECs under changing portions of corrupted elements in the original data ρ are shown. The figure shows that almost all ECs using the proposed method outperformed the ECs using the Naïve downsampling technique, and SRCFS except the UFSOL in all cases. The difference in the ECs increased with the decrease of the ρ . The difference converged at $\rho = 1$. The ECs dropped with the decrease of m' . The ECs using the proposed method outperformed the

ECs using Naïve downsampling, and SRCFS in general except the UFSOL. UFSOL may be better suited for tackling the feature in this face recognition cases as it has performed worse than the proposed method for artificial data and another practical data presented in next section.

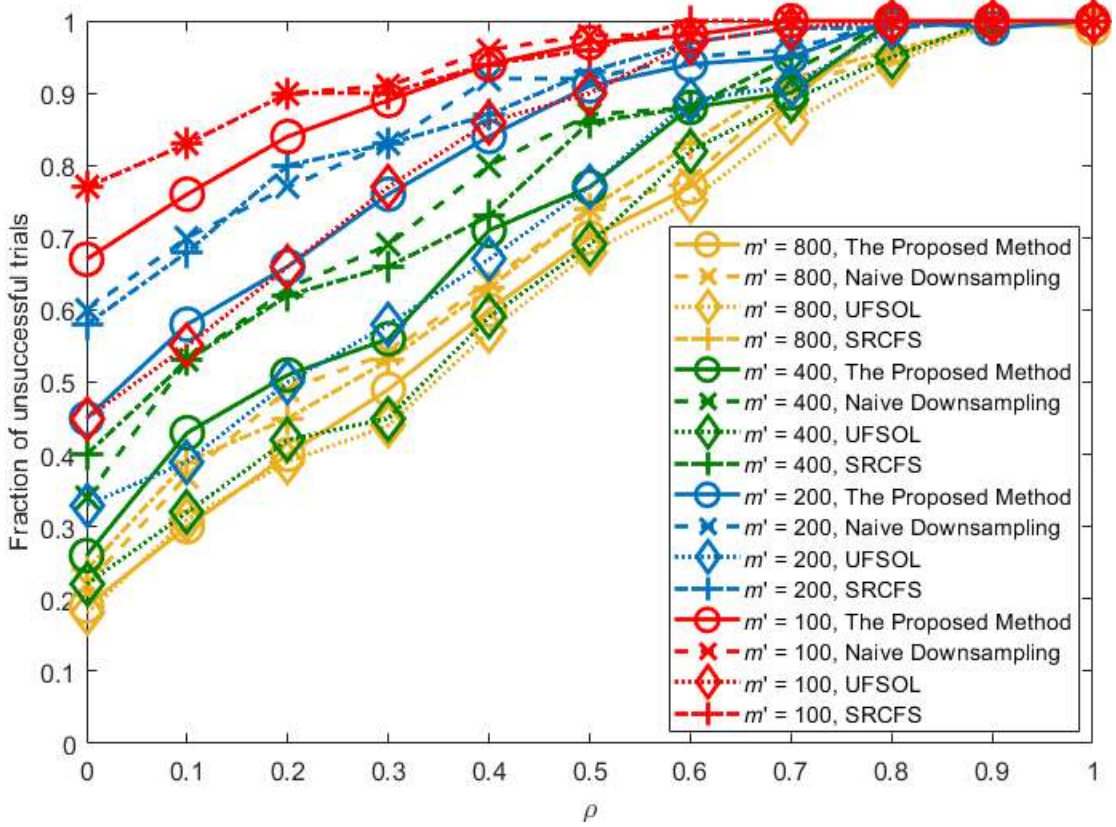


Figure 5.7 ECs under changing dimension m' and model parameter ρ when $m = 30 \times 24$ and $N = 4$

Figure 5.8 shows the EC ability of L_1 -minimizations in eqn. (5.1) using the proposed dimension reduction method, Naïve downsampling technique, UFSOL and SRCFS. The original input/observed data and dictionary were constructed with $m = 42 \times 48$ and $N = 9$. m' equals to 1600, 800, 400 and 200 were used respectively. The ECs under changing portions of corrupted elements in the original data ρ are shown. The figure shows similar observations with Figure 5.7. The figure shows that almost all ECs using the proposed method outperformed the ECs using the Naïve downsampling technique, and SRCFS except the UFSOL in all cases. The difference in the ECs generally increased with the decrease of the ρ . The difference converged at $\rho = 1$. The ECs dropped with the decrease

of m' . The ECs using UFSOL is not stable when m' is low (e.g. $m' = 100$). The ECs using the proposed method outperformed the ECs using Naïve downsampling, and SRCFS in general except the UFSOL. UFSOL may be better suited for tackling the feature in this face recognition cases as it has performed worse than the proposed method for artificial data and another practical data presented in next section.

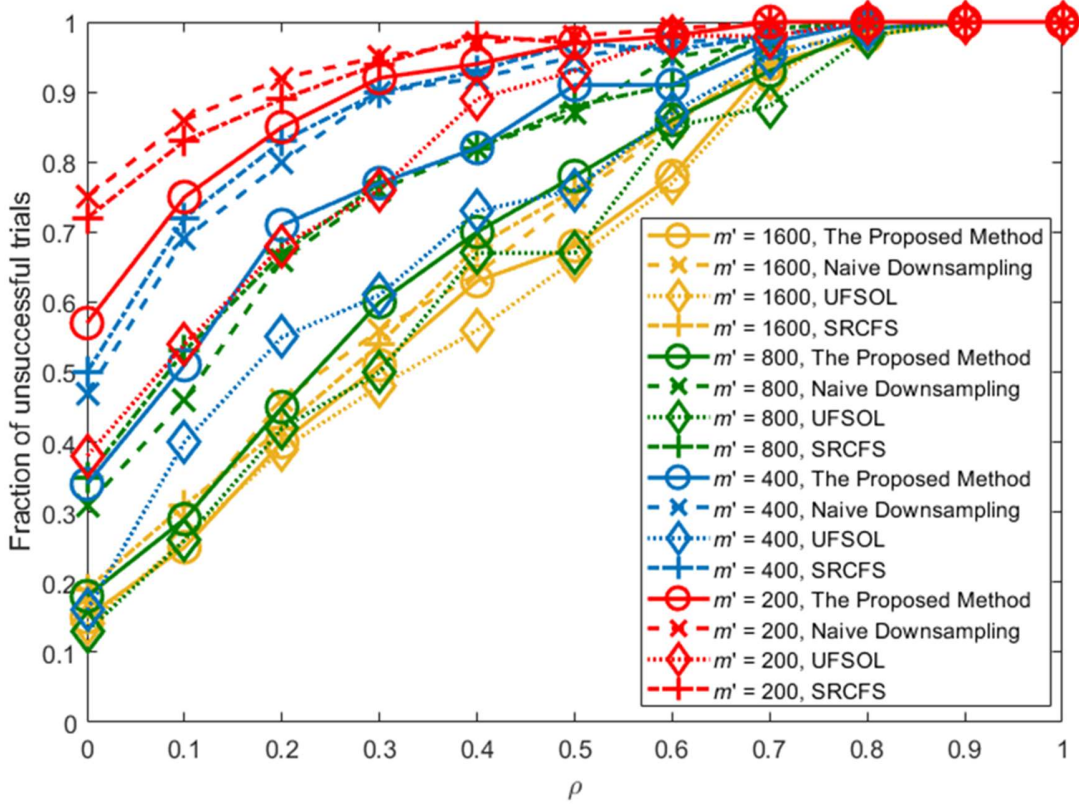


Figure 5.8 ECs under changing dimension m' and model parameter ρ when $m = 42 \times 48$ and $N = 9$

From all the above-mentioned results, it can be concluded that the overall EC ability using the proposed method is superior to the overall EC ability using Naïve downsampling technique, and SRCFS except the UFSOL in the simulations with practical data which is used in face recognition and fulfills the multi-bouquet model. Since UFSOL performed worse than the propose method for artificial data and another practical data presented in next section. We conclude that UFSOL may be better at tackling the feature in this face recognition cases.

5.5.3. Simulations using Practical Data used in Viewpoint Classification

In this subsection, the comparisons of the EC ability of L_1 -minimizations in eqn. (5.1) using the four following unsupervised feature selection methods on the original input/observed data and dictionary are presented:

- the proposed dimension reduction method,
- Naïve downsampling technique and
- the two state-of-the-art techniques, UFSOL, and SRCFS.

The data used was derived from the Rolls-Royce database [5], [6] which is the same dataset used in Section 4.9.4. More details of the database can be found in Section 4.9.4. In the following results, 19 images of each chosen viewpoint were used to construct the dictionary A .

Two pairs of m and N , as follows:

- $m = 30 \times 34$ and $N = 4$
- $m = 42 \times 48$ and $N = 9$

were used in the two experimental cases respectively in which

m is the product of height and width of the downsampled images before being vectorized, and

N is the viewpoint number which is used to construct the dictionary A

$k_1 = 7$ (7 random elements of the ground truth sparse coefficient vector x_o are set to 1) was used in all cases.

UFSOL with the numbers of the nearest neighbors equal to 3 and cluster number equal to ceiling of 10 (for the first pair of m and N) 20 (for the second pairs of m and N) are used. SRCFS with the numbers of the nearest neighbors equal to 3 and both the numbers of random subspaces in each basic feature partition and basic feature partitions equal to ceiling of 10 (for the first pair of m and N) 20 (for the second pairs of m and N) are used.

Figure 5.9 shows the EC ability of L_1 -minimization in eqn. (5.1) using the proposed dimension reduction method, Naïve downsampling technique, UFSOL and SRCFS. The original input/observed data and dictionary were constructed with $m = 30 \times 24$ and $N = 4$. m' equals to 800, 400, 200 and 100 were used respectively. The ECs under changing portions of corrupted elements in the original data ρ are shown. The figure shows that almost all ECs using the proposed method outperformed the ECs using the Naïve downsampling technique, UFSOL and SRCFS in all cases. The difference in the ECs was high when ρ is near to 0.5 and was low when ρ was either near to 0 or 1. All ECs dropped with the decrease of m' . The ECs drop much more when ρ was near to 0.5 than the drop when ρ was either near to 0 or 1. The ECs changed from convex shape to concave shape when m' decreases. In general, the ECs using the proposed method outperformed the ECs using Naïve downsampling, UFSOL and SRCFS.

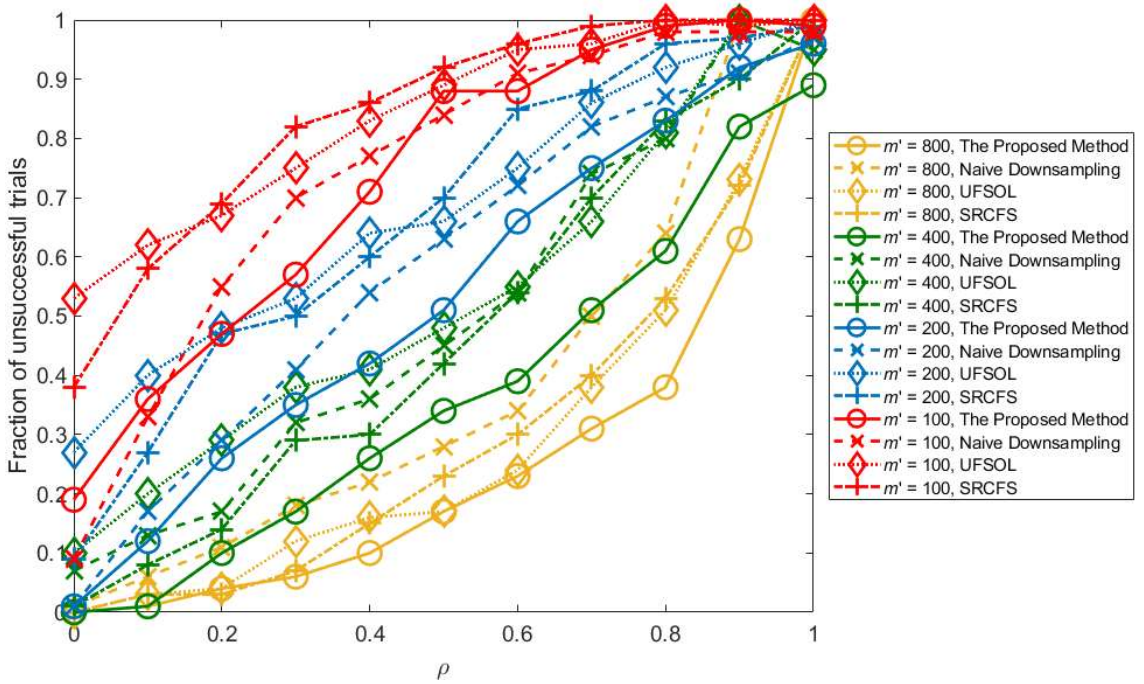


Figure 5.9 ECs under changing dimension m' and model parameter ρ when $m = 30 \times 24$ and $N = 4$

Figure 5.10 shows the EC ability of L_1 -minimization in eqn. (5.1) using the proposed dimension reduction method, Naïve downsampling technique, UFSOL and SRCFS. The original input/observed data and dictionary were constructed with $m = 42 \times 48$ and $N = 9$. m' equals to 1600, 800, 400 and 200 were used respectively. The ECs under changing

portions of corrupted elements in the original data ρ are shown. The figure shows that almost the ECs using the proposed method either outperformed (when m' is high) or nearly equal to (when m' is very low relative to m) the ECs using the Naïve downsampling technique, UFSOL and SRCFS in all cases. All ECs dropped with the decrease of m' . The ECs dropped much more when ρ was near to 0.5 than the drop when ρ was either near to 0 or 1. The ECs changed from convex shape to concave shape when m' decreased. Again, in general, the ECs using the proposed method outperformed the ECs using Naïve downsampling, UFSOL and SRCFS.

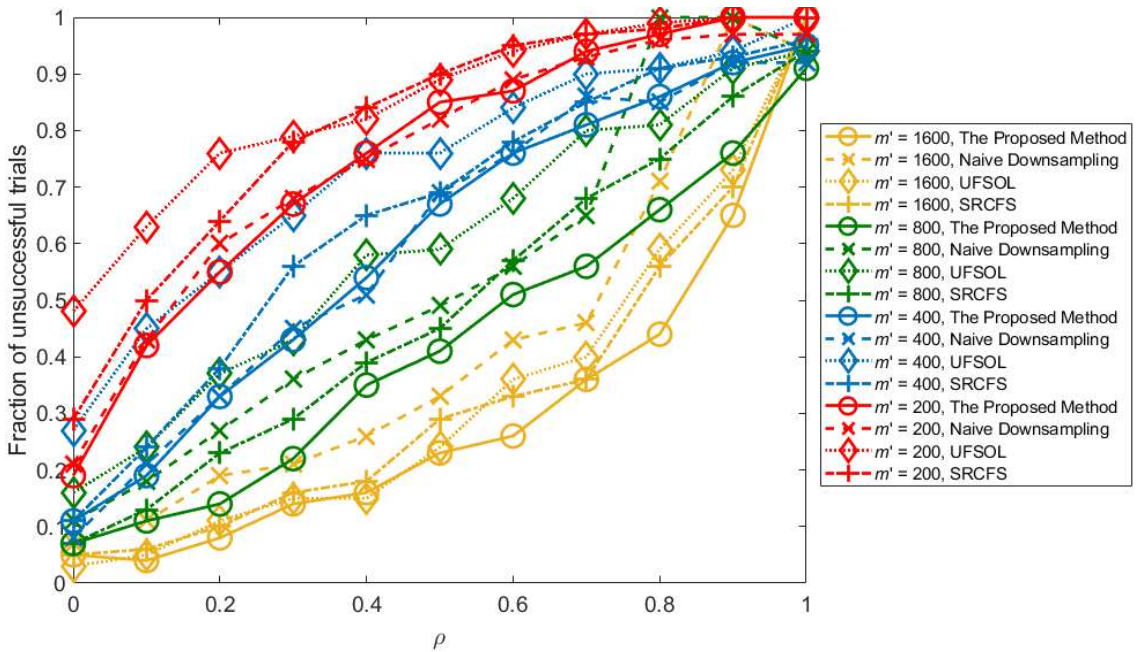


Figure 5.10 ECs under changing dimension m' and model parameter ρ when $m = 42 \times 48$ and $N = 9$

From the all above-mentioned results, it can be concluded that the overall EC ability using the proposed method is superior to the overall EC ability using Naïve downsampling technique, UFSOL and SRCFS in the simulations with practical data which is used in viewpoint classification and fulfills the multi-bouquet model.

5.6. Summary

This chapter presented a dictionary reconstruction using unsupervised feature selection technique to resolve a feasibility issue of sparse representation problem when computational resources are limited (see Figure 1.1 and Figure 1.2). An approach for

converting a nonfeasible dictionary-based problem e.g. eqn. (1.2) (due to the curse of dimensionality) to a feasible problem without retraining the dictionaries. The technique aims to preserve the outputs of the new problem similar to the ones derived from the infeasible high dimensional problem. The technique is highly practical for reusing existing dictionaries for the same problem in a computational unit that has lower capacity. The technique assumes the underlying relationships between features and the training process of dictionaries are unknown. A simple unsupervised feature selection technique which is a combination of unsupervised clustering and nearest neighbor approach is presented in the demonstration. Empirical results showed that the proposed technique is consistently superior to the naïve down-sampling techniques and which are frequently used for dimensional reduction especially in image-based problems and two state-of-the-art techniques. Next chapter presents the performance of an anomaly detection and localization approach which applies eqn. (1.2) in the problem design for a real-word problem is investigated.

Chapter 6

Anomaly Detection and Localization

“To study the abnormal is the best way of understanding the normal.”

- William James

In the previous two chapters,

- *a novel optimization technique for solving nonsmooth optimization problem which a lot sparse recovery problem involves,*
- *the theoretical derivation of successful sparse recovery probability bound using an L_1 -norm minimization, and*
- *a feature selection method for the dimension reduction of dictionary to make computation infeasible dictionary-based problem which a lot sparse recovery problem use to be feasible*

are presented. In this chapter, we presents an application of sparse recovery on anomaly detection and localization problem and demonstrates its performance on an industrial anomaly detection and localization problem.

This chapter introduces an approach which applies eqn. (1.2) in the problem design for a real-word anomaly detection and localization problem and investigate its performance, please refer to Figure 1.1 and Figure 1.2.

$$\begin{aligned} & \min_{x,e} \|x\|_1 + \lambda \|e\|_1 \\ & \text{s. t. } y = Ax + e \end{aligned} \quad \begin{array}{l} \text{Recap} \\ (1.2) \end{array}$$

where λ is a tradeoff parameter. Anomaly is defined as something which deviates from the standard, normal, or expected. It is also something which cannot be reconstructed from the templates which describes the standard. The

results in Chapter 4 states that the successful recovery probability of sparse coefficient vector x and error vector e through the L_1 -norm minimization in eqn. (1.2) is overwhelming when the dimension of the data y is high. Assuming the error vector recovered by the L_1 -norm minimization is correct. The anomaly which cannot be reconstructed by the templates stored in the dictionary A contains all the anomaly information. An approach which performs accurate anomaly detection and localization by applying the proper operations on the error vector is presented. A study of the performance of the proposed method on an application in automated visual inspection is also conducted. The empirical experimental results show that the proposed approach is potentially useful and outperforms the state-of-the-art saliency detection approaches.

In this chapter, Section 6.1 outlines how sparse reconstruction can be used in data decomposition with corrupted data. Section 6.2 presents how the sparse reconstruction approach has been successfully applied in face recognition. Section 6.3 introduces a proposed anomaly detection and localization approach using the sparse reconstruction. Section 6.4 presents the extension (SRA) of the anomaly detection and localization approach to an industrial application; namely: defect detection and localization for automated visual inspection (AVI). Section 6.5 presents an extended approach (extended SRA) which solves some critical issues of the proposed defect detection and localization approach (SRA). Section 6.6 explains theoretically how many images are needed to reconstruct the light reflected from a Lambertian surfaces in an image which are acquired under different illumination condition. Section 6.7 analyses the computation time and the memory demand

of the sparse reconstruction algorithms. Section 6.8 presents the empirical experimental results. Section 6.9 summaries the chapter.

6.1. Sparse Reconstruction with Noise

The main objective of sparse reconstruction is to decompose data into a useful form for specific usages such as face recognition [3] and compressed sensing [457]. One of the forms is a summation of a dot product of a sparse coefficient vector and a dictionary and a corrupting error. The original concept of sparse reconstruction was introduced in compressed sensing [457].

An m -dimensional input/observed data vector $y \in \mathbb{R}^m$ can be decomposed as in eqn.

$$(6.1) \quad y = Ax + e \quad (6.1)$$

where

$A \in \mathbb{R}^{m \times n}$ is a dictionary (or design matrix) with n atoms (or words) of size m ,
 $x \in \mathbb{R}^n$ is called sparse coefficient vector if it is sparse, and
 $e \in \mathbb{R}^m$ is the corrupting error (residuals).

The decomposition in eqn. (6.1) is called a sparse reconstruction (or representation/approximation/decomposition) when x is sparse. The corrupting error/residuals e is caused by anomalies, noise, image corruption, occlusion, shadow, etc. Note that **Theorem 1** in Chapter 4 shows that the sparse coefficient vector x and the corrupting error vector e can be recovered using the L_1 -norm minimization in the following eqn. (1.2) with overwhelming probability under the assumption stated in the theorem including the cases which the portions of the corrupted elements of the input/observed data are very high.

$$\begin{aligned} & \min_{x,e} \|x\|_1 + \lambda \|e\|_1 \\ \text{s. t. } & y = Ax + e \end{aligned} \quad \begin{array}{l} \text{Recap} \\ (1.2) \end{array}$$

where λ is a tradeoff parameter.

The L_1 -norm of the sparse coefficient vector x is used to learn the sparse x . The L_1 -norm of the corrupting error/residual is used so that the L_1 -norm minimization can recover sparse error although it is also able to recover dense error (as stated in **Theorem 1** in Chapter 4). λ controls the tradeoff between the two L_1 -norms and is dependent on the problem dependent characteristics of the original/ground truth sparse coefficient vector x_o and the original/ground truth corrupting error e_o .

6.2. Sparse Reconstruction Approach for Face Recognition

Basic face recognition attempts to estimate the identity of the face presented in an input/observed image. A face recognition approach was proposed by Wright et al. [3] using sparse reconstruction. The L_1 -norm minimization in the following eqn. (1.1) (which is the eqn. (1.2) with $\lambda = 1$) is used to determine the input/observed image's sparse coefficient vector x .

$$\begin{aligned} \min_{x,e} & \|x\|_1 + \|e\|_1 \\ \text{s.t. } & y = Ax + e \end{aligned} \quad \begin{array}{l} \text{Recap} \\ (1.1) \end{array}$$

The sparse coefficient vector x is used for identifying the identity of a human face in which the frontal view is presented in the input/observed image. In this problem, the input/observed data y is formed by stacking columns or rows of the input/observed image. The dictionary A stores the information of all labelled training images and each of its atoms is formed by stacking columns and rows of a normalized training image. All atoms are labelled with the human identities. A visualization of sparse decomposition of an input/observe image is presented in Figure 6.1. The estimated sparse coefficient vector x learnt from the L_1 -norm minimization in eqn. (1.1) is expected to have only a few non-zero elements. The non-zero elements are expected to correspond to the dictionary's atoms which are labelled with the identical identity.

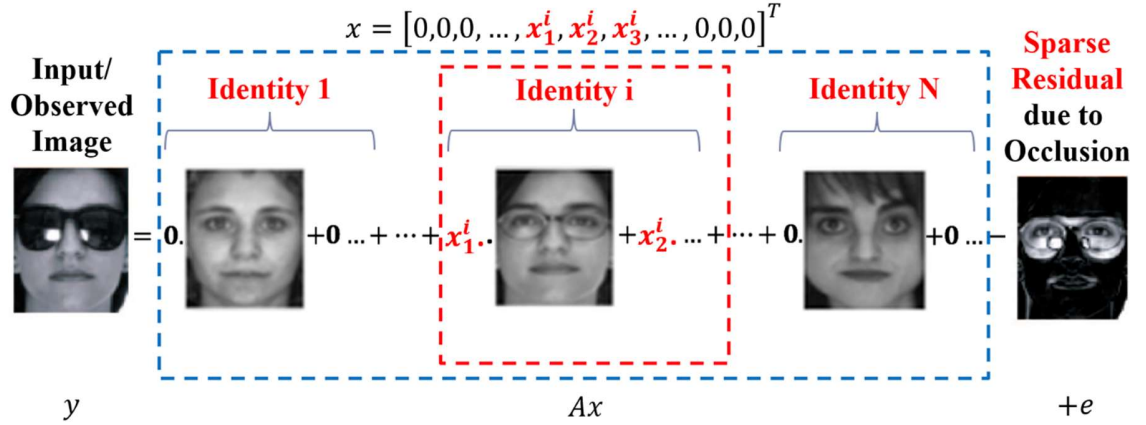


Figure 6.1 Image-based sparse decomposition for face recognition

The proposed anomaly detection and localization approach which is described in the following section is inspired by the image-based sparse decomposition.

6.3. Proposed Anomaly Detection and Localization Approach

Inspired by the image-based sparse decomposition described in the previous section, a sparse reconstruction approach for anomaly detection and localization for numeric data type is proposed and is described in this section.

The basic idea of using sparse reconstruction for anomaly detection and localization is to reconstruct an “anomaly-free” template (the outcome under one of the normal situations). The “anomaly-free” template is further to be used for template matching against the input/observed data which may be time series, image or any numeric type of data. It determines the residuals between the expected “anomaly-free” (normal) template and the input/observed data. Anomalies are potentially located at the residuals. The main motivation of using sparse reconstruction to detect the anomalies are as follows:

- a data that does not contain anomalies is ideally reconstructed using sparse reconstruction with zero residual when the dictionary consists of atoms that describe the normal situations, and
- a data that contains anomalies is represented by a sparse coefficient vector together with the sparse residuals which cannot be reconstructed.

The input/observed image used in anomaly detection and localization can be decomposed into a weighted combination of atoms of a dictionary and residuals. The information in the residuals is mainly the anomaly information. Providing an input/observed image and a dictionary A , the sparse coefficient vector x and the residuals e can be estimated through the sparse reconstruction approach with the L_1 -norm minimization in eqn. (1.2). The anomaly can be detected and localized by applying appropriate data post-processing operations (such as thresholding etc.) on the estimated residuals. Figure 6.2 demonstrates the idea of using proposed sparse reconstruction approach to reveal the anomaly information. The anomaly information is stored in the residuals determined through applying the idea of the proposed approach on the image data using the L_1 -norm minimization in eqn. (1.2). In the figure, the input/observed data y is the vectorized input/observed downsampled image. The dictionary A stores the global information of anomaly-free (normal) samples. Each atom in the dictionary A is a vectorized normalized downsampled anomaly free (normal) image. In the example shown in Figure 6.2, the anomaly is a defect on a metallic item and the anomaly-free data are the images of a metallic item acquired under different viewpoints and different illumination conditions. The anomaly information can be used for the application of defect detection and localization of automated visual inspection. In the following sections, the case studies of using the proposed anomaly detection and localization approach for the inspection application will be shown.

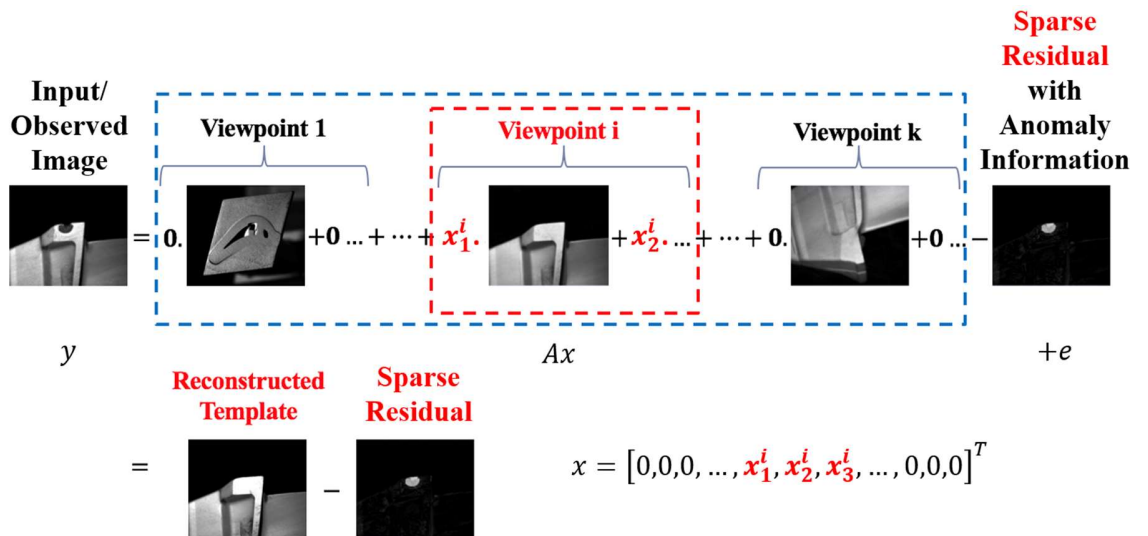


Figure 6.2 Demonstration of anomaly information in the residuals which can be determined using sparse reconstruction approach

6.4. Sparse Reconstruction Approach (SRA) for Defect Detection and Localization

In the previous section, a proposed method using sparse reconstruction approach (SRA) for anomaly detection and localization is elaborated. Moreover, an example of the proposed approach with image data is shown in the Figure 6.2.

In this section, the extension of the anomaly detection and localization approach for an inspection application which is the aircraft turbine blade surface defect detection and localization [5].

The task is highly challenging with the following characteristics:

- highly complex surfaces' shape (edges and curvatures without global repeated pattern and global uniformity),
- diverse defects' characteristics (shape, size and form),
- multiple viewpoints (all facets need to be inspected), and
- various illumination conditions (appearance of defect affected by lighting condition)

The SRA for defect detection and localization contains two steps:

- use sparse reconstruction to get the residuals which contain the defect information
- learn the classifier to detect and localize defects from the residuals.

The intuition for the proposed approach is that defects are unwanted portions of an image reconstruction. A two-step greedy search is used for parameter learning.

In Section 6.4.1, how to use sparse reconstruction to derive residuals and residual image is shown. Section 6.4.2 describes how the residuals are used to classify the pixels in the residual image into defective and normal pixels. Section 6.4.3 presents a way of finding the best parameters used in the proposed SRA. Section 6.4.4 analyse how each parameter affects the performance behaviour of the proposed SRA.

6.4.1. Sparse Reconstruction

Given sufficient downsampled defect-free training images which are taken from i^{th} viewpoints under various illumination conditions, a matrix $A_i = [a_1^i, \dots, a_{N_i}^i] \in \mathbb{R}^{m \times N_i}$ can be constructed. Each atom a_j^i in A_i is a vectorized normalized downsampled defect-free training image. For a training dataset with N viewpoints, a dictionary $A = [B_1 | \dots | B_N] \in \mathbb{R}^{m \times n}$ can be constructed by concatenating N matrix B_i . The sparse coefficient vector x and residual image e of the downsampled input/observed image which is acquired from one of the N viewpoints under a particular illumination can be estimated using the dictionary A , through the L_1 -norm minimization in the following eqn. (6.2) which is equivalent to the eqn. (1.2).

$$\begin{aligned} \min_{x,e} & \|x\|_1 + \|e\|_1 \\ \text{s.t. } & y = Ax + \lambda e \end{aligned} \quad (6.2)$$

where λ is a tradeoff parameter.

Note that eqn. (6.2) will be used in the following subsections in determining the representation x and residual image e . The estimated sparse coefficient vector is expected to be sparse and only contain non-zero elements which correspond to the atoms which are acquired under identical viewpoint. After applying a thresholding classifier on each pixel of residuals, the estimated anomalies can be detected and their estimated locations can be known too. Note that the defect to be detected and localized must have the information which is presented in the input/observed image. Figure 6.3 shows the flow diagram of using sparse reconstruction approach for the defect detection and localization. In the final results, the white and black pixels represent the labels of defective and normal pixel respectively.

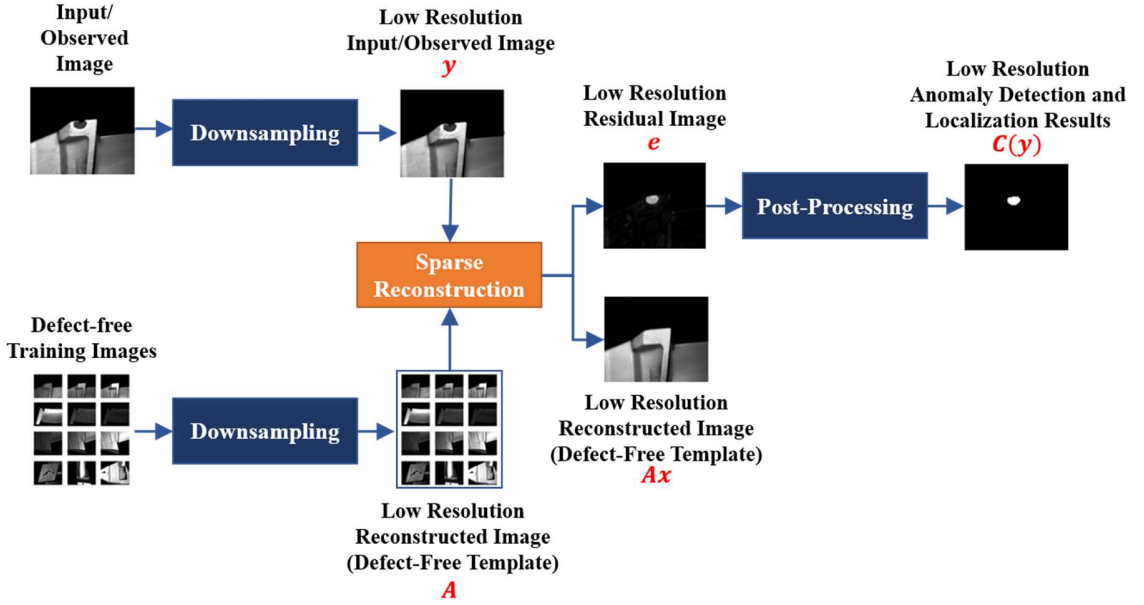


Figure 6.3 Defect detection and localization using sparse reconstruction approach

The defect to be detected and localization is the surface defect which is exposed on the surface of the product and is visible from the images.

In the following subsection, a classifier which is proposed and used for the pixel classification (defective and normal pixel classification) is elaborated.

6.4.2. Defect Detection and Localization

In this subsection a thresholding classifier and its training process is presented.

After an estimated residual image $\hat{e}$ is determined, the simple pixel-based threshold classifier in eqn. (6.3) can be used to classify each pixel to be either defective or normal pixel.

$$C(y_i) = \begin{cases} 1, & |\hat{e}_i| - \bar{T} > 0 \\ 0, & |\hat{e}_i| - \bar{T} \leq 0 \end{cases} \quad (6.3)$$

where

y_i is the i^{th} element/pixel value in the low resolution input/observed image y

- $\hat{e}_i$ is the estimated residual value which corresponds to the location of the i^{th} element/pixel, and
- $\bar{T}$ is the threshold value which is used to classify the i^{th} element/pixel to be either defective or normal element/pixel.

$C(y_i)$ returns 1 when the i^{th} pixel is predicted to be one of the pixels where the defect is located. It is based on the absolute residual value $|e_i|$ derived from the L_1 -norm minimization in the eqn. (6.2) and a predefined threshold $\bar{T}$. $C(y_i)$ returns 0 when the i^{th} pixel is predicted to be normal. The main challenge is to select the most discriminative threshold $\bar{T}$.

Although the defect detection and localization task is formulated as a binary classification task (See Figure 6.3), it is similar to the task of one-class classification with outliers. Hence, the discriminative threshold $\bar{T}$ can be learnt using a simple variation of the one-class Support Vector Machine (one-class SVM) [458] as the optimization problem in the following eqn. (6.4).

$$\begin{aligned} \min_{\bar{T}} \quad & \bar{T} + \frac{1}{\eta} \sum_{i,j} \xi_{i,j} \\ \text{s. t.} \quad & \bar{T} - |\hat{e}_{i,j}| \geq -\xi_{i,j}, i = 1 \dots N_t, j = 1 \dots m, \\ & \xi_{i,j} \geq 0, \bar{T} \geq 0 \end{aligned} \quad (6.4)$$

where

- N_t is the total number of training images for training the threshold $\bar{T}$,
- $\xi_{i,j}$ are the slack variables corresponding to the residual $|\hat{e}_{i,j}|$ for j^{th} pixel in i^{th} image, and
- η is a parameter controls the tradeoff between the threshold $\bar{T}$ and the total sum of slack variables $\xi_{i,j}$.

In the following subsection, a way of finding the best parameters λ in eqn. (6.2) and η in eqn. (6.4) and the threshold $\bar{T}$ for the classifier in eqn. (6.3) is presented.

6.4.3. Finding Best λ , η and $\bar{T}$

To achieve a competitive detection performance, one needs to determine the best λ , η and $\bar{T}$ using the training images whose pixels are marked either as defective or normal pixels.

Algorithm E.1 in Appendix E shows one of the procedures to derive these parameters. Given a dictionary A , the inputs to the algorithm consist of the set of labeled training images I_t , the stopping criterion S for the procedure, and the search granularity N that affects the computational time of the algorithm. In the algorithm balanced accuracy which is defined in eqn. (6.5) is used as the performance measure.

$$Acc_b = \frac{1}{2} \left(\frac{TP}{P} + \frac{TN}{N} \right) \quad (6.5)$$

where TP and TN are the number of true positive pixels and true negative pixels, respectively. P and N are the numbers of positive and negative test pixels, respectively. Acc_b is the average between the sensitivity and specificity measures.

Note that the best λ and η for the defect detection and localization approach is highly dependent on the dictionary. Hence, **Algorithm E.1** needs to be rerun whenever the dictionary is changed. It is also noted that the focus of this chapter is not on the learning of the dictionary. In Section 6.8.1, the performance of the proposed approach on different fixed and learned dictionaries are investigated.

Algorithm E.1 can be considered as a two-step greedy search first on λ and then on η . The training accuracy, in general, is concave with respect to λ and η . However, it is non-smooth locally. Hence, a greedy approach is utilized to determine a near-optimal λ and η that achieve good accuracy. The best λ , η and $\bar{T}$ can also be estimated by using more time consuming but more accurate approach such as the particle swarm algorithm (PSO) [459]–[461].

6.4.4. Parameter Analysis

In Section 6.4.3, a learning method for selecting the best parameters for the defect detection and localization model to achieve good detection performance has been presented. In this subsection, the effects of parameters, λ and η , in sparse reconstruction of input/observed images using L_1 -norm minimization in eqn. (6.2) are analyzed.

A. Sparse Reconstruction Parameter, λ

Suppose one is given a dictionary consisting of the three normalized $m \times m$ images (see the three images on the left-hand side of Figure 6.4). The right input/observe image in Figure 6.4 is reconstructed with an anomalous horizontal bar.

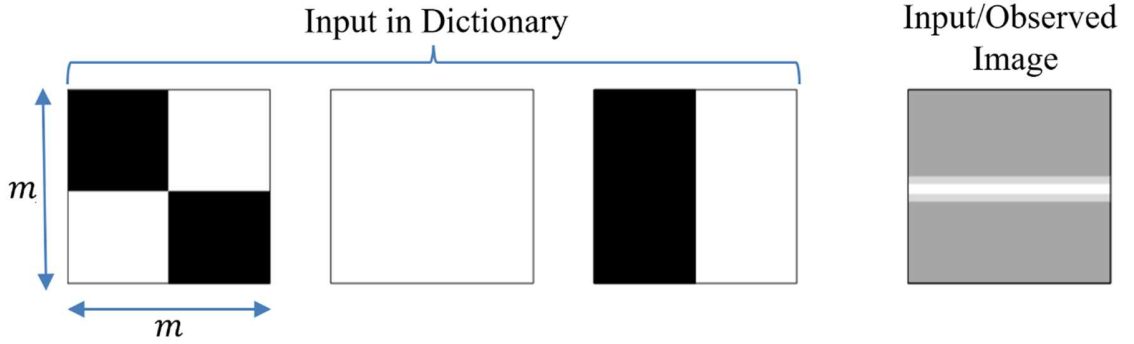


Figure 6.4 The examples of training images for constructing a dictionary and the input/observed image

B. $\lambda = 1$

Note that when the sparse coefficient vector $x = [0,0,0]$, the sum of all absolute residual values ($\|e\|_1$) is the sum of pixel values ($\|y\|_1$) in the input/observed image. When the sparse coefficient vector $x = [0,0.5,0]$, most pixel residual values become zero except those that are likely to be anomalous. When the input/observed image has a small anomalous portion with a non-significant sum of the absolute discrepancy between anomalous portions and the normal pattern ($\sum_j |e_i|$, j are indexes of anomalous pixels), the reconstruction can be effective with $\lambda = 1$. However, the reconstruction may be bad if either the anomalous portion or the sum of the discrepancy between the anomalous portion and normal pattern is large. The L_1 -norm minimization in eqn. (6.2) will minimize the sum of absolute residuals ($\|e\|_1$) with increasing the residuals at normal portion and decreasing the residuals at anomalous residuals. Hence, the anomalous regions which have low discrepancy from the normal pattern will be misclassified.

C. Extreme λ

If a very small value of λ is used, the residual values are amplified. The objective function in eqn. (6.2) is approximately the sum of absolute residual values ($\|e\|_1$). The sparsity enforcement on sparse coefficient vector x is lost. However, in most cases, there are only

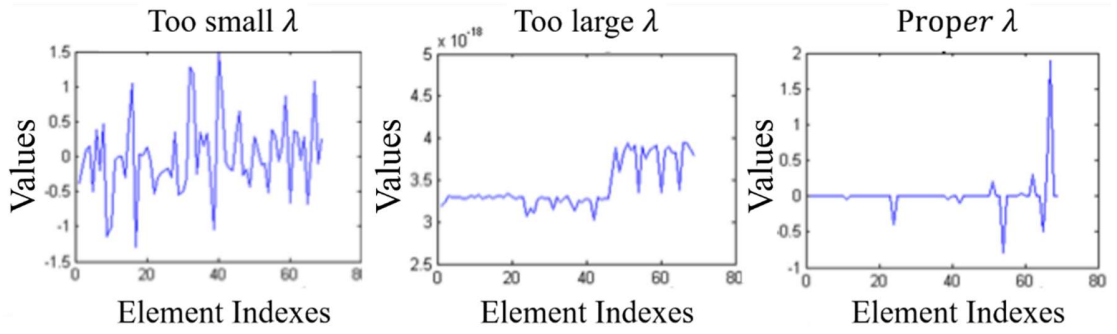
a few atoms which are relevant to the input/observed image in the dictionary to reconstruct the input/observed image. Hence, the loss of the sparsity enforcement on sparse coefficient vector x may cause an improper reconstruction due to the overfitting produced by using too many atoms in the dictionary. Hence, the anomaly detection performance drops. Unless all atoms in the dictionary are relevant to the input/observed image such as the normal images used in background subtraction problem for surveillance image sequence. Figure 6.5 shows an example of a reconstructed image and its corresponding contrast stretched residual image (all pixel values are absolute residual values) with a very small λ .



Figure 6.5 Reconstruction results using very small λ

When a large value is assigned to λ , the objective function in eqn. (6.2) approximates the sum of absolute values ($\|x\|_1$) of all elements of the sparse coefficient vector x . Hence, the sum of absolute residual values ($\|e\|_1$) will be ignored and the optimized a vector will be a zero vector. It causes the pixel classification of the residual image to have the same results as the pixel classification on input image directly and the classification is meaningless. Figure 6.6 shows an example of a reconstructed image and its corresponding amplified residual image (all pixel values are absolute residual values) with very large λ .

Figure 6.7 shows the element values of three different sparse coefficient vectors x which are estimated using the L_1 -norm minimization in eqn. (6.2) with different λ . The figure shows that when λ is too small, the sparse coefficient vector x is not sparse. On the other hand, when λ is too large, all element values of the sparse coefficient vector x are very small. A reasonable λ needs to be selected to ensure sparsity is achieved for the sparse coefficient vector x .

Figure 6.6 Reconstruction results using very large λ Figure 6.7 The estimated sparse coefficient vectors x with different λ

D. η

η provides an estimate of the number of defects in an image. Too high or too low η can dramatically affect defect detection performance. Large η causes the sum of $\xi_{i,j}$ ($\sum_{i,j} \xi_{i,j}$) in the minimization in eqn. (6.4) to be insignificant. The optimal threshold $\bar{T}$ (learnt from the minimization in eqn. (6.4)) tends to zero and all the pixels are classified as defective pixels. Low η causes the threshold $\bar{T}$ to increase and cause most of the defective pixels to be classified as normal pixels as there are only very few pixels having absolute residual values which are higher than the threshold. The best way to choose the parameter η is to make the values slightly larger than the largest expected total number of defective pixels in the training input/observed images. Again, this parameter depends on the dataset and it can either be determined by a domain expert or using machine learning techniques such as the proposed **Algorithm E.1** to derive it from a training dataset.

E. Effect of Parameters on Defect Detection Performance

In the following analysis, the balanced accuracy (Acc_b) [462] which is defined in eqn. (6.5) is used as the evaluation measure.

It was noted earlier that accuracy becomes worse as λ becomes smaller or when λ is too large. A suitable λ is between the extremely small and extremely large λ . Hence, the relationship between performance accuracy and λ has a concave trend in general. When λ approaches the optimal λ , the input/observed image is better reconstructed and the absolute residual values at defective pixels become more significant and the absolute residual values at the normal pixels are sparser. Therefore, better anomaly detection performance can be achieved.

The similar concave trend is also observed in the accuracy with respect to η across the η domain. The learned classification threshold $\bar{T}$ is zero for extremely large η . This threshold value approaches the largest absolute residual value for extremely small η . Using either one of the two extreme thresholds degrades the defect detection accuracies Acc_b to near 0.5, which is equivalent to random guess. Hence, the threshold learned from the minimization in eqn. (6.4) should be between these two extreme values to achieve a better Acc_b . Again, Acc_b with respect to η has a concave trend.

Note that the observations are based on the assumption that the images under the input/observed images' viewpoints are used in the dictionary construction. If there are no such images used in dictionary construction, the reconstructed image is not a representative "defect-free" template. As a result, the residuals derived from the sparse reconstruction does not help in defect detection and localization.

Even though one observes a concave trend in general for Acc_b in the λ and η domains, this concave trend is not locally observed. As mentioned earlier, when λ approaches the optimal λ , the reconstructed image gets better. The sparse coefficient vector x element values corresponding to the correct atoms (or images) in the dictionary become larger and the ones corresponding to the wrong atoms in the dictionary are reduced. Ideally, Acc_b is concave.

However, in reality, all images and atoms in the dictionary have noise and this affects this concave trend. Hence, some pixels are wrongly classified as their residuals are affected due to the use of wrong atoms or noises within the correct atoms or the input/observed image itself during the sparse reconstruction. As a result, defect detection performance varies randomly within a short range of λ resulting in a locally non-concave Acc_b . Another cause of the non-concave Acc_b is the failure of reconstruction due to the lack of illumination information in the dictionary A . Figure 6.8 (second left) shows how Acc_b varies with λ . One observes that Acc_b is locally noisy but it has a concave trend. One also observes from the rest of the graphs, the relationships of $\bar{T}$, classification accuracies of defective pixels and normal pixels and λ in Figure 6.8. The relationship between Acc_b and η is similar to the relationship between Acc_b and λ with a smoother trend (see Figure 6.9). With such general (and local) concave trends between Acc_b and η , and between Acc_b and λ , the greedy search algorithm in **Algorithm E.1** described in Section 6.4.3 and Section E.1 in Appendix E is used to find the best parameters, λ and η .

Figure 6.9 also shows the relationships of the classification threshold $\bar{T}$, classification accuracies of defective pixels and normal pixels and η .

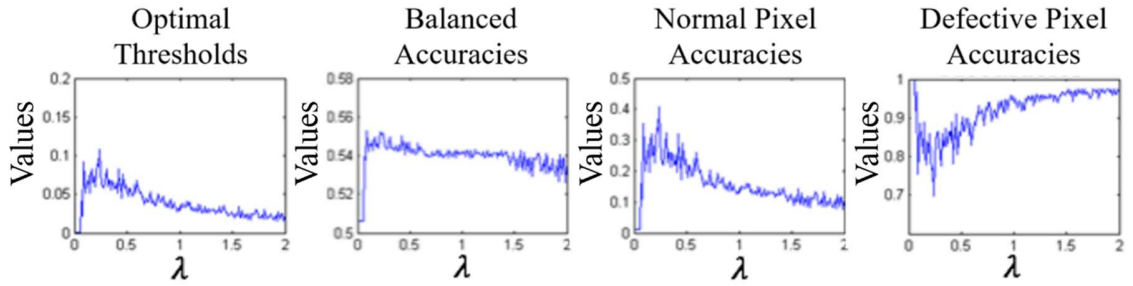


Figure 6.8 Accuracies and the optimal threshold $\bar{T}$ versus λ at a fixed η

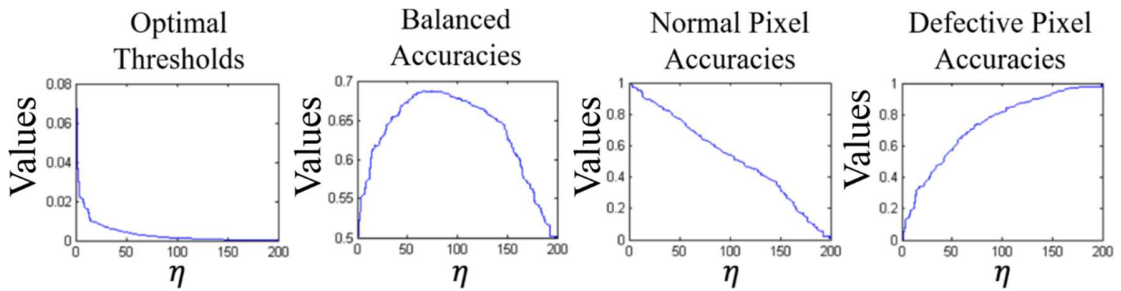


Figure 6.9 Accuracies and the optimal threshold $\bar{T}$ versus η at a fixed λ

6.5. Extended SRA for High-Resolution Images

The SRA approach [5] introduced in the previous section have two critical issues:

- large computation, and
- high memory demand.

To resolve these two issues, images need to be downsampled to be at low resolution. However, it comes with the tradeoff that the resultant results are also at low resolution and, hence the defect detection and localization is not precise and accurate.

In this section, an extended SRA [6] is proposed to resolve these critical issues. Four novel new improvements for the original SRA:

- high-resolution results which are not feasible to the original SRA can be generated,
- the flexibility of free adjustments of the computation and the memory demand is provided,
- more operations (not only thresholding) are applied on the high-resolution residual image, and
- a new proposed illumination adaptive threshold is used.

The new proposed extended SRA consists of two main steps:

- fast sparse reconstruction, and
- defect Detection and Localization

The first step aims to generate accurate high-resolution residual values. The second step learns a classifier and uses it to produce the final high-resolution defect detection and localization results. Figure 6.10 illustrates the flow chart describing the methodology of the proposed extended SRA.

In Section 6.5.1, the fast sparse reconstruction is elaborated. Section 6.5.2 presents how the classifier of the proposed extended SRA is to be learnt and how to use it for defect detection and localization. Section 6.5.3 introduces a way of learning the best parameter of the proposed approach.

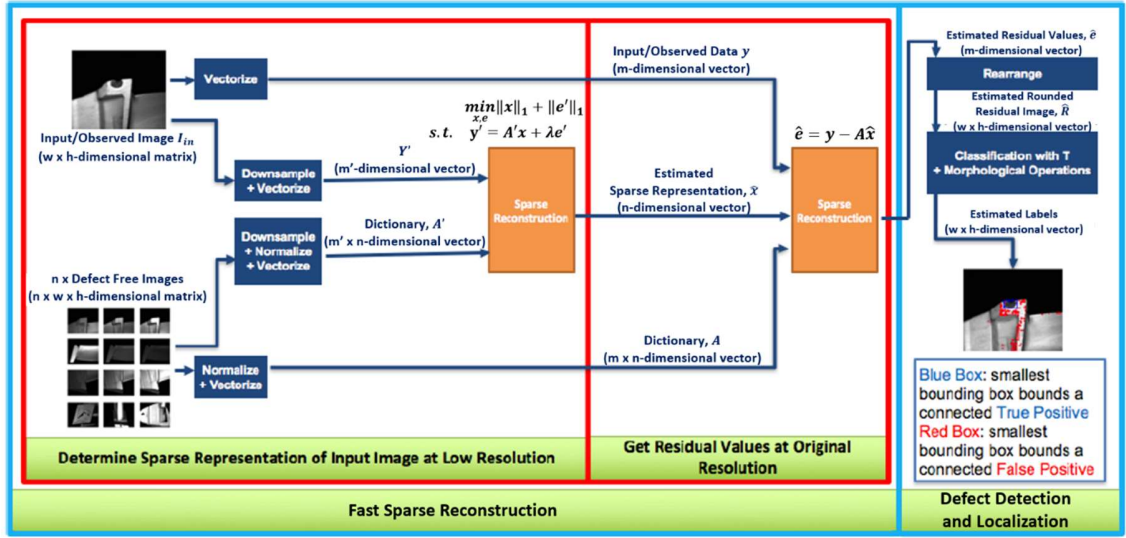


Figure 6.10 The flow chart of the extended SRA

6.5.1. Fast Sparse Reconstruction

In this subsection, the first step of the extended SRA is described. The step generates a high-resolution residual image for the defect detection and localization step which will be described in the next subsection.

The fast sparse reconstruction consists of two sparse reconstructions:

- the first reconstruction estimates the sparse coefficient vector x of the input/observed image with low resolution (low downsampled resolution) images, and
- the second reconstruction generates the high-resolution residual values with the estimated sparse coefficient vector $\hat{x}$ and the high resolution (original resolution or high downsampled resolution) images.

Note that the new proposed fast sparse reconstruction allows the free adjustment of computational time and memory demand in its optimization process. It resolves the bottleneck of the original SRA [5].

Given a $w \times h$ -dimensional high-resolution input/observed image and its downsampled $w' \times h'$ -dimensional lower resolution input/observed image, two high and low-resolution input/observed data, $y \in \mathbb{R}^m$ and $y' \in \mathbb{R}^{m'}$ can be formed by concatenating the element in

the two images either row-wise or column-wise. The input/observe data y' can be reconstructed by a dot product of a dictionary/design matrix (with m' rows/features and n atoms/words) $A' \in \mathbb{R}^{m' \times n}$ and a sparse coefficient vector $x \in \mathbb{R}^n$ and residuals $e' \in \mathbb{R}^{m'}$ which are caused by anomalies, noise, image corruption, occlusion, shadow, etc. as shown in eqn. (6.6).

$$y' = A'x + e' \quad (6.6)$$

The first sparse reconstruction estimates the sparse coefficient vector x in eqn. (6.6) through the L_1 -norm minimization in the following eqn. (6.7).

$$\begin{aligned} (\hat{x}, \hat{e}') &= \underset{x, e'}{\operatorname{argmin}} \|x\|_1 + \|e'\|_1 \\ \text{s.t. } y' &= A'x + \lambda e' \end{aligned} \quad (6.7)$$

where λ is a parameter which indirectly controls the trade-off between $\|x\|_1$ and $\|e'\|_1$.

The second sparse reconstruction generates the estimated high-resolution residuals $\hat{e} \in \mathbb{R}^m$ with a dictionary $A \in \mathbb{R}^{m \times n}$ through the following eqn. (6.8).

$$\hat{e} = y - A\hat{x} \quad (6.8)$$

Note that both the dictionaries A' and A are constructed with $w' \times h'$ -dimensional lower resolution defect-free (normal) training images and $w \times h$ -dimensional high-resolution defect-free (normal) training images respectively. The downsample methods used to downsample the original data to the $w' \times h'$ -dimensional input/observed image and training images of A' are the same. Similarly, the downsample methods for generating $w \times h$ -dimensional input/observed data and training images of A are the same. In general, dictionaries A and A' can be constructed with dictionary learning [40], [463]. In the experiments of this section, the atoms of A and A' are the vectorized normalized image vectors of the $w \times h$ -dimensional and $w' \times h'$ -dimensional training images respectively as dictionary learning is not the focus of this chapter.

In the following subsection, a classifier which is proposed and used for the pixel classification (defective and normal pixel classification) for the extended SRA is elaborated.

6.5.2. Defect Detection and Localization

After the estimated high-resolution residuals $\hat{e}$ has been derived from the fast sparse reconstruction, morphological operations and a thresholding classifier are applied to remove noise and to classify each pixel in the original input/observed image into two classes; namely: defective and normal pixels.

Firstly, all the values in $\hat{e}$ are rounded to integers between 0 and 255 (grayscale pixel value). The rounded residuals $\hat{e}_r$ and its corresponding estimated rounded residual image $\hat{R}$ are used for classifying all pixels in the input/observed image to defective and normal pixels using the classifier in the following eqn. (6.9)

$$C(y_i) = \begin{cases} 1 & |\hat{e}_{r,i}| - \bar{T} > 0 \text{ \& } D_l \leq D_i(\bar{T}) \leq D_u \\ 0 & \text{otherwise,} \end{cases} \quad (6.9)$$

where

y_i is the i^{th} element of input/observe data y ,

$\hat{e}_{r,i}$ is the i^{th} element of the rounded residuals $\hat{e}_r$,

$\bar{T}$ is a threshold,

$D_i(\bar{T})$ are the dimensions (width and height) of the smallest bounding box of the connected region which pixels' rounded residuals are higher than the threshold $\bar{T}$ and contains the pixel corresponding to the y_i in the estimated rounded residual image $\hat{R}$,

D_l is the lower bound of the dimensions $D_i(\bar{T})$, and

D_u is the upper bound of the dimensions $D_i(\bar{T})$

$C(y_i)$ returns 1 (classify the pixel corresponding to y_i to defective pixel) when all the following conditions are true:

- The absolute estimated residual value $|\hat{e}_{r,i}|$ at the pixel is higher than the threshold $\bar{T}$,
- The pixel is in a connected region (estimated defective connected region) where all the absolute residual values in the region in the rounded residual image R are higher than the threshold $\bar{T}$
- The dimensions (width and height) $D_i(\bar{T})$ of the smallest bounding box of the connected region are between D_l and D_u .

Otherwise $\mathcal{C}(y_i)$ returns 0 (classify the pixel to normal pixel)

Note that the bounding box constraints are for removing the noise in the estimated rounded residual image $\hat{R}$ and depend on the inspection requirements (i.e. the tolerance of acceptable defect and unacceptable anomaly specifications such as sizes of defects). Moreover, the threshold $\bar{T}$ in the formulation eqn. (6.9) is input/observed image dependent and is designed to adapt to the illumination condition in the input/observed image. It replaces the universal threshold used in the original SRA [5] which was introduced in Section 6.4. Morphological operations (opening and closing morphological operations of the square kernel of size S_k) may be applied to the thresholded residual image before forming the connected region for noise removal.

In the classifier model in eqn. (6.9), the main challenge is the selection of a proper threshold $\bar{T}$ under the specific illumination condition of the input/observed image. Let $N_c(t)$ be the number of estimated defective connected regions after applying eqn. (6.9) with $\bar{T} = t$. Define the threshold $\bar{T}$ as in eqn. (6.10).

$$\bar{T} = \min(\{t | t = 0, 1, \dots, 255, N_c(t) \geq \eta\}) \quad (6.10)$$

where

η is a parameter which indicates the number of the estimated connected defective regions.

$\bar{T}$ is the lowest integer threshold which has at least η estimated defective connected regions. η is dataset dependent (i.e. the characteristics of noise and the possible number of defect occurrence in an input/observed image). Hence, η needs to be re-estimated whenever a new dataset which has different characteristics (i.e. the changes of target defect type, the changes of equipment such as the camera, the setup of illumination sources, the changes of inspected product, etc.) is used.

In the following subsection, a way of finding the best parameters λ in eqn. (6.7) and η in eqn. (6.10) is presented.

6.5.3. Finding Best λ and η

In eqns. (6.7) and (6.10), there are two dataset dependent parameters, λ and η which have to be learned. These two parameters depend on the characteristics of the defect type, the background, the image acquisition devices, the inspected surface material, etc. In Section 6.4.4, the parameter analysis of original SRA shows that the behaviour of the weighted accuracy defined in eqn. (6.5) changing with any one of the parameters is convex globally and is non-convex locally. Hence, a specifically supervised parameter learning algorithm to address this special non-convex problem is proposed. The parameter learnings are performed in a two-step greedy search described by **Algorithm E.2** (in Appendix E). **Algorithms E.3** and **E.4** (in Appendix E) learn λ and η respectively. The two Algorithms (Step 4 and 5 (Parameter Optimizations E.1) in **Algorithm E.2**) are iterated until either the validation accuracy converges (depends on the stopping criterion of accuracy, S_1) or the maximum number of iterations S_2 has been reached.

Note that the weighted defect detection accuracy in the following eqn. (6.11) is used instead of the balanced accuracy in eqn. (6.5). The weighted accuracy $Acc_w(\beta)$ is the generalization of balanced accuracy [462].

$$\begin{aligned} Acc_w(\beta) &= \beta(sensitivity) + (1 - \beta)(specificity) \\ &= \beta \frac{TP}{P} + (1 - \beta) \frac{TN}{N} \end{aligned} \quad (6.11)$$

where

- β is a trade-off coefficient,
- TP is the number of true positives,
- TN is the number of true negatives,
- P is the number of positives, and
- N is the number of negatives.

Acc_w is the sum of the weighted sensitivity (recall) [464] and specificity [464] measures. Acc_w alleviates the imbalanced dataset challenges in many real-world problems especially for anomaly detection task which has infrequent anomalies.

The details of the parameter learning algorithms can be found in Section E.2 in Appendix E.

6.6. Illumination Reconstruction for Lambertian Surfaces

This section shows the illumination condition reflected by surfaces presented in an image can be expressed by a linear combination of illumination conditions in other images which are acquired under the same viewpoint. The surfaces appear in images are assumed to be Lambertian surfaces. A single directed lighting source is assumed to be used.

Reflection of light from a Lambertian surface can be expressed in the following formula [465] eqn. (6.12):

$$p_{i,j} = \tilde{\rho} n_{i,j} \cdot l \quad (6.12)$$

where

- $p_{i,j}$ is the pixel intensity of a captured image at location coordinate i and j ,
- l is the lighting direction,
- $\tilde{\rho}$ is the diffuse albedo, and
- $n_{i,j}$ is the surface normal at i and j .

Hence, pixels $p_{i,j}$ in an image I_k can be expressed with the following formula: see eqn. (6.13)

$$p_{i,j|k} = \tilde{\rho} n_{i,j|k} \cdot l_k \quad (6.13)$$

where

- l_k is the lighting direction under which I_k was acquired, and
- $n_{i,j|k}$ is the surface normal at i and j under the viewpoint V_i .

Eqn. (6.13) can be expanded to the following eqn. (6.14)

$$p_{i,j|k} = \tilde{\rho} (\cos\theta_{a,i,j|k} \cos\alpha_{a|k} + \cos\theta_{b,i,j|k} \cos\alpha_{b|k} + \cos\theta_{c,i,j|k} \cos\alpha_{c|k}) \quad (6.14)$$

where $n_{i,j|k}$ and l_k can be expressed as in eqns (6.15) and (6.16).

$$n_{i,j|k} = (\cos\theta_{a,i,j|k}, \cos\theta_{b,i,j|k}, \cos\theta_{c,i,j|k})^T \quad (6.15)$$

$$l_k = \tilde{\rho}(\cos\alpha_{a|k}, \cos\alpha_{b|k}, \cos\alpha_{c|k})^T \quad (6.16)$$

Considering that all pixels, $p_{i,j|1}$ in an image, I_1 , can be expressed as a linear combinations of the pixels, $p_{i,j|2}, \dots, p_{i,j|N_v}$, which are at the same location in other images, $I_1, \dots, I_{N_v}$ that were acquired under the same viewpoint as the following eqn. (6.17)

$$p_{i,j|1} = \sum_{k=2}^{N_v} c_k p_{i,j|k} \quad (6.17)$$

where

N_v is the number of images for which the pixels are used to express the pixel $p_{i,j|1}$ in image I_1 , and

c_i are variables.

The following eqns. (6.18)-(6.20) can be formed:

$$\cos\alpha_{a|1} = \sum_{k=2}^{N_v} c_k \cos\alpha_{a|k} \quad (6.18)$$

$$\cos\alpha_{b|1} = \sum_{k=2}^{N_v} c_k \cos\alpha_{b|k} \quad (6.19)$$

$$\cos\alpha_{c|1} = \sum_{k=2}^{N_v} c_k \cos\alpha_{c|k} \quad (6.20)$$

According to the Rouché-Capelli theorem [466], the rank of coefficient (coefficients of variables c_k) matrix must be larger than or equal to the rank of augmented matrix in order to have one or infinite solution of variables c_k . The maximum rank of augmented matrix of eqns. (6.18)-(6.20) is 3. Hence, it is guaranteed to have at least one solution if 3 images which coefficient matrix's rank is equal to 3 are used to reconstruct I_1 .

The following formula eqns (6.21) and (6.22) are extended from eqn. (6.17) by incorporating residual terms, $e_{i,j|1}$ and $e_{i,j|k}$, which are potentially caused by noises, corruptions, occlusions, shadows, defects, etc.

$$p_{i,j|1} - e_{i,j|1} = \sum_{k=2}^{N_v} c_k (p_{i,j|k} - e_{i,j|k}) \quad (6.21)$$

$$p_{i,j|1} = \sum_{k=2}^{N_v} c_k (p_{i,j|k} - e_{i,j|k}) + e_{i,j|1} \quad (6.22)$$

The L_1 -norm minimization in the following eqn. (3.16) can learn the best variable solution which have the lowest reconstruction residuals.

$$\begin{aligned} & \min_{c,e} \|e\|_1 \\ \text{s.t. } & p_{i,j|1} = \sum_{k=2}^{N_v} c_k p_{i,j|k} - \sum_{k=2}^{N_v} c_k e_{i,j|k} + e_{i,j|1} \end{aligned} \quad (6.23)$$

where c and e are defined in eqns (6.24) and (6.25).

$$\check{c} = (c_2, \dots, c_k)^T \quad (6.24)$$

$$e = - \sum_{k=2}^{N_v} c_k e_{i,j|k} + e_{i,j|1} \quad (6.25)$$

Given that a dictionary A contains atoms constructed by training images which were acquired under various viewpoints, the L_1 -norm minimization in eqns (6.2) and (6.7) are used to learn the sparse coefficient vector x (which is similar to the role of $\check{c}$ defined in eqn. (6.24)) and the error vector e .

Although the definition of the error vector in eqn. (6.25) implies that the more images added in to reconstruct the image I_1 , the lesser the resultant of the objective function $\|e\|_1$. Hence, the more images used in the reconstruction can diminish more reconstruction error. However, **Theorem 1** and the experimental results in Chapter 4 shows the following two scenarios:

- the more images used (more supports of the original/ground truth sparse coefficient vector x_o have) to reconstruct the input/observe image, and
- the more images are stored in the dictionary A (more atoms in A).

will lower the successful recovery probability bound. Moreover, it will also increase the computation time and memory demand of the minimization due to the increase of complexity of the problem. Hence, it becomes a tradeoff in increasing the images used.

Note that the proposed method is for solving the illumination reconstruction problem under the assumption that any illumination condition can be linearly constructed by a combination of the other illumination conditions such as the illumination reflected from Lambertian surface. If the surface is non-Lambertian or the illumination condition reflected from the surface cannot be linearly constructed by the other illumination conditions, the feasibility of our proposed method will need to be checked with a studies of the approximation errors of our proposed method. Moreover, a data transformation function may need to be applied to transform the illumination conditions into the forms which are linearly dependent [467]. Changing of the light acquisition method to other photometric stereo methods to collect linear dependent lighting information can be also applied [468]–[470]. Otherwise, the other non-linear illumination reconstruction techniques can be considered [471].

6.7. Computation Time and Memory Demand Analysis

This section analyses the computation time (Section 6.7.1) and memory demand (Section 6.7.2). Downsampled input/observed image is applied in estimating the sparse coefficient vector x of the input/observed image using the L_1 -norm minimizations in eqns (6.2) and (6.7). It shows that, the smaller the downsampled resolution, the lesser the computation bound (computation time in the worst case) as well as the memory demand. The L_1 -norm minimizations in eqns (6.2) and (6.7) can be reformulated to linear programming problems. Such linear programming algorithms are much faster than the non-smooth convex optimization techniques. The computation time and memory demand analyses are done based on the various linear programming algorithms.

6.7.1. Computation Time Analysis

Cohen et al. [145] showed that the fastest algorithm solving linear programming in time shown in eqn. (6.26).

$$O^* \left(\left(\tilde{D}^\omega + \tilde{D}^{2.5 - \frac{\tilde{\theta}}{2}} + \tilde{D}^{2 + \frac{1}{6}} \right) \log \frac{\tilde{D}}{\tilde{\epsilon}} \right) \quad (6.26)$$

where

- $\tilde{D}$ is the number of variables
- ω is the exponent of matrix multiplication
- $\tilde{\vartheta}$ is the dual exponent of matrix multiplication
- $\tilde{\epsilon}$ is the relative accuracy

The numbers of variables $\tilde{D}$ after converting L_1 -norm minimizations eqns. (6.2) and (6.7) into linear programming problems are $2(n + m)$ and $2(n + m')$ respectively. n is the number of the atoms (columns) in dictionary A (see both eqns. (6.2) and (6.7)). $m = w \times h$ is the product of width and height of the output image and m' ($m' < m$) is the product of the width and height of the low resolution input/observed image used in (6.7).

When the original SRA is applied, the image resolutions of output image and the images used in the L_1 -norm minimization in eqn. (6.2) are the same. Hence, the worst-case complexity (see eqn. (6.26)) of the fastest linear programming algorithm increases as the m increase ($m = w \times h$ is the product of width and height of the output image). When extended SRA is used, the image dimension used for solving L_1 -norm minimization in eqn. (6.7) is independent on the output image resolution. It depends on m' and, hence the extended SRA is more practical and can have much less computational bound for the defect detection and localization task to obtain very high resolution output images ($m' \ll m$).

6.7.2. Memory Demand Analysis

When L_1 -norm minimizations in eqns. (6.2) and (6.7) are arranged that hence into linear programming problems, there are two $p \times p$ -dimensional identity matrices to be stored in the memory before solving the L_1 -norm minimizations. Note again that $p = m$ and $p = m'$ in eqns (6.2) and (6.7) respectively. $m = w \times h$ is the product of width and height of the output image and m' ($m' < m$) is the product of the width and height of the low resolution input/observed image used in (6.7). Hence, the memory demand of storing the two identical matrices quadratically increases as p increases. The memory demand for solving the optimization problem in eqn. (6.2) to have a high resolution output image is very high ($p = m$). While the value p in optimization problem eqn. (6.7) is independent of the high

resolution output image ($p = m'$) and, hence the memory demand can be low. As the result, the extended SRA is suitable for high resolution defect detection and localization ($m' \ll m$).

6.8. Experiments

In this section, the experimental results for both original *SRA* (in Section 6.8.1) and extended *SRA* (Section 6.8.2) are presented. In addition, the performance comparisons of the both original and extended *SRA*s (Section 6.8.3) is also presented.

6.8.1. Experimental Results for Original SRA

In this subsection, the experimental results of multiple performance comparisons for the proposed original *SRA* are presented. Section 6.8.1 A presents the performance comparisons of the proposed original *SRA* with different dictionaries. Section 6.8.1 B presents the effect of downsampled input/observed image resolution on the performance of original *SRA*. Section 6.8.1 presents the performance comparison of the proposed original *SRA* and different state-of-the-art saliency detection algorithms.

A. Performance Comparison with Different Dictionaries

Metallic Object Dataset

The dataset used in the experiments is a subset of a large number of images collected from a visual inspection process of quality control in a metallic object manufacturing process⁷. The anomalies which are interest are defects on a metallic surface. The dataset consists of 641 images with a resolution of 2448×2050 pixels. 368 images do not contain any defect and 273 images contain one of the three defects: “melt”, “plus metal”, and “shadowing”. Each image is taken from one of the six viewpoints of the object of interest. Some defects have complicated illumination dependent characteristics. For example, “plus metal” can only be visible under dark condition. These defects are labeled as the anomalies in the images for the experiments. Each image with anomaly has a ground truth image for

⁷ Due to confidentiality, the description is limited.

performance evaluation purposes. In the ground truth image, a pixel can be either a defective (anomalous) pixel or a normal pixel.

Experimental Design

Due to high computational cost of high resolutions, images were downsampled. Bicubic interpolation is used to downsample the images. The pixel values which are larger than zero in the downsampled ground truth images are labeled as defective pixels. All 368 images containing defect-free objects are used to construct the dictionaries. For comparison purposes, three dictionaries are used in the experiment, namely: (i) normalized images, (ii) dictionary constructed using the online robust dictionary learning (*ORDL*) [463], and (iii) dictionary constructed using scale adaptive dictionary learning (*SADL*) [40]. For performance evaluation, the balanced accuracy [462] in the eqn. (6.5) is used. Note that the default values for parameter and variable initialization are described in **Algorithm E.1** in Appendix E.

Performance of Different Dictionaries at Fixed λ

Figure 6.11 shows the Receiver Operating Characteristic (ROC) curves for the three dictionaries stated in previous subsection and three λ values: 0.001, 1, and 10. The image resolution is 30×30 pixels after downsampling. To learn a dictionary using *ORDL*, the number of images in a batch and the number of atoms were set to 200 and 368 (the number of images without defect), respectively. To learn a dictionary using *SADL*, the maximum number of atoms in the dictionary is set to 500. Note that the dictionary learnt using *SADL* has only 14 atoms. The pixels of the 273 images with anomaly are used to construct the ROC curves. The true and false positive rates in the ROC curves were computed by varying the thresholds from the lowest to the highest absolute residual value.

Figure 6.11 (bottom right) shows a comparison of the best anomaly detection performance using the three dictionaries for sparse reconstruction. One interesting observation is that the best performance using the learnt dictionary of only 14 atoms from *SADL* is comparable to using a dictionary with 368 atoms. The online approach of learning dictionary may result

in lost in critical information. As a result, it does not perform as well as the other two dictionaries.

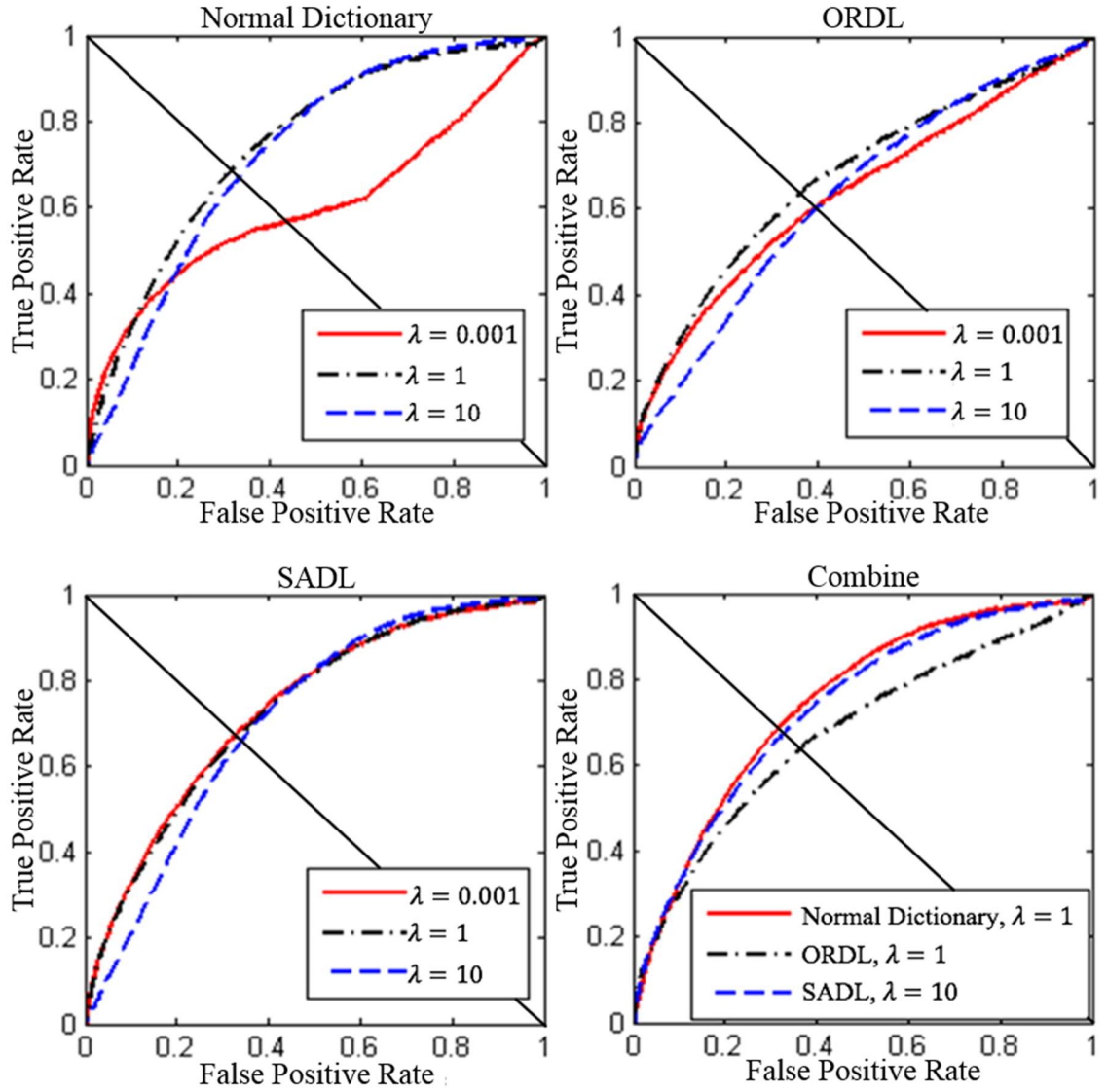


Figure 6.11 ROC curves of normal, *ORDL* and *SADL* dictionaries with different λ

Performance of Different Dictionaries using Proposed Parameter Learning Method

The images containing defects are split into five sets to perform a five-fold cross-validation for the proposed parameter learning method (**Algorithm E.1**).

Table 6.1 shows the average balanced accuracy Acc_w when the near-optimal λ and learnt classification threshold $\bar{T}$ from

Dictionary	Normalized images	ORDL	SADL
Average Acc_b	0.69	0.63	0.63

Algorithm

E.1 are used to identify the defective pixels in the test images with image resolution of 15×15 pixels. It shows that **Algorithm E.1** performs the best when all the images containing defect-free objects are used in the dictionary.

Table 6.1 Comparison of (balanced) accuracies with different dictionaries

Dictionary	Normalized images	ORDL	SADL
Average Acc_b	0.69	0.63	0.63

A. Effect of Downsampled Resolution on Performance

Table 6.2 shows the average Acc_b (in eqn. (6.5)) when parameter learning (**Algorithm E.1**) with dictionary containing images of defect-free object at different image resolutions are used. It should not be surprising to observe that higher resolution images result in better Acc_b . However, higher resolution images mean higher computational cost since some iterative steps in **Algorithm E.1** requires the handling of the images in the dictionary. Note that, the memory demand for the L_1 -norm minimization eqn. (6.2) of original *SRA* increase quadratically as the image resolution used in the optimization problem increases. The memory demand analysis is shown in Section 6.7.2.

Table 6.2 Comparison of (balanced) accuracies with different resolutions

Resolution (pixels)	5×5	10×10	15×15	20×20
Average Acc_b	0.56	0.65	0.69	0.70

B. Performance Comparisons with Other Saliency Detection Algorithms

Metallic Object Dataset

Another subset of the large number of images collected from the same visual inspection process for quality control in a metallic object manufacturing process⁸ was the experimental dataset in the experiments. Anomalies are the defects on metallic surfaces.

⁸ Due to confidentiality, the description is limited.

The dataset consists of 891 images with original resolution of 2050×2448 pixels and 500 images are defect-free images. The non-defect free images contain a defect of one of defect types: “Melt”, “Plus Metal”, and “Scuff”. Each image is taken from one of the seven viewpoints of the object of interest. The other image characteristics are described in the Metallic Object Dataset in A. Performance Comparison with Different Dictionaries.

Experimental Design

Same as in Section 6.8.1, bicubic interpolation is used to downsample the images and the pixel values which are larger than zero in the downsampled ground truth images are labeled as defective pixels. The proposed original *SRA* (*SR* (in following figures and table)) algorithm and two state-of-the-art saliency detection algorithms, graph-based manifold ranking (*MR*) [472] and spatial filtering (*SF*) [473] are used for performance comparisons. For performance evaluation, the balanced accuracy [462] in formula eqn. (6.5) is used. Note that the default values for parameter and variable initialization are declared in **Algorithm E.1**.

In each experimental trial, images of two defect instances using same defect type and their corresponding defect-free images that have the same viewpoints with images of any one of these two instances were used. Images of one instance is the training set and the other one is the test set. All defect-free images of the instance pair were used in constructing the dictionary. All performance measures are either the average results of defect detection on both instance dataset or the individual result of defect detection of each instance dataset. All instances are named as: “Melt A” (58 [images contain defects], 69 [defect-free images]), “Melt B” (58, 161), “Melt C” (58, 69), “Plus Metal A” (53, 69), “Plus Metal B” (52, 23), “Scuff A” (53, 46) and “Scuff B” (57, 63). All instance pairs are organized as follows: “Melt 1” (“Melt A” and “Melt B”), “Melt 2” (“Melt A” and “Melt C”), “Melt 3” (“Melt B” and “Melt C”), “Plus Metal” (“Plus Metal A” and “Plus Metal B”) and “Scuff” (“Scuff A” and “Scuff B”).

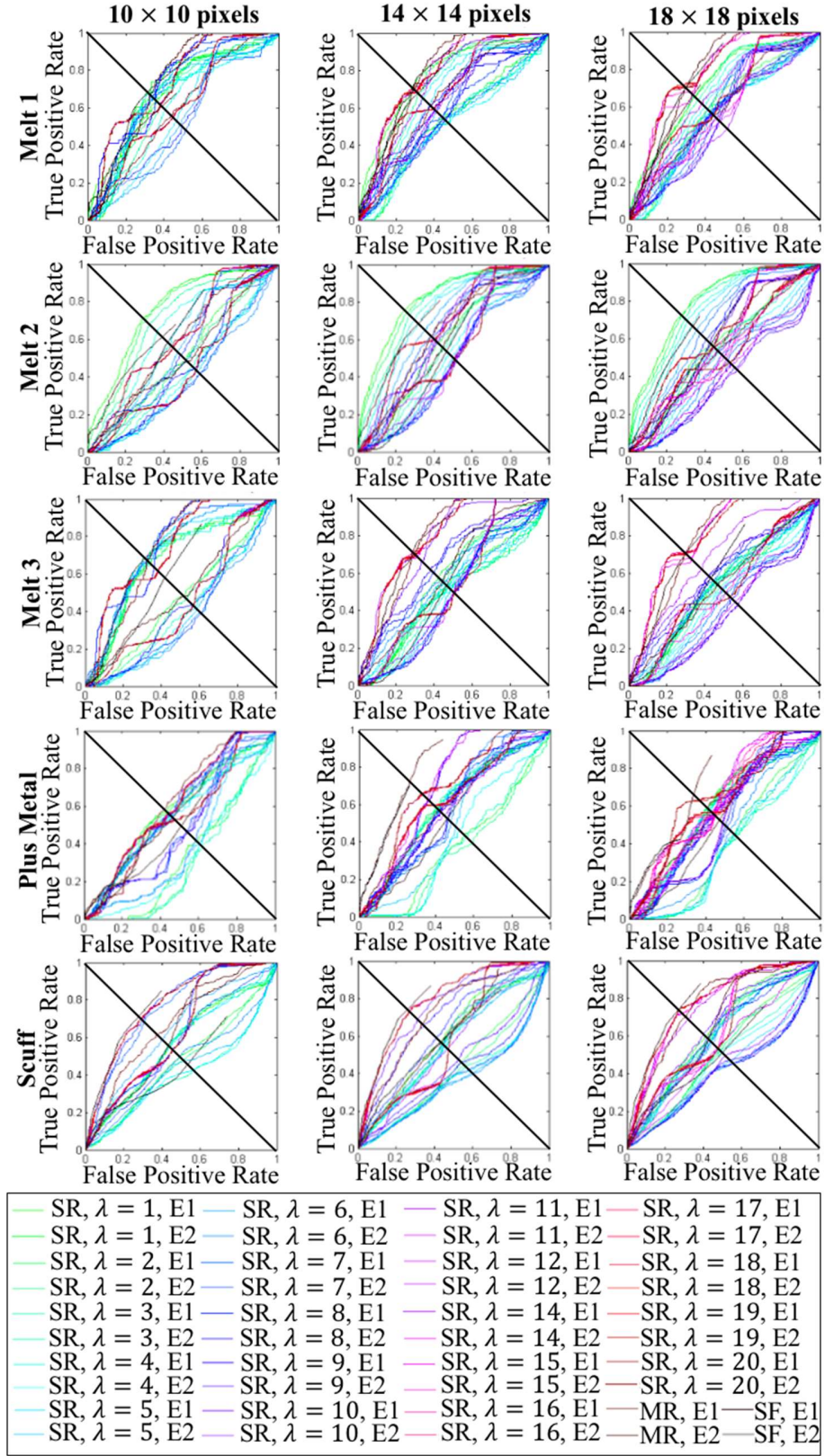
A simple thresholding algorithm is applied to extract a mask of the background and the shadow for each input image. The threshold for background and shadow extraction is the

pixel value which is at the second trough in the gray pixel value histogram. All input images were normalized using linear stretching of all pixel values so that the median values of foreground pixels were transformed to 128. The linear stretching was used for reducing the variation of images due to the illumination lightness.

Performance Comparisons

ROC curve was used to analyse the proposed model in eqn. (6.2) and the other two saliency detection models. For each instance pair, the images are downsampled to 3 resolutions, 10×10 , 14×14 and 18×18 pixels. There are 5 experimental trials for each instance pair with all downsampled resolutions. All ROC curves of the proposed original *SRA* (*SR*) with various λ , *MR* and *SF* on dataset of each instance pair and various resolutions are shown in Figure 6.12. In the figure, “E” stands for “Experiment” and “E1” represents the experiment using images which contain defects of first instance of the instance pair as the input dataset.

Results show that the proposed model outperformed *MR* and is competitive to the *SF*. At low resolutions, the proposed approach’s ROC curves are better than the ROC curves of both saliency detection algorithms. While the proposed approach has similar results with *SF* which outperformed the proposed approach at high resolutions when proper λ is chosen. In this case, the proposed optimization algorithm in Section 6.4.3 can be used to learn the optimal λ and the optimal classification threshold $\bar{T}$. Moreover, the proposed model significantly outperformed the two algorithms in experiments of “Melt 2”.

Figure 6.12 ROC curves of different models with different λ

Effect of Downsampled Resolution on Performance

Table 6.3 shows that the mean balanced accuracy of each experimental trial with the proposed approach using **Algorithm E.1** and two saliency detection algorithms, *MR* and *SF*, at resolutions 10×10 and 20×20 pixels. The defect detection classifier in Section 6.4.2 has been applied on the saliency maps which were generated by the two saliency detection algorithms. The saliency maps contain pixel values ranging from 0 to 255. The defect detection classifier classified all pixels in the saliency maps to the 2 classes (defective pixel and normal pixel). Hence, the mean balanced accuracies (mean of Acc_b in eqn. (6.5)) of defect detection on the saliency maps can be computed for the performance comparisons. It shows that the performance of the proposed approach is competitive with the two saliency algorithms especially at resolution 10×10 pixels and in experiments of “Melt 1” and “Melt 2”.

Table 6.3 Comparison of (balanced) accuracies with different resolutions and algorithms

	SR		MR		SF	
	10×10	20×20	10×10	20×20	10×10	20×20
Melt 1	0.6867	0.6858	0.5653	0.6035	0.6393	0.6857
Melt 2	0.6260	0.6780	0.5202	0.5118	0.6184	0.6495
Melt 3	0.6088	0.5838	0.5547	0.6035	0.6488	0.6509
Plus Metal	0.5694	0.5990	0.5472	0.6359	0.5086	0.6004
Scuff	0.5390	0.5657	0.5500	0.5889	0.6121	0.6459

6.8.2. Experimental Results for Extended SRA

In this subsection, the experimental results of multiple performance comparisons for the proposed extended *SRA* are presented. Section 6.8.2 A and Section 6.8.2 B present the dataset and the experimental design used in the following experimental trials. Section 6.8.2 C, D and E present the effects of the following factors on the performance of the proposed extended *SRA*:

- the low input/observed input image resolution (Section 6.8.2 C),
- the coefficient in weighted accuracy β in eqn. (6.11) (the weighted accuracy is the objective function for finding the best parameters of the proposed extended *SRA* (Section 6.8.2 D), and
- the morphological operation kernel size S_k (Section 6.8.2 E).

Section 6.8.2 F presents the performance comparisons of the proposed extended *SRA* [6], and two following state-of-the-art saliency detection algorithms:

- graph-based manifold ranking (*MR*) [472], and
- spatial filtering (*SF*) [473]

A. Metallic Object Dataset

The dataset which is used in the experiments for extended *SRA* is a subset of a large number of images collected from a visual inspection process for quality control in a metallic object manufacturing process⁹. Anomalies are the defects on metallic surfaces. The dataset consists of 891 images with an original resolution of 2050×2448 pixels. 500 images are defect-free images and the other images contain a defect of one of three defect types; namely: “Melt”, “Plus Metal”, and “Scuff”. Each image is taken from one of the seven viewpoints of the object of interest. Some defects have complicated illumination dependent characteristics. For example, “plus metal” can only be visible in dark condition. These defects are labeled as the anomalies in the images for the experiments. Each image with anomaly has a ground truth image for performance evaluation purposes. In the ground truth image, a pixel can be either a defect (anomalous) pixel or a normal pixel.

B. Experimental Design

Figure 6.13 shows the overview of the experimental design. Each of the 2 instances of the same defect type are grouped together for an experimental trial. All images of an instance are taken under the same viewpoint and different illumination conditions. Defect-free (normal) Images of the 2 instances are used to construct the dictionary for the sparse reconstructions. The images which contain defect of each instance take turn to be the training and test sets for the parameter learning. Note that, the experimental setup is based on the following assumptions:

- Viewpoint of each instance are critical. Hence, defect-free images of the input/observed image need to be in the dictionary.

⁹ Due to confidentiality, the description is limited.

- The parameters for defect detection and localization of the same defect type should be similar and hence it can be learnt and can be reused for detecting the same defect type in images of other instances.

In performance evaluation, the pixel values which are larger than zero in the high resolution ground truth images are labeled as ground truth defective pixels and the others pixels are labeled as ground truth normal pixels. Table 6.4 and Table 6.5 show the details of the 7 instances and the 5 instance pairs used in the experiments.

A simple thresholding algorithm is applied to extract a mask of the background and shadow for each input image. The threshold for background and shadow extraction is the pixel value which is at the second trough in the gray pixel value histogram. All input images are normalized with linear stretching of all pixel values so that the median values of foreground pixels are transformed to 128. The linear stretching is used to reduce the variation of images due to the illumination lightness.

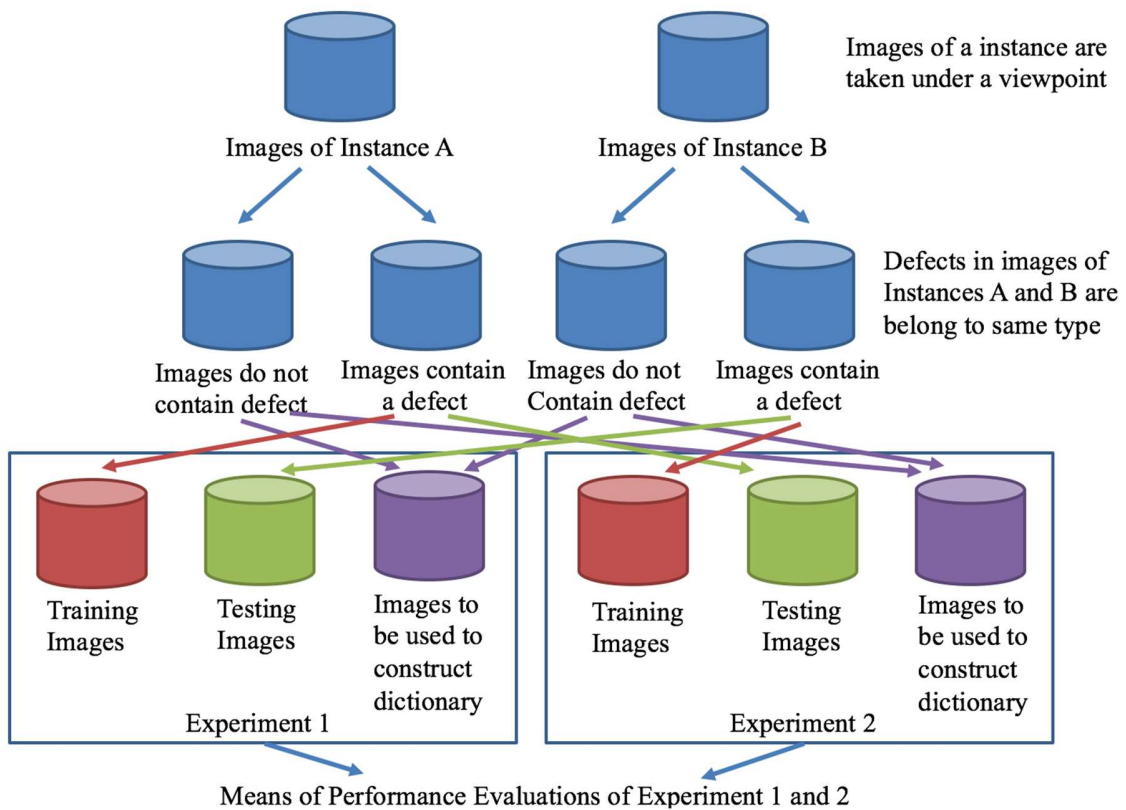


Figure 6.13 The overview of the experimental design

Table 6.4 Instance details

Instance name	Melt A	Melt B	Melt C	Plus Metal A	Plus Metal B	Scuff A	Scuff B
Number of images contain defects	58	58	58	53	52	53	57
Number of defect-free images	69	161	69	69	23	46	63

Table 6.5 Instance pair details

Instance pair name	Melt 1	Melt 2	Melt 3	Plus Metal	Scuff
First instance	Melt A	Melt A	Melt B	Plus Metal A	Scuff A
Second instance	Melt B	Melt C	Melt C	Plus Metal B	Scuff B

The default values for the parameter and variable initialization are declared in all algorithms presented in Section E.2 in Appendix E. Figure 6.14 shows four examples of defect detection and localization results based on the proposed extended *SRA*. The images contain “Melt” (Left), “Plus Metal”, “Melt” and “Scuff” defects. Blue and red boxes are the smallest bounding boxes which bound individual connected true positives and false positives respectively. Four performance measures, namely: weighted accuracy, balanced precision, precision [464] and recall [464]) are used. The weighted accuracy in eqn. (6.11) is a generalization of the balanced accuracy in eqn. (6.5) [462].

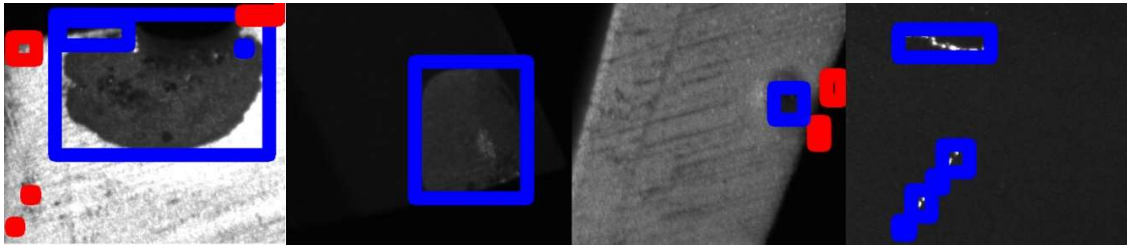


Figure 6.14 Examples of defect detection and localization results

Balanced Precision is defined in eqn. (6.27) and is a weighted version of the precision.

$$\begin{aligned}
 \text{Balanced Precision} &= \frac{\frac{TP}{P}}{\frac{TP}{P} + \frac{FP}{N}} \\
 &= \frac{N \cdot TP}{N \cdot TP + P \cdot FP}
 \end{aligned} \tag{6.27}$$

where

- TP is the number of true positives,
- TN is the number of true negatives,
- P is the number of positives, and
- N is the number of negatives.

It is computed by assigning a weight ($\frac{1}{P}$ for true positive and $\frac{1}{N}$ for false positive) to each pixel. It is a newly introduced performance measure to cope with the imbalanced data problem. The weight assignment is to alleviate the bias due to the imbalanced numbers of defective and normal pixels.

A. Effects of Image Resolution in the First Sparse Reconstruction

In the experiments of this subsection, one of 5×5 (width $\times$ height), 10×10 and 25×25 is the low image resolution used in the first sparse reconstruction (in eqn. (6.7)) and 25×25 pixels is the high image resolution are used in the second reconstruction (in eqn. (6.8)) to generate the high resolution output. All downsampled resolutions are achieved by applying bicubic interpolation on the images at original resolution.

Figure 6.15 shows the following performance measures of the proposed extended *SRA*:

- the individual balanced accuracies (weighted accuracy $Acc_w(\beta)$ in eqn. (6.11) with $\beta = 0.5$) of all experimental trials,
- the means of balanced accuracies of experiment trials under the same instance pair name (refer to Table 6.5), and
- the overall means of all means of balanced accuracies

The results of the second bullet are the mean results when each instance under the instance pair name (refers to Table 6.5) takes turn to be the test and training datasets. X-axis represents the low image resolution (width $\times$ height) used in the first sparse reconstruction (in eqn. (6.7)). No morphological operation is used in all experimental trials. In the top left figure, the green line with instance name Melt A are the balanced accuracies when the images of that Melt A is used as test images and the images of the Melt B is used for training. The defect-free images of both Melt A and Melt B are used to construct the dictionaries. The line with instance pair name Melt 1 (refers to Table 6.5) is the mean line

of the other two lines in Figure 6.15. Same explanation of legends applies to the other 4 figures. The right bottom figure shows the mean of all means shown in the other 5 figures at various low image resolutions (x-axis).

The figure shows that all results are better than random guess (0.5). Moreover, most results do not change much with the increase in the low image resolution. It implies that the proposed new proposed approach is robust in sparse coefficient vector estimation with various low image resolutions in the first sparse reconstruction.

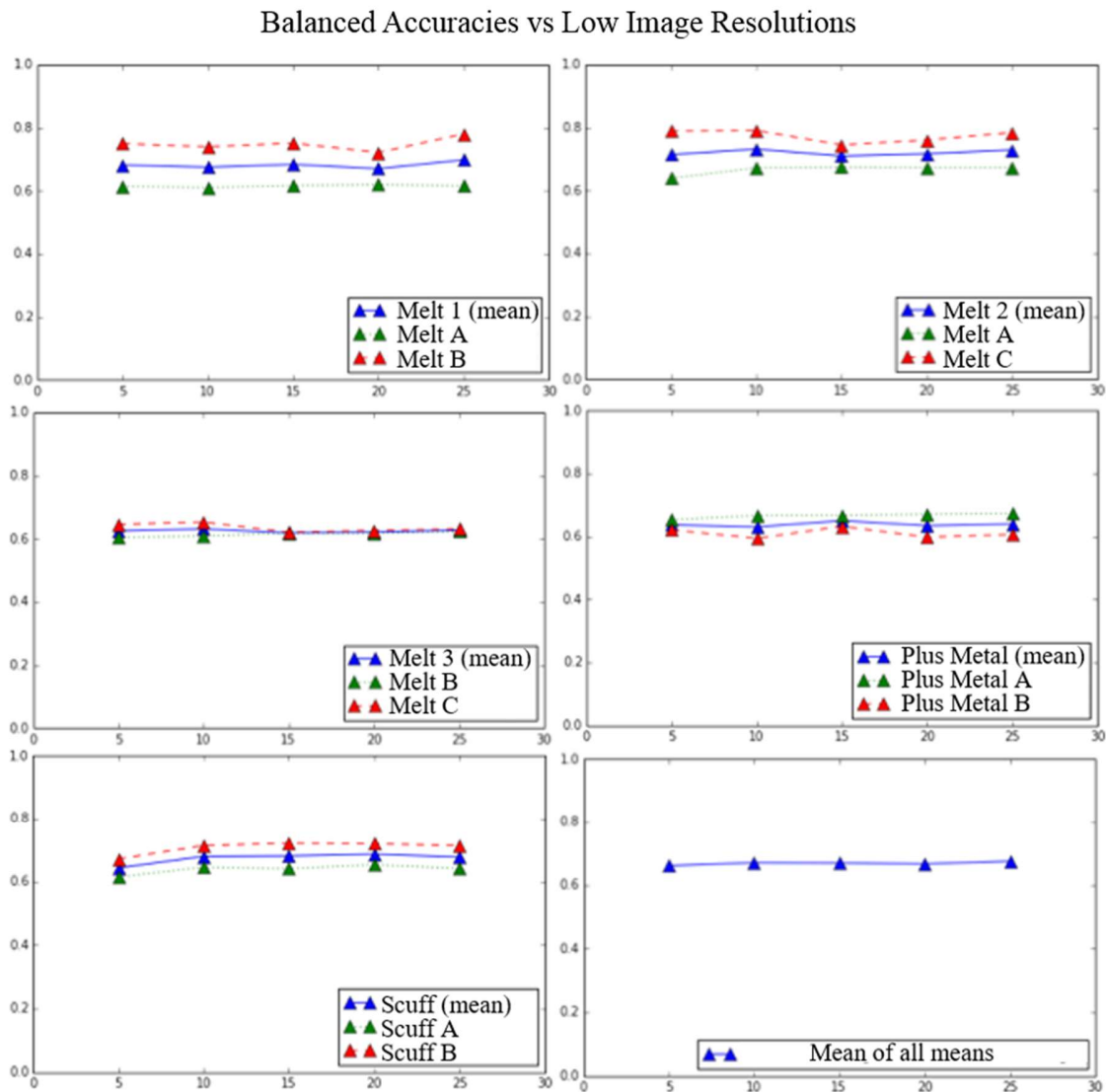


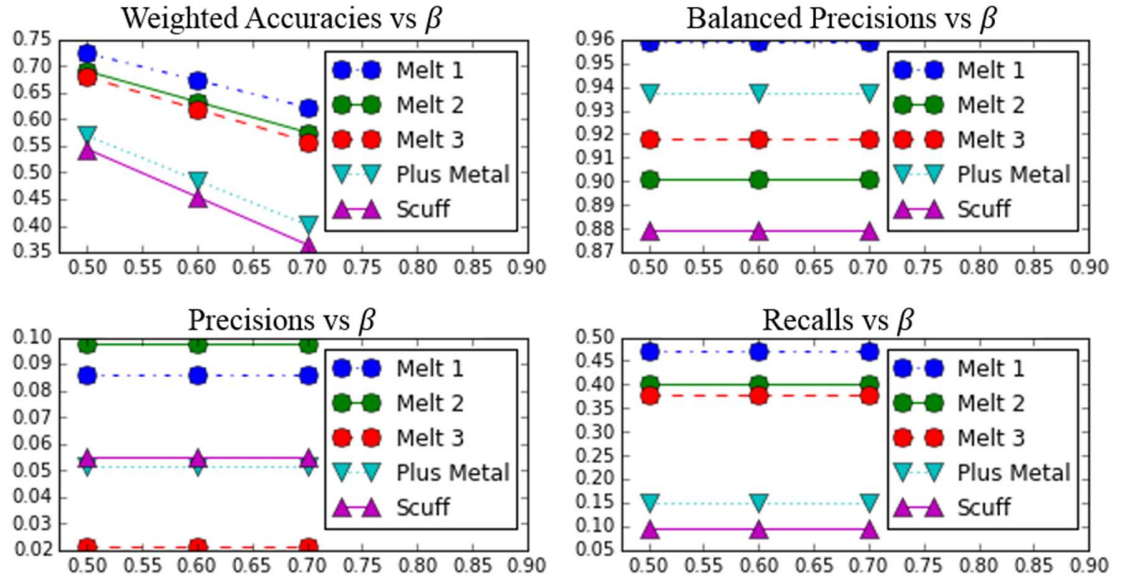
Figure 6.15 Effects of low image resolution used in the first sparse reconstruction

A. Effects of β

In the experiments in this subsection, the effects of β in the weighted accuracy $Acc_w(\beta)$ in eqn. (6.11) are analysed. The low image resolution used in the first sparse reconstruction (in eqn. (6.7)) and high resolution for the result outputs are the same with the experiments in previous subsection. The results are the mean results when each instance under the instance pair name (refers to Table 6.5) takes turn to be the test and training datasets.

Figure 6.16 shows the effects of different trade-off coefficients, β (0.5, 0.6 and 0.7), used in the algorithms for finding the best parameters λ and η . The performance measures (weighted accuracies, balanced precisions, precisions and recalls) of the proposed extended *SRA* with the learnt parameters in all experiment trials are shown. The kernel size S_k used for the morphological operations is fixed at 2.

Figure 6.16 shows that the weighted accuracies drop and the other performance measures do not change with the increase in β . It implies that the defect detection results are the same for each tested β . The weighted accuracy drop implies that the specificity is higher than sensitivity. Hence, the classifier is better in classifying normal pixels. It is reasonable as the characteristics of light reflection from the surfaces at normal pixels are less complicated. It is due to the simple characteristics of defect-free surfaces in term of texture, smoothness, size and geometrical shape compare to the defective surfaces. Defect-free surfaces have regular texture, smooth surface and same size and same geometrical shape under the same viewpoint. Moreover, the light reflections from some defective surfaces and normal surfaces are the same under certain illumination conditions. It makes the pixels at defective surfaces harder to be classified correctly. All balanced precisions are high (> 0.87). It implies that most of the estimated defective pixels are correct with the assumption that the dataset is balanced.

Figure 6.16 Effects of trade-off coefficient β

B. Effects of S_k

In this subsection, the effects of morphological operator square kernel size S_k are analysed. Similar with the previous two subsections, one of 5×5 (width $\times$ height), 10×10 and 25×25 pixels are used as the low image resolution in the first sparse reconstruction (in eqn. (6.7)) and the output's high image resolution is 25×25 pixels. The results are the mean results when each instance under the instance pair name (refers to Table 6.5) takes turn to be the test and training datasets.

Figure 6.17 shows the effects of the kernel size S_k (2, 3 and 4) of the morphological operators on the performance measures (weighted accuracies, balanced precisions, precisions and recalls) of the proposed extended *SRA*. Weighted accuracies $Acc_w(\beta)$ with $\beta = 0.5$ (a.k.a. balanced accuracies) is used.

Figure 6.17 shows that most weighted accuracies except the weighted accuracies $Acc_w(\beta)$ of “Melt 1” and “Plus Metal” increase with the increase in S_k from 2 to 4. It implies that the overall weighted accuracy improves in these cases. It also shows similar effects in the recalls of the same cases. It implies that true positive increases in these cases. However, most balanced precisions and precisions drop (except balanced precision of “Melt 3” and

precisions of “Melt 2” and “Melt 3”) with the increase of S_k . It implies that there is much more increase in false positives rather than the increase in true positives with the increase of S_k . Hence, if a system which can detect extremely low false positives is needed, new evaluation metric for the parameter optimization’s objective function is needed to replace the weighted accuracy.

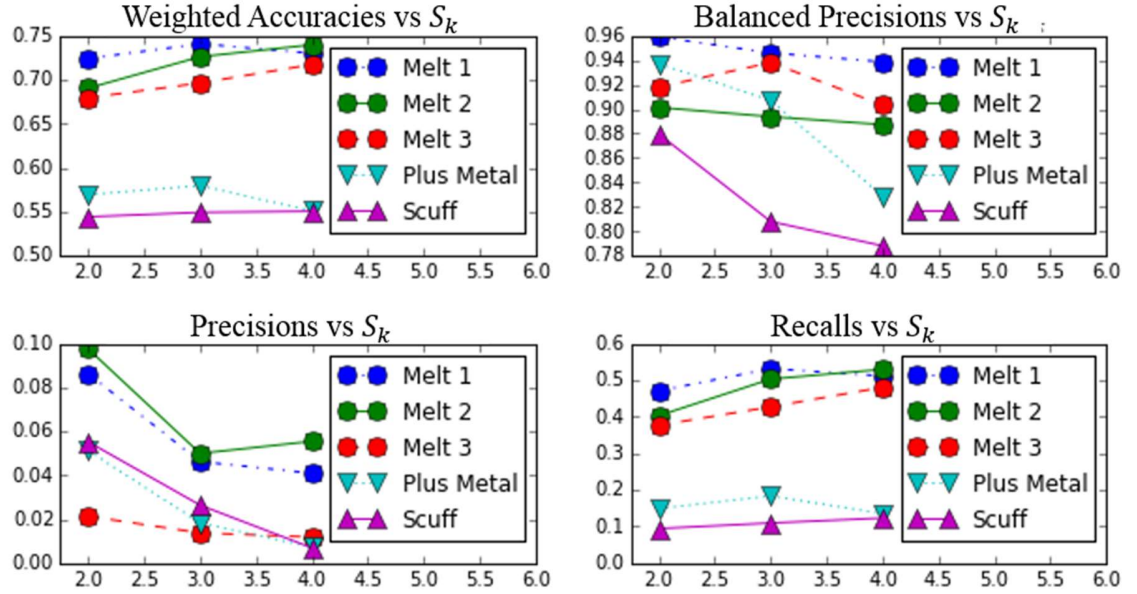


Figure 6.17 Effects of kernel size S_k

C. Performance Comparisons of Extended SRA, MR and SF

In this subsection, performance comparisons of the proposed extended *SRA* and two state-of-the-art saliency detection algorithms, graph-based manifold ranking (*MR*) and spatial filtering (*SF*) are presented. The experimental settings are as follows:

- **Extended *SRA*:** The low image resolution used in the first sparse reconstruction is 5×5 (width $\times$ height) pixels. The morphological operation kernel size S_k is set to 3.
- ***MR* and *SF*:** The resultant images which are generated with *MR* and *SF* algorithms published in [111] and [112] are used as residual images. The proposed defect detection and localization algorithm which is described in Section 6.5.2 is used for detecting anomaly in the resultant images.

The output resolution of all algorithms is 2050×2448 pixels. Weighted accuracy $Acc_w(\beta)$ in eqn. (6.11) with $\beta = 0.5$ (balanced accuracy) is used as the performance measure.

Table 4.3 shows that all the algorithms' overall means of the weighted accuracy $Acc_w(\beta)$ ($\beta = 0.5$) means of experimental trials under all instance pair names ("Melt 1", "Melt 2", "Melt 3", "Plus Metal" and "Scuff") stated in Table 6.5.

Table 6.6 Instance pair details

	Extended SRA	MR	SF
Means	0.6124	0.009	0.01

The results in Table 4.3 shows that the proposed extended *SRA* outperforms both *MR* and *SF* significantly in term of the means of accuracy means.

6.8.3. Performance Comparisons of Original and Extended SRAs

In this subsection, the same metallic object dataset in Section 6.8.2 is used. All low image resolutions used in the first sparse reconstruction of the proposed extended *SRA* are fixed at 5×5 (width $\times$ height) pixels. The high image resolution (output image resolution) used in the experiments of the proposed extended *SRA* and the image resolution used in experiments of the proposed original *SRA* is one of 5×5 , 10×10 and 25×25 pixels.

Figure 6.18 shows the same performance measures shown in Figure 6.15 for both proposed original and extended *SRAs* on same dataset used in generating the results in Figure 6.15. The X and Y-axes in the charts in Figure 6.18 are the same as shown in Figure 6.15.

Figure 6.18 shows that the extended *SRA* outperforms the original *SRA* when low output image resolution is used. The performance of both approaches converges at high output image resolution.

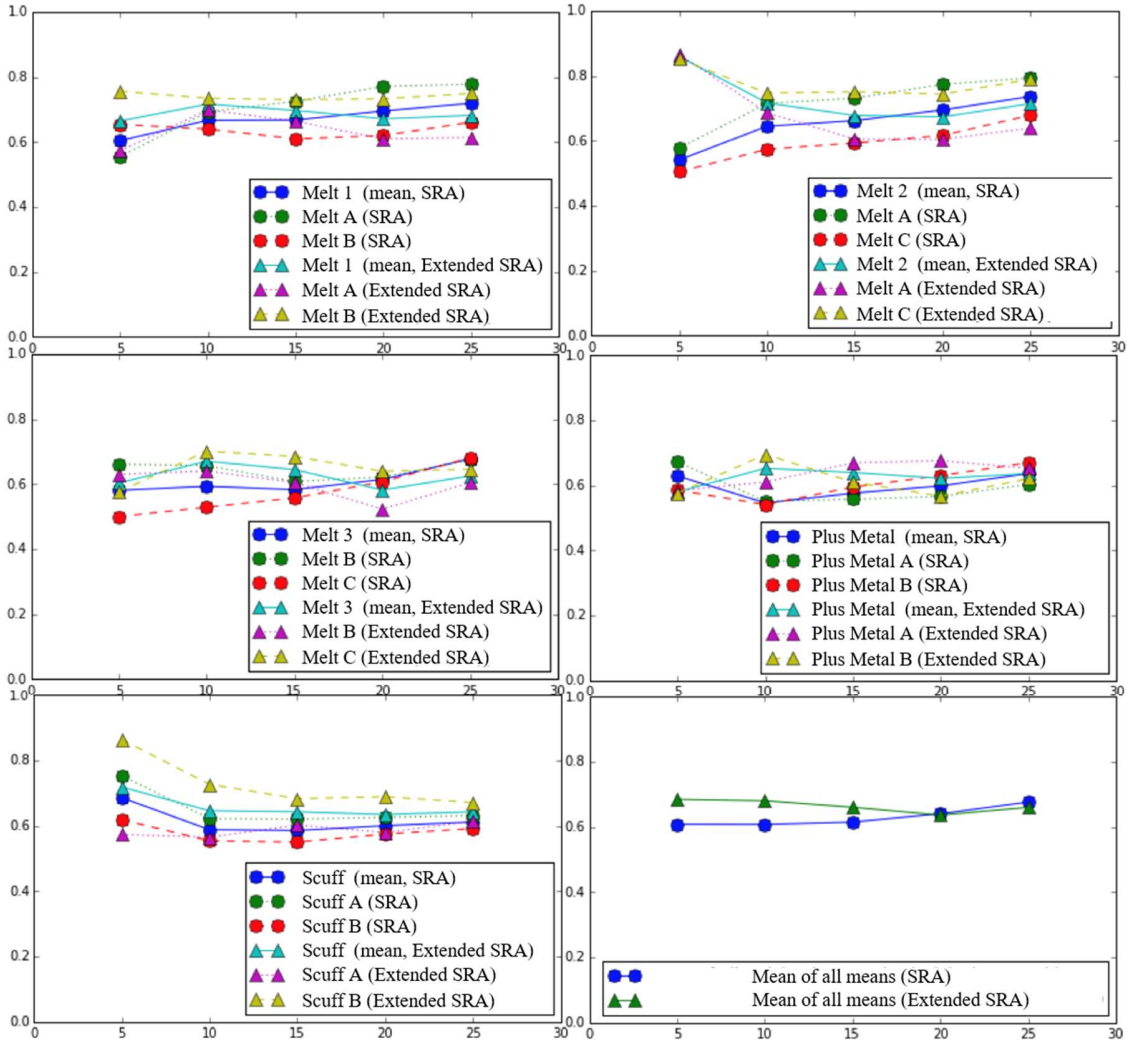


Figure 6.18 Performance comparisons of original and extended *SRAs*

Note that extended *SRA* used an average of 0.0446 seconds to resolve the optimization problem eqn. (6.7) with the linear programming function of Matlab when the low image resolution is 5×5 pixels and the high image resolution is 25×25 pixels. The original *SRA* takes an average computation time of 3.5727 seconds to resolve the optimization problem with the same function of Matlab for an output image resolution of 25×25 pixels. Two identity matrices corresponding to the residuals e in the equality in eqn. (6.2) consumes 10GB RAM memory when the image resolution of 160×160 pixels is used with the original *SRA*. Hence, the extended *SRA* is significantly more practical for high resolution output.

6.9. Summary

An approach which applied eqn. (1.2) in the problem design for a real-world anomaly detection and localization problem and its performance are presented (see Figure 1.1 and Figure 1.2). The novel anomaly detection and localization approach uses the residuals information from sparse reconstruction. Two extensions (SRA [5] and extended SRA [6]) are presented for an industrial application; namely: defect detection and localization for automated visual inspection (AVI). The second extension resolves the computation time and memory demand issues of the first extension. The empirical results demonstrate the feasibility of the proposed approaches on a real-world metallic turbine blade image dataset which is obtained from an industrial setting. Moreover, the results also showed that the second extension outperformed two recently proposed saliency detection algorithms; that are: graph-based manifold ranking (MR) [472] and spatial filtering (SF) [473] in the industrial application. Note that a closed work which utilizes eqn. (1.2) for training ensemble classifiers for multi-class classification is presented in Appendix B. The approach has been demonstrated on an industrial application, defect classification, with the dataset acquired from the same products (aircraft turbine blade) and defects presenting in the datasets used in the experiments in this chapter.

Chapter 7

Conclusions and Future Works

“Without continual growth and progress, such words as improvement, achievement, and success have no meaning..”

- Mark Twain

This chapter concludes the Thesis and discusses the future works. Section 7.1 presents the conclusions and Section 7.2 elaborates on the future works.

7.1. Conclusions

Referring to Figure 1.1 and Figure 1.2, this thesis focusses on proposing techniques and theoretical works for solving problem design and optimization techniques issues to boost the performance and improve the feasibility of the sparse representation problem. Four works, namely:

- A novel second order composite convex minimization technique (Chapter 3)
- Representation recovery via L_1 -norm minimization with corrupted data (Chapter 4),
- A feature selection technique for dictionary transfer (Chapter 5), and
- Anomaly detection and localization (Chapter 6).

are presented.

The first work aims at improving the performance of the state-of-the-art optimization techniques for sparse representation problem. The proposed approach integrates first and second order techniques into a single novel parallelable Jacobi iteration technique for composite convex minimization problems. It improves the convergence speed and accuracy from the other state-of-the-art Jacobi iteration techniques which only use first order information. The technique can be used for L_1 -norm minimizations which are widely used in sparse representation problems.

The second work presents a theoretical work to show the performance, lower bound of successful recovery probability of the original sparse representation, of an L_1 -norm minimization which is used as the recovery problem design from the corrupted data. The work generalizes a the work presented by Wright et al. [147] by introducing a regularized parameter in the problem and a more relaxed assumption which can be fulfilled by many more applications. Simulation results shows that the new L_1 -norm minimization performs well in recovering the original sparse representation and much better than the L_1 -norm minimization use in [147] under both highly corrupted simulated and practical data.

The third work presents an idea for solving a common feasibility issue in reusing dictionary in a new computational unit when sparse representation problem eqn. (1.2) is used in the problem design. It resolves the curse of dimensionality issues of sparse representation problem with high dimensional dictionary. The method converts an infeasible problem which is ported to a new computational unit of either poorer computational power or lesser memory than the original unit used to learnt the dictionary. The method is proposed for replacing the naïve downsampling method which are widely used in constructing a low dimensional dictionary. The underlying of dictionary training process and relationship of feature are assumed to be unknown. The empirical results show that the proposed method outperforms the naïve downsampling method in the benchmark with simulated and practical data.

The last work introduces an approach which employs eqn. (1.2) in the problem design for a real-word anomaly detection and localization problem as well as investigating its performance. A template-based anomaly detection and localization technique via sparse representation is presented. The approach has been demonstrated in a very important task, visual inspection, in many industries. Nowadays, no suitable automated inspection technique exists to be the benchmark of a challenging image-based inspection problem involves complicated surfaces to be inspected and different illumination conditions. A detailed analysis using different dictionary construction methods were performed. A comprehensive analysis of the effects of the changes of parameters and resolutions of input and training images were also conducted. Multiple state-of-the-art saliency methods which

is used to generate an intermediate image in the process are used in the comparison. The results show that the proposed sparse representation generate a better intermediate image with better detection and localization accuracy.

7.2. Future Works

7.2.1. Composite Convex Minimization Technique

The proposed work in Chapter 3 does not integrate an adaptive parameter tuning technique into the algorithm as the work presented by Shen et al. [1]. An adaptive parameter tuning with quasi-Newton method can be introduced into the technique. Moreover, Broyden–Fletcher–Goldfarb–Shanno (BFGS) being used can be replaced by limited-memory version of BFGS (L-BFGS), BFGS’s low memory extension, for memory saving.

7.2.2. Representation recovery via L_1 -norm minimization

The L_1 -norm minimization (eqn. (1.2)) which theoretical sparse representation recovery probability is analysed in Chapter 4 is also a Lagrangian function of another L_1 -norm minimization in eqn. (2.17). The relationships between the parameters λ (see eqn. (1.2)) and ε (see eqn. (2.17)) and two L_1 -norm minimizations in eqns (1.2) and (2.17) can be further analysed. Moreover, the feasibility of choosing the parameter λ in eqn. (1.2) based on analyzing the formula of the theoretical recovery probability lower bound can be analysed while a particle swarm algorithm (PSO) is used to choose the parameter in the current works.

7.2.3. Feature Selection Technique for Dictionary Transfer

Converting an infeasible one-time sparse representation problem to a feasible sequential sparse representation problem as shown in Figure 7.1 can be explored. The notations can refer to eqn. (1.2) while f_{ij} represents the i^{th} feature of j^{th} atom. Solving a sparse representation in one-time manner can be infeasible due to the high dimensional dictionaries. Since then, a feasible sequential optimization with low dimensional dictionaries in each optimization can replace it. In the initial optimizations, dictionaries with low number of features (row) and original number of atoms (column) are used. In each

newer optimization, dictionaries with higher number of features and lower number of atoms are used. Several atoms are removed in each optimization and the atoms to be removed are chosen based on the sparse representation results (values of elements in coefficient vector x) of previous optimization. The proposed technique in Chapter 5 can be used to select representative features for lower dimensional dictionary constructions. The work in Chapter 4 can be used for the derivation of the theoretical sparse representation recovery probability via the sequential optimization.

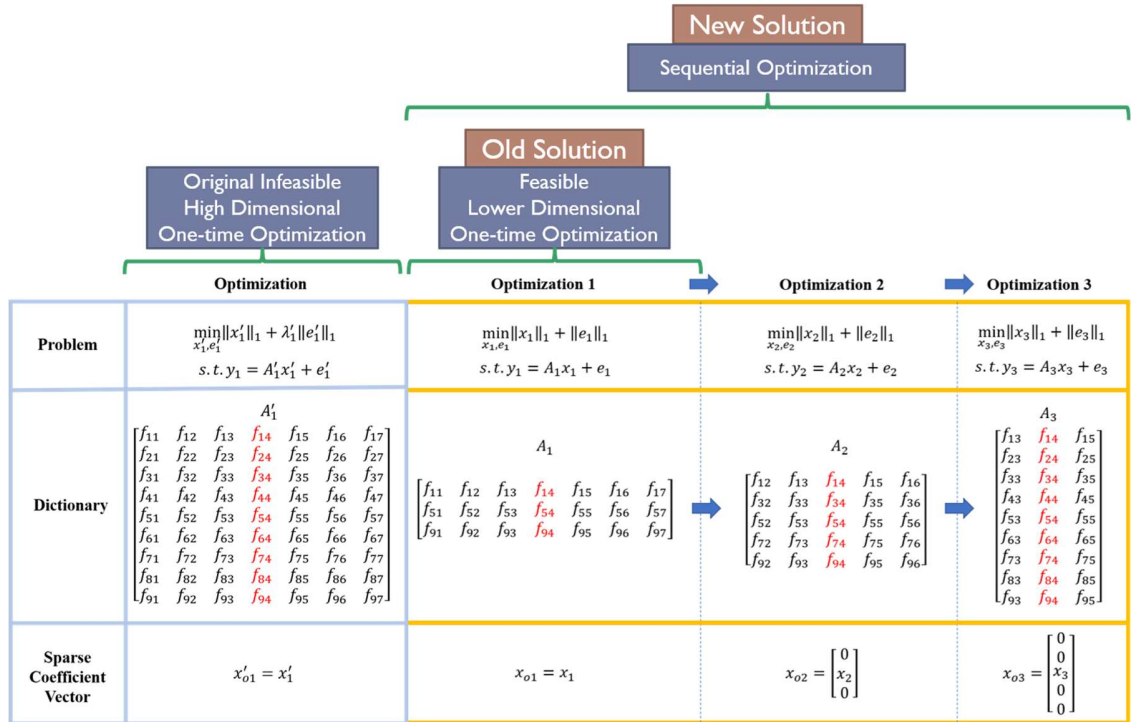


Figure 7.1 Sequential Optimization

7.2.4. Anomaly detection and localization

Shadow removal and alignment techniques can be integrated to the proposed work in Chapter 6. Moreover, feature selection technique proposed in Chapter 5 can be used for dictionary constructions. Furthermore, the sequential optimization introduced in previous section can be used for enhancing the estimation of the coefficient vector x and error vector e and hence boost the anomaly detection and localization accuracy. On the other hand, leaning-based dictionary can be explored too.

Appendix A (Appendix for Chapter 2)

A.1. Characteristics of Surface Defects

This section reviews the characteristics of various surface defects.

A.1.1. Fabric Surface Defects

The surface patterns of fabric product can be categorized into 17 wallpaper groups with particular symmetrical rules [474]. These patterns are called patterned texture. Surface defects on fabric products are anomalies which do not conform with the symmetrical rules of the surface pattern. Figure A.1 shows examples of fabric surface defect [369].

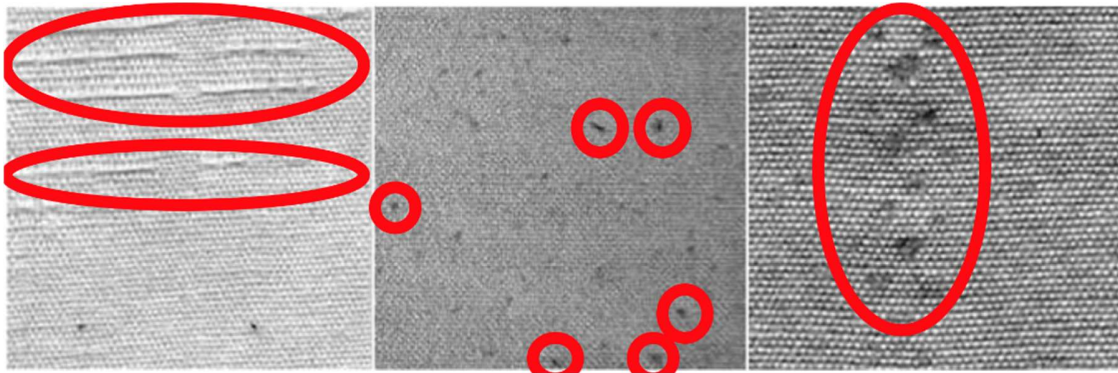


Figure A.1 Examples of fabric surface defect: (left to right) directional defect, disseminated defect and localized defect [369]

A.1.2. Random Texture Surface Defects

Another type of texture is called random texture. Some construction components contain random texture surfaces. Random texture does not follow any fixed symmetric rule. It is constructed by primitives with specific spatial and colour distribution which can be captured by a particular feature extraction [475]. Random texture defect is the anomaly which does not comply the characteristics of its texture primitives. Figure A.2 shows the examples of random texture surface defect [347], [358], [359].

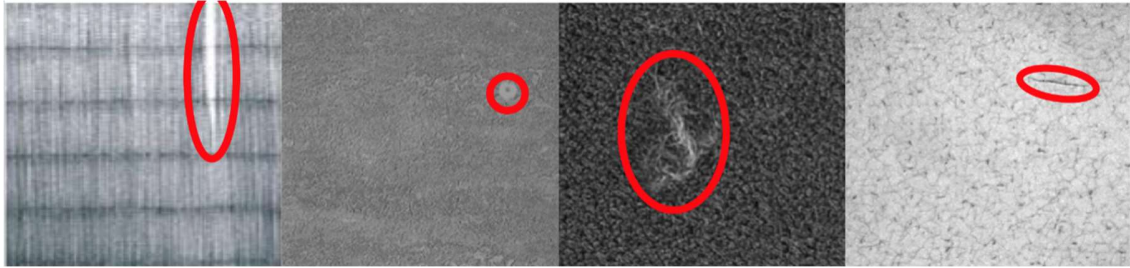


Figure A.2 Examples of random texture surface defect [347], [358], [359]

A.1.3. Electronic Device Surface Defects

In electronic device production, defect may exist in the integrated circuit (IC) [348], printed circuit board (PCB) assembly process [364], [476], soldering process [366], etc. They are the regions where components are missing or wrongly placed, contain improper solderings or damages, etc. Figure A.3 shows some examples of PCB assembly defect [348], [364], [476]. The right first image is the defect free template of the right second image.

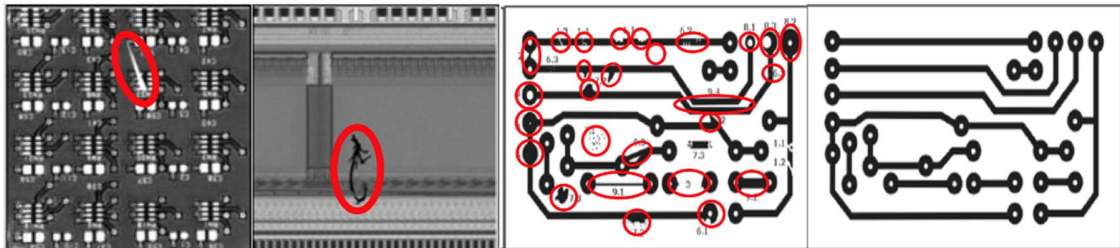


Figure A.3 Examples of PCB surface defect [348], [364], [476]

A.1.4. Complex Surface Defects

In some productions which contain complex manufacturing processes, complex defect may exist. The defect can be caused by factors of:

- Extreme environmental characteristics (i.e. high temperature, high pressure, etc.) in manufacturing process,
- Mistakes in polishing for complex surface finish patterns,
- External materials on the mold, and
- Unexpected solidification outcome of crystallized materials; etc.

Figure A.4 shows some examples of complex defect which exist in complex manufacturing processes [5].

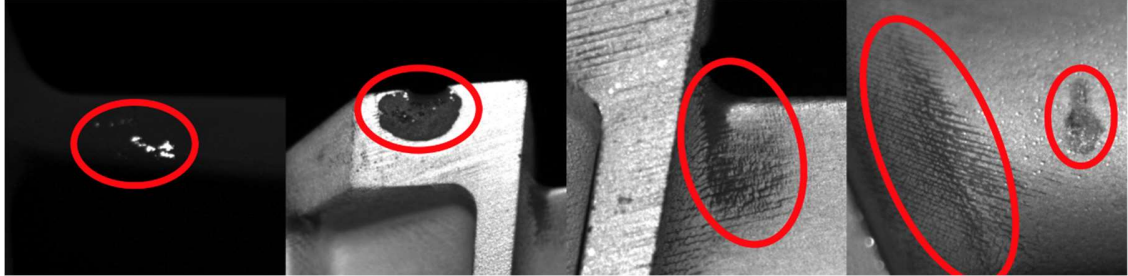


Figure A.4 Examples of complex surface defect [5]

The other two industries where products may contain complex defect is the fruit and vegetable industries. The techniques developed for detecting defects in these two industries are not covered in this literature review.

A.2. Details of published AVI techniques

Table A.1 to Table A.5 shows the details of the proposed techniques presented in Table 2.5. It includes classifiers and features used in the techniques, names of the inspected defect classes, empirical results, inspected surface characteristics (shape, colors and materials), illumination condition used and inspection types of the techniques presented in Table 2.5. Section A.2.1 to A.2.6 further elaborates the details of the techniques.

Table A.1 Classifiers used in AVI Techniques

Techniques	Classifier
O-TM	-
Z-GA	Similarity with Manhattan distance
K-KNN	K-nearest neighbors (k-NN)
X-TM	Template matching
N-M	Decision boundary approach
S-I	Thresholding, segmentation
J-OSVM	One-class support vector machine (One-class SVM)
A-PHOT	Thresholding
R-C4.5	C4.5
A-WHC	Ward's hierarchical clustering
Z-SVM	Support vector machine (SVM)
M-MPCNN	Convolutional neural network (CNN)

S-TM	Otsu algorithm
A-WHC-2	Ward's hierarchical clustering
B-HMMPP	Hierarchical multi-marked point process (HMMPP)
W-TM	Thresholding
R-SW	Template matching with thresholding
G-SVM	Vector-valued regularized kernel function approximation (VVRKFA)
R-T	Thresholding
S-FL	Fuzzy logic
G-S	Maximum between-cluster variance approach with region growing approach
L-FDCT	Rule (set by expert) based classifier
Y-SVM	SVM
K-FPM	Feature point matching
C-FFANN	Feed-forward artificial neural network
T-MS-SSIM	Multiscale structural similarity index (MS-SSIM)
L-T	Thresholding
C-TWSVM	Twin support vector machine (TWSVM) and binary tree
G-ELM	Extreme learning machine (ELM)
G-T	Thresholding
P-RF	Random forest
L-DCDR	Thresholding and least square SVM (LSSVM)
Z-PSVM	Polar support vector machine (PSVM)
J-SL	Thresholding
J-PW	Detection rules with thresholding
K-TM	Thresholding and matching
C-SR	Thresholding
R-T-2	Thresholding
C-SOMO	Fuzzy-based hyper-rectangular composite neural networks (FHRCNN)
D-SDDP	Thresholding and regression
O-FWT	Thresholding
R-DL	Deep learning
B-T	Thresholding
C-NB-CNN	Convolutional neural network (CNN)
Y-DCNN	Deep CNN
W-TM-2	Thresholding
M-DL	CNN
G-BODI	Thresholding
L-SVM	SVM
O-FL	Fuzzy logic

H-ROI	Thresholding
K-DL	Deep convolutional neural networks (CNNs)
M-RF	Random Forest (RF)
S-DT	Decision tree with thresholding
E-MR-ELM	ELM
C-FCN	Deep full convolutional networks (FCN)
W-R-CNN	Region-based CNN (R-CNN)
C-DL	Shallow CNN

Table A.2 Features used in AVI Techniques

Techniques	Features
O-TM	Difference between image intensities and phase differences
Z-GA	Fusion of two shape descriptor, shape geometric descriptor and generic Fourier descriptor
K-KNN	Multiscale Fourier and Boundary Scale Fourier descriptors
X-TM	Local common vector (local salient chromatic feature)
N-M	Energy and variance of moving subtraction
S-I	Hough Transform, Gaussian function, gray pixel histogram, Renyi entropy
J-OSVM	Haralick features extracted from gray-level co-occurrence matrix (GLCM) of different wavelet subbands
A-PHOT	Phase only transform (PHOT) and Mahalanobis distance
R-C4.5	Histogram features
A-WHC	Energy extracted from several periodic blocks of Gabor wavelet transformation in multi-scales and multi-orientations
Z-SVM	Spectral measure of binary images generated with Otsu thresholding, segmentation and wavelet smoothing
M-MPCNN	Not available (N.A.)
S-TM	Difference between connected-component labellings extracted from input image and template
A-WHC-2	Gray value in images
B-HMMPP	Data and prior energies
W-TM	Partial information correlation coefficient (PICC)
R-SW	Sliding window-based textural features, energy values, extracted from GLCM
G-SVM	Three-level Haar
R-T	Energy based features extracted from discrete cosine transform (DCT) of a homogeneity image (H-image)
S-FL	Textural features extracted from GLCM
G-S	Wavelet feature maps and sub-maps and saliency map
L-FDCT	Fast discrete curvelet, texture features extracted from GLCM and length, area and Hu moments of the extracted contours
Y-SVM	Wavelet
K-FPM	Corner feature point
C-FFANN	Texture features extracted from gradient-only co-occurrence matrices (GOCM)
T-MS-SSIM	Luminance metric, contrast metric and structure metric
L-T	Image processed with an enhancement operator based on mathematical morphology (EOBMM)
C-TWSVM	Statistical texture features extracted from two images generated with applying the technique of combination of concentric discrete square

	sampling template (CDSST), GLCM and smoothed local binary pattern (SLBP)
G-ELM	Histogram of oriented gradients (HOG)
G-T	Gaussian pyramid decomposition, image fusion, saliency map construction
P-RF	Texture, geometrical, statistical and intensity features extracted from the neighbouring different filter (NDF)-processed image
L-DCDR	Pixels values in Red-Green-Blue (RGB) and Hue-Saturation-Intensity (HSI) colour spaces
Z-PSVM	Wavelet and Radon transforms
J-SL	Sub-optimal filtering, clustering, texture features (mean value, contrast, smoothness, uniformity, third moment, entropy and moment invariants)
J-PW	Solder ball area, center of gravity and compactness of the solder balls
K-TM	Binary Robust Independent Elementary Features (BRIEF), blob and edge
C-SR	L_2 -norm of residual and L_1 -norm of coefficient vector of estimated sparse representation
R-T-2	Pre-processed binary image which are generated based on the pixel value distributions
C-SOMO	Local Directional Pattern
D-SDDP	Estimated spalling depth
O-FWT	Textons representation generated by wavelet transform
R-DL	N.A.
B-T	Overlapping area of the real ball and a model ball and the areas of the real ball and the model ball.
C-NB-CNN	N.A.
Y-DCNN	N.A.
W-TM-2	N.A.
M-DL	N.A.
G-BODI	Gray value, mean value and median value of spatial neighborhood
L-SVM	Local frequency features
O-FL	Red, Green Blue values of image pixel and the luminance value of the surface
H-ROI	Shape and size of Region of Interest (ROI)
K-DL	N.A.
M-RF	“GIST of scene” (GIST)
S-DT	Defect areas, defect brightness, gray-value curves
E-MR-ELM	Similarity measure (a kind of minimum ratio between local neighborhood sliding window)
C-FCN	N.A.
W-R-CNN	N.A.
C-DL	N.A.

Table A.3 Defect Classes Inspected by AVI Techniques

Techniques	Class Names
O-TM	Phase defect (edge defect, center defect, corner defect, bump defect and divot defect)
Z-GA	Hole and crack
K-KNN	N.A.
X-TM	Normal and abnormal (defect)
N-M	Defect-free and defective lattices
S-I	Hole, scratch, coil break and rust defect
J-OSVM	Normal and abnormal
A-PHOT	Non-defective region and defected patterns
R-C4.5	Good, minor scratch and deep scratch
A-WHC	Defect-free and defective periodic blocks
Z-SVM	Burr, loophole, oil stain, peeling, perforation, pit and scratch
M-MPCNN	N.A.
S-TM	OK, Not OK (missing or wrongly placed components)
A-WHC-2	Defect-free and defective periodic blocks
B-HMMPP	Solder paste (SP) and scoop object (SO)
W-TM	Particles, foreign matters and distorted components
R-SW	Defective image samples
G-SVM	Defect-free, light scratch, dark scratch, head mark, heavy scale, longitudinal crack, massive rupture, m-defect, blister, bruise, rolled in dust, rolled in scale, roll mark, sagger, salt and pepper, scale grain, scale pits, seam, slab edge, spall, waviness, whip, deposits, sticker break and damage
R-T	Non-defect and defect blocks
S-FL	Fault free, holes
G-S	Directional, localized and disseminated defects
L-FDCT	Crack
Y-SVM	Normal and defect
K-FPM	Normal and defect
C-FFANN	Non-defective, blisters, scratches and die-lines
T-MS-SSIM	Wrinkles, gels, coating voids, die lines (streaks), structural defect, dark areas, light areas, and intensity defects
L-T	N.A.
C-TWSVM	Rust, hole, scratch and scale
G-ELM	Background, dent, fold, scarring, scale, bruise, hole and damage
G-T	N.A.
P-RF	Normal and defect (both real (scratches and pits) and fake (stains, long dust and circular dust) defects)
L-DCDR	Normal and defect

Z-PSVM	Crack and non-crack
J-SL	Normal and defect
J-PW	Normal and defect
K-TM	Speckle and Scratch
C-SR	Normal and anomalous patch
R-T-2	Crack and non-crack
C-SOMO	Crack and non-crack
D-SDDP	Spall and background
O-FWT	Pothole and defect-free
R-DL	Normal and defect
B-T	Normal and defect
C-NB-CNN	Crack and non-crack
Y-DCNN	Crack and non-crack
W-TM-2	Normal and defect
M-DL	Miscellaneous, empty duct, dust, sand and swarf
G-BODI	Normal and defect
L-SVM	Corrugation and background
O-FL	Normal and defect
H-ROI	Normal and defect
K-DL	Root, intrusion, deposits, crack and normal
M-RF	Faulty and normal
S-DT	Normal and defect
E-MR-ELM	Normal and defect
C-FCN	Crack and non-crack
W-R-CNN	Efflorescence and spalling damages
C-DL	Crack and non-crack

Table A.4 AVI Techniques' Performance

Techniques	Results
O-TM	Detection: N.A.
Z-GA	Detection accuracy: 91% (hole), 86% (crack)
K-KNN	5-NN classification accuracy: 48.42% (Multiscale Fourier descriptor) and 43.85% (Boundary Scale Fourier descriptor)
X-TM	Detection accuracy: 93.9%
N-M	Detection accuracy: 93.32% (patterned fabric) and 93.86% (fabrics, wallpaper ceramics, ceramics, ornament, tile, etc.)
S-I	Detection accuracy: 86.78%
J-OSVM	Detection: 0.06% false rejection rate and 12.5% false acceptance rate
A-PHOT	Detection: N.A.
R-C4.5	Classification accuracy: 90.4%
A-WHC	Detection: 93.9%, 100% and 68.8% (accuracy, precision and recall of pmm wallpaper group), 93.9%, 100% and 68.6% (p2 group) and 99.6%, 100% and 90.2% (p4m group)
Z-SVM	Classification accuracy: 85%
M-MPCNN	Classification accuracy: 93%
S-TM	Detection: N.A.
A-WHC-2	Detection: 98.8% (average accuracy, 97.9%, 99.3% and 99.3% for pmm, p2 and p4m respectively), 100% (average precision for all pmm, p2 and p4m) and 83% (86.4%, 80.0% and 82.6% for pmm, p2 and p4m respectively)
B-HMMPP	Detection: 99.96% (SP object level F-rate), 93.90% (SP area F-rate), 1.36 (SP average orientation error), 92.40% (SO object level F-rate) and 83.04% (SO area F-rate), SP: solder paste, SO: scoop object
W-TM	Detection accuracy: 100%
R-SW	Detection: N.A.
G-SVM	Detection: Average of G-mean and F-measure of 93.8 and 90.4% for defect detection in images which contain defects and average accuracy of 99.70% for defect detection in defect-free images
R-T	Detection: 96.55%, 95.98% and 96.9% (average sensitivity, specificity and detection success rate)
S-FL	Detection: N.A.
G-S	Detection: 97% (directional defect), 99% (localized defect) and 96% (disseminated defect)
L-FDCT	Detection: 93.9% accuracy, 3.57% false acceptance rate and 8.5% correct rejection rate
Y-SVM	Detection accuracy: 92.68%
K-FPM	Detection: 94% accuracy, 95% sensitivity and 90% specificity
C-FFANN	Classification accuracy: 98.9%
T-MS-SSIM	Detection: 99.1% accuracy, 97.7% recall/sensitivity and 100% specificity
L-T	Detection accuracy: 84.16%

MR-SLBP- GLCM -TWSVM	Classification accuracy: 88.14%
G-ELM	Detection: 97.5% (defect detection), 99% (background detection)
S G-ATA	Detection: 96% accuracy and 5% accuracy volatility
P-RF	Detection: mean accuracy, recall and precision of 92%, 84.67% and 83.67% and 0.99 ratio of correctly classified pixels (RCCP) per image
L-DCDR	Detection accuracy: 86%
Z-PSVM	Detection: 95.59% accuracy, 89.1% sensitivity, 97.1% selectivity, 88.56% repeatability, 2.88% Type I error, 11.44% Type II error, 11.4% false discovery rate, and 2.84% for false omission rate
J-SL	Detection: 0.9799 (20 clusters, average performance index proposed by the author)
J-PW	Detection accuracy: 97.89%
K-TM	Detection accuracy: 85.95% (average on 2 datasets)
C-SR	Detection AUC: 92.6%
R-T-2	Detection: N.A.
C-SOMO	Detection accuracy: 89% (crack), 91% (non-crack)
D-SDDP	Detection: 89.3% accuracy, 91.7% recall, 94.8% precision
O-FWT	Detection: 87.5% average Dice coefficient of similarity and 8.5% mean and 4.9% standard deviations for the errors
R-DL	Classification: 99.21% and 94.29% accuracies in patch classification for hot-rolled steel strip and wood inspection Segmentation: 10.06%, 26.23% and 11.6% of error escape, false alarm and misclassification rate in defect segmentation for wood inspection
B-T	Detection: N.A.
C-NB-CNN	Detection: 98.3% hit rate against 0.1 false positive per frame
Y-DCNN	Detection accuracy: 80.7% Localization accuracy: 86%
W-TM-2	Detection accuracy: 96.2%
M-DL	Classification: 91.82% (accuracy in experiment) and 1.229% (miss rate in real scenario)
G-BODI	Detection accuracy: 100%
L-SVM	Detection: 98.47% precision and 96.5% recall
O-FL	Detection: 83.5% accuracy, 99.9% sensitivity and 99.9% specificity
H-ROI	Detection: 73.7%, 52.9%, 65.3% and 100% positive predictive values for cracks, settled deposits, ovality and displaced joint defect respectively.
K-DL	Detection: 86.2% (average accuracy), 87.7% (average precision) and 90.6% (average recall)
M-RF	Detection: 80% accuracy and 89% area under the curves (AUC)

S-DT	Detection accuracy: more than 95% (experiments) and 99% (actual production)
E-MR-ELM	Detection: 98.07% accuracy, 91.29% sensitivity and 99.67% specificity
C-FCN	Detection: 89.3% maximum F1 score and 89.3% average precision
W-R-CNN	Classification: 99.9% and 90% average precisions for the two damage categories, respectively, with a 95% mean average precision (mAP) in the best case
C-DL	Detection accuracy: 98%

Table A.5 Summary of AVI Techniques

Inspection Characteristics		K-KNN	S-I	R-C4.5	M-MPCNN	G-SVM	L-FDCT	C-FFANN	T-MS-SSIM	G-ELM
Surface Characteristics	Uniform Surface	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Uniform Random Texture Surface									
	Patterned Textured Surface									
	Non-uniform Surface									
	Non-Complex Surface									
	Complex Surface									
Illumination Condition	Uniform Illumination	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Non-uniform Illumination									
Technique Type	Defect Detection and/or Localization		✓			✓	✓		✓	✓
	Defect Classification	✓		✓	✓			✓		

Inspection Characteristics		L-DCDR	Z-PSVM	W-TM-2	O-FL	Z-SVM	Y-SVM	L-T	C-TWSVM	G-T
Surface Characteristics	Uniform Surface	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Uniform Random Texture Surface									
	Patterned Textured Surface									
	Non-uniform Surface									
	Non-Complex Surface									
	Complex Surface									
Illumination Condition	Uniform Illumination	✓	✓	✓	✓					
	Non-uniform Illumination					✓	✓	✓	✓	✓
Technique Type	Defect Detection and/or Localization	✓	✓	✓	✓		✓	✓		✓
	Defect Classification					✓			✓	

Inspection Characteristics		J-SL	G-BODI	X-TM	J-OSVM	C-SOMO	D-SDDP	O-FWT	G-S	C-NB-CNN
Surface Characteristics	Uniform Surface	✓	✓							
	Uniform Random Texture Surface			✓	✓	✓	✓	✓	✓	✓
	Patterned Textured Surface									
	Non-uniform Surface									
	Non-Complex Surface									
	Complex Surface									
Illumination Condition	Uniform Illumination			✓	✓	✓	✓	✓		
	Non-uniform Illumination	✓	✓						✓	✓
Technique Type	Defect Detection and/or Localization	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Defect Classification									

Inspection Characteristics		M-DL	C-FCN	C-DL	N-M	A-WHC	A-WHC-2	R-SW	S-FL	P-RF
Surface Characteristics	Uniform Surface									
	Uniform Random Texture Surface	✓	✓	✓						
	Patterned Textured Surface				✓	✓	✓	✓	✓	✓
	Non-uniform Surface									
	Non-Complex Surface									
	Complex Surface									
Illumination Condition	Uniform Illumination				✓	✓	✓	✓	✓	✓
	Non-uniform Illumination	✓	✓	✓						
Technique Type	Defect Detection and/or Localization		✓	✓	✓	✓	✓	✓	✓	✓
	Defect Classification	✓								

Inspection Characteristics		J-PW	B-T	E-MR-ELM	R-T	W-R-CNN	L-SVM	Z-GA	R-T-2	O-TM
Surface Characteristics	Uniform Surface									
	Uniform Random Texture Surface									
	Patterned Textured Surface	✓	✓	✓	✓	✓				
	Non-uniform Surface						✓	✓	✓	
	Non-Complex Surface						✓	✓	✓	
	Complex Surface									✓
Illumination Condition	Uniform Illumination	✓	✓	✓			✓			✓
	Non-uniform Illumination				✓	✓		✓	✓	
Technique Type	Defect Detection and/or Localization	✓	✓	✓	✓		✓		✓	✓
	Defect Classification					✓		✓		

Inspection Characteristics		W-TM	K-FPM	K-TM	C-SR	S-TM	B-HMMPP	Y-DCNN	H-ROI	K-DL
Surface Characteristics	Uniform Surface									
	Uniform Random Texture Surface									
	Patterned Textured Surface									
	Non-uniform Surface									
	Non-Complex Surface									
	Complex Surface	✓	✓	✓	✓	✓	✓	✓	✓	✓
Illumination Condition	Uniform Illumination	✓	✓	✓	✓					
	Non-uniform Illumination					✓	✓	✓	✓	✓
Technique Type	Defect Detection and/or Localization	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Defect Classification									

Inspection Characteristics		M-RF	A-PHOT	R-DL
Surface Characteristics	Uniform Surface			✓
	Uniform Random Texture Surface		✓	✓
	Patterned Textured Surface		✓	
	Non-uniform Surface			
	Non-Complex Surface			
	Complex Surface	✓		
Illumination Condition	Uniform Illumination		✓	✓
	Non-uniform Illumination	✓		✓
Technique Type	Defect Detection and/or Localization	✓	✓	✓
	Defect Classification			

A.2.1. Techniques for Uniform Surface

Uniform Illumination

Kunttu et al. [353] presented an approach (K-KNN) for paper defect image retrieval. It is based on two Fourier-based multi-resolution shape descriptors, Multiscale Fourier and Boundary Scale Fourier descriptors, and k-nearest neighbour (k-NN) classifier. 5-NN classifier is used in the experiments. The results showed that the classification rates with the two Fourier-based descriptors respectively were 48.42% and 43.85% and outperformed the classification rate (41.87%) with the Contour Fourier descriptor (single-scale Fourier descriptor).

Sharifzadeh et al. [357] presented an image processing method (S-I) for defect detection and 4-defect-class classifications of steel inspection. Classification and detection rules were proposed for each defect type respectively with techniques of thresholding, segmentation, Hough transform, Gaussian function modelling, etc. Thresholding classifier is used. The results showed that the classification achieved 88.4%, 78%, 90.4% and 90.3% accuracies for detecting and classifying defects of a hole, scratch, coil break and rust from normal surface respectively.

Ravikumar et al. [360] proposed a machine learning-based defect classification approach (R-C4.5) for powder-coated metallic surface inspection. The approach utilizes a library of histogram features and a decision tree C4.5 classifier. The features are extracted from a binary image derived from the thresholding on the difference between ideal and input images. It is a 3-class (good, minor scratch and deep scratch) classification. The C4.5 algorithm achieved a misclassification rate of 9.6% comparing against 17.3% of a Naive Bayes classifier.

Masci et al. [363] proposed a defect classification approach (M-MPCNN) for steel surface inspection with Max-Pooling Convolutional Neural Networks (MPCNN). The input of the approach are grey-images and the problem contain 7 defect classes. The best MPCNN

achieves an error rate of 7% comparing with 15% of Support Vector Machine (SVM) trained using Pyramid of Histograms of Orientation Gradients (PHOG) features. The results are obtained from raw pixel intensities of the detected and segmented steel defects without any prior knowledge. The author also claimed that the scalability of MPCNN is well as the approach can be extended for the inspection problem with the input of multivariate images (colour, hyperspectral).

Ghorai et al. [367] investigated the performances of various machine learning algorithms (G-SVM) for defect detection of hot-rolled steel strip surface inspection. Kernel-based classifiers such as SVM and Vector-Valued Regularized Kernel Function Approximation (VVRKFA) are used. The classifiers are used with various wavelet feature sets, texture-based segmentation and thresholding technique. Binary classification (1 of 24 classes of defects and a defect-free class) are conducted. The results show that the three-level Haar feature set outperformed the other wavelet feature sets (Daubechies 2 (DB2), Daubechies 4 (DB4), biorthogonal spline and multiwavelet) as well as the texture-based segmentation or thresholding defect detection technique. The best performance was obtained by the combination of VVRKFA and three-level Haar feature. Two performance measures; namely, *G – mean* and *F – measure* which are defined in eqns (A.1) and (A.2) are used.

$$G - mean = \sqrt{TPr \times TNr} \quad (A.1)$$

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (A.2)$$

where *TPr* (TP rate), *TNr* (TN rate), *Precision* and *Recall* are defined in eqns (A.3)-(A.6).

$$TPr = \frac{TP}{TP + FN} \quad (A.3)$$

$$TNr = \frac{TN}{TN + FP} \quad (A.4)$$

$$Precision = \frac{TP}{TP + FP} \quad (A.5)$$

$$Recall = \frac{TP}{TP + FN} \quad (A.6)$$

where *TP*, *TN*, *FP* and *FN* are true positive, true negative, false positive and false negative.

The combination achieves averages of $G - mean$ and $F - measure$ of 93.8% and 90.4% for defect detection in images which contain defects and an average accuracy of 99.70% for defect detection in defect-free images. The limitation of the proposed method is that it is not able to perform in real-time when the rolling speed is greater than 5m/s. Other than that, the investigation assumes that no noise appears in the images due to the uneven illumination and vibration of the sheets in the image acquisition phase. On the other hand, there is an increase of false positive near to the defective region if there are specular reflections induced by the deformation due to certain type of defects (such as waviness, damages, etc.). The defects being investigated in the experiments are Light scratch, Dark scratch, Head mark, Heavy scale, Longitudinal crack, Massive rupture, M-defect, Blister, Bruise, Rolled in dust, Rolled in scale, Roll mark, Sagger, Salt and pepper, Scale grain, Scale pits, Seam, Slab edge, Spall, Waviness, Whip, Deposits, Sticker break and Damage.

Li et al. [370] presented a crack detection approach (L-FDCT) for ferrite magnetic tile inspection using Fast Discrete Curvelet Transform (FDCT) and texture analysis. Canny operator is applied on the image reconstructed from inverse FDCT using thresholded FDCT coefficients to extract crack defects' contours. The purpose of thresholding is to remove the grinding texture which causes false alarms in the defect detection. The threshold is determined from texture feature measurements (variance, entropy, homogeneity, angular second moment (energy) extracted from GLCM of the input image). The length, area and Hu moments of the extracted contours are used to determine whether the extracted contour belongs to the defect. The results showed that the approach is better than another approach which directly applies the Otsu algorithm and morphological filtering algorithm on the input image. The accuracy rate is up to 93.9% and it can accurately extract cracks which are greater than 0.8mm. The false acceptance rate and correct rejection rate are 3.57% and 8.5% respectively. The approach strictly needs specific lighting condition and field of view. It needs diffuse reflection and bright field to make the crack presentations clearer.

Chondronasios [373] proposed a defect classification approach (C-FFANN) for extruded aluminium surfaces inspection. A novel texture features extracted from Gradient-Only Co-occurrence Matrices (GOCM) of ROI (anomaly) patches is used for classifying the patches.

Feed-forward artificial neural network classifier is used. The classifier is trained with a machine learning approach. The features are similar to the commonly used texture features extracted from Gray-Level Co-occurrence Matrix (GLCM). The difference between the two is that GOCM's statistical results are extracted from gradient patch generated by applying the Sobel operation on grey-level patch. The statistical results of GLCM are directly extracted from the grey-level patch. Experimental results showed that the classification accuracy is 98.9% comparing with 55.9% which apply the texture features extracted from GLCM in a 4 class (non-defective, blisters, scratches and die-lines) classification case study.

Tolba and Raafat [374] proposed a defect detection approach (T-MS-SSIM) for inspection of flat surface products with a multiscale structural similarity index (MS-SSIM) and features of image luminance, contrast and structure. The approach can be used on surface inspection of thin films, paper, foils aluminium plates, steel slabs, fabrics, glass sheets, etc. Thresholding classifier on MS-SSIMs was used. They applied that to the inspection of flat surface products such as thin films, paper, foils aluminium plates, steel slabs, fabrics, and glass sheets. The defects can be wrinkles, gels, coating voids, die lines (streaks), etc. It is an index comparing two images, the input image and the defect-free reference image. The approach achieved an accuracy of 99.1%, precision of 98.62%, recall/sensitivity of 97.7% and specificity of 100% with quasi-real-time processing speed.

Gong et al. [346] presented an on-line defect detection framework (G-ELM) for steel strip surface inspection with grey-images acquired. The framework consists of three steps: noise filtering with a mean filter, feature extraction for 5 statistical projections and binary classification of defect and background with classifiers trained with Extreme Learning Machine (ELM). A background pre-detection algorithm is used for improving the classification. The framework is claimed to be able to work highly efficiently and fast. It can process 77 images per second (12.9ms per image) and can achieve near 100% accuracy (97.5% and 99% for defect and background detection respectively). The framework was tested on a steel strip surface image dataset which consists of images either containing one

of 7 defect types (dent, fold, scarring, scale, bruise, hole, damage, background) or do not contain a defect.

Liao et al. [378] proposed a rust defect detection approach (L-DCDR) for steel bridge inspection. The approach consists of K-means clustering in various colour space and a proposed Double-center-double-radius (DCDR) technique for determining the rusted area and radius. Least Square Support Vector Machine is used in determining the radius of rusted area in the DCDR. The proposed approach achieved 86% accuracy.

Zakeri et al. [379] proposed a crack defect detection approach (Z-PSVM) for asphalt pavement inspection. The approach involves image enhancement, wavelet transform, radon transform, flexible thresholding and classification with an expert system based on polar SVM (PSVM). The approach achieved 95.59% accuracy, 89.1% sensitivity, 97.1% selectivity, 88.56% repeatability, 2.88% Type I error, 11.44% Type II error, 11.4% false discovery rate, and 2.84% for false omission rate on average.

Wang et al. [392] proposed a template-based defect detection method (W-TM-2) for steel strip surface inspection. The method uses a large number of defect-free images to establish a statistical characteristic template of normal texture for a specific test image. Defect detection is carried out on the subtraction between the template and test images with an adaptive threshold. The method achieved 96.2% accuracy on a data set of 1500 test images.

Öztürk et al. [396] proposed a fuzzy logic-based defect detection method (O-FL) for glass surface inspection with segmentation. The method applies fuzzy logic on red, green and blue values in the images and the luminance of the glass surface in the classifying each pixel to either normal or defect class. The method achieved 83.5% accuracy, 99.9% sensitivity and 99.9% specificity.

Non-uniform Illumination

Zhang et al. [362] proposed an approach (Z-SVM) which utilizes texture features and one-against-one kernel-based SVM for surface defect classification of strongly reflected copper

strip inspection. Classification with the one-against-one scheme is applied. 7 defect classes (burr, loophole, oil stain, peeling, perforation, pit and scratch) were included. The features are the 5 characteristics of spectral measure of binary images which are the results of Otsu thresholding segmentation on input images after the noise removal with wavelet smoothing method. The test accuracy is 85% and outperformed the accuracy of 82.3% of Back-propagation neural network method.

Jeon et al. [371] proposed a defect detection approach (Y-SVM) for corner cracks detection of steel billet surface inspection using wavelet reconstruction and SVM. Weighted wavelet reconstruction is used for reducing the effect of scales. Texture and morphological features are extracted for identifying corner cracks among the defective candidates. SVM was used for a 2-class (defect and normal) classification. The method achieved 92.68% accuracy with 93.698 ms/frame.

Liu et al. [375] proposed a defect detection approach (L-T) for steel strip surface inspection. The approach is a binarization (thresholding) method based on genetic algorithm. The input image is processed with an enhancement operator based on mathematical morphology. The approach classifies pixels of grey-image into non-defective and defective pixels. Three performance measures; namely, misclassification error, edge mismatch and relative foreground area error are used in the experiments. The experimental results showed that the proposed approach has an accuracy (one minus the average of the three performance measures) of 84.16%. The author claimed that the algorithm can perform in a real time-industrial environment.

Chu and Gong [350] proposed a defect detection approach (C-TWSVM) for steel strip surface inspection with scale, rotation, illumination and translation invariant texture features. The features are extracted from images generated with techniques of concentric discrete square sampling template (CDSST), smoothed local binary pattern (SLBP) and GLCM. The input images are grey-images and were acquired under varying illumination. The features were claimed to be effective in suppressing noise. The defect classification

accuracy is greater than 88.14% for a four-class problem (rust, hole, scratch and scale) classification with twin support vector machine (TWSVM) and binary tree.

Guan [376] proposed an adaptive thresholding-based defect detection approach (G-T) for steel strip surface inspection. The images were acquired under non-uniform illumination and were applied with gaussian pyramid decomposition, image fusion, saliency map construction and thresholding. The experimental results showed that the algorithm can achieve an average accuracy of 96% and accuracy volatility of 5% in an average time of 0.35s. It outperformed the wavelet reconstruction algorithm which was proposed by Tsai and Chiang [477] and achieved an average accuracy and accuracy volatility of 86% and 5% in 0.732s.

Jeon et al. [380] proposed a defect detection approach (J-SL) for steel-plate surface inspection using a combination of a sub-optimal filtering and a switching-lighting method with texture features. The advantage of using switching lighting method, dual-light switching lighting (DLSL), is that the appearance of defect consists of consistent black and white patterns in the acquired image regardless of the size, shape, orientation of the defect. When single lighting is used, the appearance of a defect is inconsistent and dependent on the direction of the light. The inspected image was divided into small patches and the patches were then clustered with k-mean clustering. A designed threshold is applied to the value generated from sub-optimal filtering (an optimized general-finite impulse-response filter (FIR)). It is a weighted combination of optimized filtering trained by representative patches of all clusters on statistical texture features (mean value, contrast, smoothness, uniformity, third moment, entropy and moment invariants) of the input patch. Sub-optimal filtering outperforms Gabor, wavelet reconstruction, Sobel and differential filtering based on a specially designed performance index proposed by the author. The speed of the algorithm is around 333.33 ms/frame which is sufficient for real-time processing using MATLAB with an Intel Core i7(3.6 GHz).

Gan et al. proposed a template matching method (G-BODI), background-oriented defect inspector (BODI), for rail surface inspection. It uses a random sampling technique to

generate background statistics for the template matching. The approach achieved 100% detection accuracy in the experiment.

A.2.2. Techniques for Uniform Random Texture Surface

Uniform Illumination

Xie et. al [354] proposed a template matching method (X-TM) for colour tonality (chromatic) anomaly (defect) detection of ceramic tile inspection. The normal surfaces have consistent colour shade and tonality. A local common vector (LCV) amongst pixels in the RGB space is computed using the vector directional processing method. An overall colour shade feature vector, which consists of LCV (a local salient chromatic feature), along with local statistics and pixel RGB colours is formed at each pixel. Principal component analysis (PCA) is applied. The first few eigenvectors with the largest eigenvalues are selected to form a multidimensional histogram. The detection was done through the comparison of the preprocessed histograms of both the input image and the reference image which was learnt from the training stage with similarity measure (normalized cross-correlation). Empirical results showed that the approach can achieve a 93.9% detection rate. However, the higher the accuracy comes with an increased computational cost. The approach was claimed to be potentially useful for the other colour textured surfaces such as wood and textiles.

Jahanbin et al. [358] presented a defect detection approach (J-OSVM) for tile surfaces with Gaussian-based One-class SVM. The input image is decomposed with wavelet frames into multiple low resolutions subbands images in various orientations and scales. Second-order statistics, Haralick Features, are extracted from GLCM for each subband. The one-class SVM is trained to learn the decision boundary of defect-free and defective samples with Gaussian kernel. Two evaluation measures, “false rejection rate” (FRR) and “false acceptance rate” (FAR), were used. The inspection results achieved 0.06% and 12.5% of FRR and FAR respectively.

Chen et al. [385] presented a crack detection method (C-SOMO) for bridge inspection. The technique adopts Self Organizing Map Optimization (SOMO), fuzzy logic control, hyper-rectangular composite neural networks (HRCNNs) and image processing techniques. The method achieved 89% for crack recognition and 91% for non-crack recognition.

Dawood et al. [386] presented a hybrid spalling (concrete defect) region detection algorithm and a regression-based spalling depth prediction method (D-SDDP) for subway network inspection. The hybrid algorithm was applied to the input image and it generates a 3D scene presentation on the segmented estimated spalling region. The hybrid algorithm involves image smoothing, thresholding for object of interest (OOI) segmentation, histogram equalization, gaussian blurring, colour transformation and filtering, image scaling and edge detection. The hybrid achieved 89.3% accuracy, 91.7% recall, 94.8% precision. Moreover, three regression model (linear, quadratic and cubic) was proposed as the spalling depth (regresssand) predictors with the regressor of pixel intensity in the 3D scene. The experiment showed that the cubic regression model performs the best.

Ouma et al. [387] presented a fuzzy and wavelet transform-based pothole detection approach (O-FWT) for asphalt pavement inspection. The approach applies a multiscale texture-based image filtering for textons representation using wavelet transform. An adaptive soft-thresholding is applied on the test image to detect candidate defective pixels. A superpixel clustering approach using fuzzy c-means (FCM) algorithm generates the estimated pothole defects and defect-free areas. Fine segmentation is being done by applying morphological operations. The experiments showed that the approach achieved 87.5% average Dice coefficient of similarity and 8.5% mean and 4.9% standard deviations for the errors in the shape and size detections between the ground-truth and the detected potholes.

Non-uniform Illumination

Guan and Gao [369] presented a defect segmentation (detection) approach (G-S) for fabric surface inspection using a visual attention mechanism. Wavelet decomposition was initially applied on the input image to have the feature maps. Centre-surround differences

operation was applied to have feature sub-maps. A saliency map is formed from fusing the feature differences sub-maps. The defect segmentation is derived by applying the maximum between-cluster variance approach and region growing approach. Empirical results showed that the approach is strong in resisting noise and the defect information can be completely segmented out. The detection rates are 97%, 99% and 96% for directional, localized and disseminated defects comparing with 89%, 92% and 80% of wavelet transform algorithm.

Chen et al. [390] proposed a deep learning-based crack detection method (C-NB-CNN) for inspection of nuclear power plant component using convolutional neural network and Naïve Bayes data fusion. The proposed methods achieved a 98.3% hit rate against 0.1 false positive per frame.

Martelli et al. [393] presented a deep learning approach (M-DL) for inspection of very small ducts and cavities with CNN. The approach achieved 91.82% average accuracy in the experiment and 1.229% miss rate in the real scenario.

Cao et al. [402] proposed a concrete crack detection method (C-FCN) using deep full convolutional networks (FCN). The approach achieved 90% in both maximum F1 score (*Max F – measure*) and average precision (*AP*) where *Max F1* and *AP* are defined in (A.7) and (A.8).

$$Max F1 = \operatorname{argmax} F - measure(t) \quad (A.7)$$

$$AP = \frac{1}{2} \sum_{r=0}^1 \max_{r': r' \geq r} Precision(r') \quad (A.8)$$

where

- t is a classification threshold,
- $F - measure(t)$ is the *F – measure* (defined in eqn. (A.2)) when t is used,
- and
- $Precision(r')$ is the *Precision* (defined in eqn. (A.5)) when recall (defined in eqn. (A.6)) r' is used.

Cao et al. [404] proposed three deep learning methods (C-DL) for early detection of fatigue cracks in gusset plate joins in steel bridges. Data augmentation is used to boost the performance of the three methods. The method with transfer learning and fine-tuning of top layers achieved the best state-of-the-art accuracy of 98% in a two-class (crack and non-crack) classification.

A.2.3. Techniques for Patterned Textured Surface

Uniform Illumination

Ngan et al. [355], [356] presented a generalized motif-based defect detection approach (N-M) for Patterned textured plain surface item inspection. The approach was claimed to be effective in detecting defects in 16 out of 17 wallpaper groups other than wallpaper group p1 which consists of only one lattice which repeating itself in 2D patterned texture. The 16 wallpaper groups can be transformed into 3 wallpaper groups, p2, pmm and p4m, with geometry transformation. The approach pursued the comparison of energy-based histograms. The histograms were extracted from lattices level with its constituents, motifs, and consists of the symmetry information of motifs and the relations between motifs and lattice of input lattice and reference lattice with norm metric measurement. The energy promotes defect detection by amplifying the defect information. The decision boundaries were determined from the energy-variance space. The theoretical foundation is provided. The approach achieved an average overall defect detection success rate of 93.32% for fabric which belonged to 1 of 3 wallpaper groups, p2, pmm and p4m. It also achieved an overall defect detection success rate of 93.86% in an evaluation experiment with samples (fabrics, wallpaper, ceramics, ornament, tile, etc.) of the 16 wallpaper groups in another manuscript.

Asha et al. [361] proposed a defect detection algorithm (A-WHC) for automatic inspection of fabric surface which belongs to 1 of 3 wallpaper groups, pmm, p2 and p4m. The algorithm utilizes the periodic property. In this algorithm, the Gabor wavelet transformation in multi-scales and multi-orientations were used. A resultant image with the value of L_2 -norm of all values from the transformation were split into several periodic

blocks. The energy of each block is extracted to form the feature for the defect detection. Ward's hierarchical clustering is used to derive the defect-free and defective clusters in the algorithm. The average performance measures (accuracy, precision and recall) of the approach for pmm, p2 and p4m groups were (93.9%, 100% and 68.6%), (99.6%, 100% and 90.2%), and (99.0%, 100% and 78.9%) respectively.

Asha et al. [365] further presented another defect detection approach (A-WHC-2) for the same inspection problem after his previous work (A-WHC) [361]. The algorithm also utilises the periodic properties of pattern. In the algorithm, Jensen-Shannon Divergence, a symmetrized and smoothed version of the Kullback-Leibler Divergence are used as a dissimilarity measure of different periodic blocks. A matrix of true-metrics is obtained from a dissimilarity matrix of all periodic blocks and is used with Ward's hierarchical clustering for the defect detection without any training sets. Empirical results showed that the approach can achieve better results than the previously proposed algorithm with an average accuracy, precision and recall of 98.8%, 100% and 83%. The accuracies and recalls are 97.9%, 99.3% and 99.3% and 86.4%, 80.0% and 82.6% for pmm, p2 and p4m respectively.

Raheja et al. [349] proposed a real-time automated fabric defect detection approach (R-SW) using texture information. Sliding window based textural features, energy values, were extracted from GLCM and were compared with the reference features. The window contained the deviation of textural energy features beyond a certain threshold was reported as defects and the processing time for a $4cm^2$ image frame was 0.5 seconds.

Sankari et al. [368] proposed a defect identification approach (S-FL) for fabric surface inspection with fuzzy logic and textural features. The textural features include autocorrelation, contrast, cluster shade, dissimilarity, energy, entropy, homogeneity, sum average and cluster prominence extracted from GLCM. The method was tested for detecting fabric hole defects.

Park et al. [377] presented a random forest-based anomaly detection approach (P-RF) for display panel module inspection. The approach aims to detect both real (scratches and pits) and fake (stains, long dust and circular dust). A neighbouring different filter (NDF) which generates an image which

1. is invariant to the light and the contrast of background and foreground
2. does not contain information of repeating patterns to separate the defective foreground

was used to process the input image. Texture, geometrical, statistical and intensity features extracted from the NDF-processed image were used as the input of the random forest classifier. The approach achieved mean accuracy, recall and precision of 92%, 84.67% and 83.67% and 0.99 ratio of correctly classified pixels (RCCP) per image.

Jin et al. [381] proposed a rule-based approach (J-PW) for solder ball grid array (BGA) inspection on PCBs. Otsu's method and an adaptive threshold were used to segment the solder balls. Path walking and Hough transform were used to determine the locations and orientations of the solder balls. Solder ball area, the centre of gravity and compactness of the solder balls are used as features. A set of rules are used for defect detection during the path walking process. The approach achieved 97.89% accuracy.

Bai et al. [389] proposed a thresholding defect detection approach (B-T) for solder ball grid array (BGA) inspection on PCBs. A distance transform-based circle detection algorithm was used to approximately locate the solder balls. Fine location of solder balls was estimated using a circle fitting optimization and hence the fine location of BGA could be determined. Defect detection on a real solder ball is carried out through thresholding on a function of two ratio pairs ($\frac{S_3}{S_1}$ and $\frac{S_3}{S_2}$), the ratios between overlapping area (S_3) of the real ball and a model ball and the areas (S_1 and S_2) of the real ball and the model ball respectively.

Essa et al. [401] proposed a defect detection approach (E-MR-ELM) for textile fabric surface inspection with extreme learning machine (ELM) and similarity measure (a kind

of minimum ratio between local neighbourhood sliding window). The approach achieved 90.07% accuracy, 91.29% sensitivity and 99.67% specificity.

Non-uniform Illumination

Rebhi et al. [347] presented a defect detection approach (R-T) for textured surface inspection with local homogeneity and discrete cosine transform (DCT). Different types of energy-based features extracted from DCT of a homogeneity image (H-image) which contains local homogeneity measure of each pixel. Defect blocks can be determined with thresholding on the Hotelling's T-squared statistic which is a function of the feature energies. Empirical results showed that the approach can achieve an average sensitivity, specificity and detection success rate (accuracy) of 96.55%, 95.98% and 96.9% respectively.

Wang et al. [403] proposed a novel damage defect classification approach (W-R-CNN) using Faster R-CNN model based on ResNet101 framework for classification of two damage categories, efflorescence and spalling damage, on historic masonry structures. The input is an orthoimage belong to one of the damage categories. The approach achieved 99.9% and 90% average precisions for the two damage categories, respectively, with a 95% mAP (mean average precision) in the best case. The approach can work in real-time.

A.2.4. Techniques for Non-uniform and Non-complex Surface

Uniform Illumination

Li et al. [395] proposed a corrugation detection approach (L-SVM) for rail inspection with support vector machine (SVM) and local frequency features. The approach achieved 98.47% and 96.50% in precision and recall respectively.

Non-uniform Illumination

Zheng et al. [352] proposed a surface defect detection approach (Z-GA) using genetic algorithms for bumpy aluminium inspection. The algorithms automatically detect two structural defects; hole and crack defects, on bumpy aluminium surfaces. It is done through

optimizing the morphological processing parameters such as structuring elements, defect segmentation threshold and noise removal threshold with genetic algorithms. The approach achieved detection accuracies of 91% and 86% for the hole and crack defect detections respectively.

Rizvi et al. [384] proposed a thresholding defect detection approach (R-T-2) for railway track inspection. A binary image is generated through smoothening, local normalization, morphological cleaning and histogram enhancement. The number of connected bright pixels in the binary image is used as a feature for the defect detection with a thresholding classifier.

A.2.5. Techniques for Complex Surface

Uniform Illumination

Ogawa et al. [351] proposed a template matching approach (O-TM) for photomask pattern defect detection. It compared the difference between the input and the references' image intensities and phase differences of the measuring wave recorded by the image sensor. The inspection is done with a special imaging setup; a differential interference imaging optics setup. The defect sensitivities were set at 520nm critical dimension of the photomask.

Wang et al. [348] proposed a pattern-matching (template matching) approach (W-TM) for defect detection of the inspection of products in integrated circuit (IC) industry, such as PCBs, BGAs, and ICs. Thresholding is used on the proposed Partial Information Correlation Coefficient (PICC) for anomaly detection. Using PICC is claimed to have better detection performance than the conventionally used coefficient, Traditional Normalized Cross-Correlation Coefficient (TNCCC). The results of using TNCCC tend to have more false alarms and are unable to tolerate small translation and rotation. Moreover, the results showed that PICC was able to allow at least 1 pixels shifting and 0.2-degree rotation. Experimental results showed that the approach achieves 100% detection rate in inspecting 50 test images at a resolution of 2820×320 and processed an image within 7 seconds.

Kim and Yoo [372] proposed a corner feature point matching method (K-FPM) for defect detection of wafer surface inspection. The method worked for a non-repetitive pattern. It is a method of comparing two feature point sets detected by the proposed corner detector in input and reference Wafer SEM images. Empirical results showed that the method achieved higher detection accuracy and is less sensitive to alignment error and noise than conventional approaches. Conventional approaches include subtraction, Non-Parametric (NP)-subtraction, Normalized Cross-Correlation (NCC)-based approach and eigenvalue-based approach. The proposed approach achieved a sensitivity of 95%, a specificity of 90% and an accuracy of 94% with a 0.26 seconds processing time for an image at a resolution of 300×300 . However, the approach cannot locate the detected defect.

Kong et al. [382] proposed a template-matching method (K-TM) for defect detection on planar surfaces of industrial products. The method automatically selects and align the template image to the test image. The method achieved an average accuracy of 85.95% on two datasets.

Carrera et al. [383] proposed a sparse-representation-based defect detection approach (C-SR) for nanofibrous products inspection with dictionary learning. L_2 -norm of residual and L_1 -norm of coefficient vector of estimated sparse representation are used as features. The approach achieved 92.6% AUC.

Non-uniform Illumination

Singh and Bharti [364] proposed a template matching approach (S-TM) for defect (missing or wrongly placed components) detection of PCB assembly surface inspection. It compared the extracted features using the Connected Component Labeling Operation on binary images generated by Otsu thresholding on test and reference images. The defect is localized using window searching algorithm.

Benedek et al. [366] presented a hierarchical multi-marked point process model (B-HMMPP) for scooping object (SO, defect) detection for PCB assembly solder paste (SP)

inspection. It is a hierarchically joined probabilistic part-based approach using an energy-based minimization algorithm and an extended accelerated Multiple Birth and Death (MBD) dynamic technique. The energies used in the objective function included data and prior energies. The first energy is related to the parent-child hierarchical appearance properties of a proper scoop and the second energy is related to the geometrical constraints of the configuration of multiple scoops. The empirical results showed that the model achieved an SP object-level F-rate (SPOF) of 99.96%, SP area F-rate (SPAF) of 93.90%, SP average orientation error (SPOE) 1.36, SO object-level F-rate (SOOF) of 92.40% and SO area F-rate (SOAF) of 83.04%.

Yuan presented [391] a defects detection and localization method (Y-DCNN) for concrete surface inspection with a deep convolutional neural network which can geolocate the defects without requiring external localization sensors. The method achieved 80.7% and 86% detection and localization accuracies respectively. The method can work in real-time.

Hawari et al. [397] proposed an approach (H-ROI) for detecting 4 defects, cracks, settled deposits, ovality and displaced joints, in sewer pipelines. Each defect is detected by a standalone algorithm. Morphology segmentation, edge detection, contour segmentation, ellipse fitting, ring shape detection are used. The approach achieved 73.7%, 52.9%, 65.3% and 100% positive predictive values (PPV) for the four classes respectively.

Kumar et al. [398] proposed a deep learning approach (K-DL) for detecting four sewer defects, root, intrusions, deposits and crack, with deep convolutional neural networks (CNNs). The approach achieves 86.2% accuracy, 87.7% precision and 90.6% recall in overall respectively.

Myrans et al. [399] proposed a defect detection approach (M-RF) for sewer pipelines inspection with random forests (RF). The video frame image collected by closed-circuit television (CCTV) is temporal smoothed by the Hidden Markov Model (HMM) and windowing to improve the false negative rate by more than 20%. “GIST of scene” (GIST)

feature is used. The approach achieved over 80% detection accuracy and 85% area under the curve (AUC) on a dataset of still images.

A.2.6. General Approaches and Supporting Techniques

D. P. Brzakovic et al. [478] proposed a feature learning approach which learnt the statistical properties of texture randomness and homogeneity of web material (metals, papers, plastics, textiles, etc) especially for material with random texture. The statistical properties are the total number of fibre crossings and the pore area distribution. The model used in the approach was cellular neural networks with parallel configuration. The approach can process an image at a resolution of 512×512 within 5 milliseconds and can potentially be utilized in real-time automated inspection in the manufacturing process.

Franz Pernkopf et al. [479] provided an overview of three different image acquisition approaches for automated defect detection on a metallic surface which can be either uniform surface or non-uniform and non-complex surface. The first approach is the grey-level intensity imaging technique with the illumination of front lighting, backlighting or structured lighting. The second approach is a range imaging technique with laser light sectioning in conjunction and fast imaging sensors which can deliver a thousand sections per second with a lateral resolution of 512 pixels. The third approach is a range imaging technique using photometric stereo. The paper claimed that range imaging techniques may outperform intensity imaging technique if the reflection property across the intact surface changed. However, range imaging approaches are restricted to surface defects with three-dimensional characteristics, e.g. cavities, scratches, nicks, etc. Some resultant pictures are presented in the paper.

Aiger and Talbot [359] presented a simple and fast algorithm (A-PHOT) for defect detection of regular and non-regular textured surface inspection. Phase Only Transform (PHOT) is the Discrete Fourier Transform (DFT) divided by its magnitude. PHOT removes regularities in the images and preserve irregular patterns. The inverse transform followed by the manually selected thresholding from Mahalanobis distance was used for the localization of the defect. The algorithm took advantage of the computational speed of Fast

Fourier Transform (FFT) and the availability of parallel computation. Empirical results showed that the approach worked well for defect detection on various textures.

Ren et al. [388] proposed a general deep-learning-based approach (R-DL) which can also work for AVI with transfer learning. The proposed approach have been demonstrated in AVI on both uniform surface under non-uniform illumination (hot-rolled steel strip inspection) and the uniform random surface under uniform illumination (wood inspection). The approach achieves 99.21% and 94.29% accuracies in patch classification for hot-rolled steel strip and wood inspection, respectively and 10.06%, 26.23% and 11.6% of error escape, false alarm and misclassification rate in defect segmentation for wood inspection.

AVI techniques in the fruits and vegetable industry are not reviewed in this Thesis. The authors are referred to a comprehensive review paper published by Baohua Zhang et al. [480].

There are also comprehensive review papers of AVI techniques for fabric defect detection [474], [481]–[486], texture surfaces inspection [487], steel surface inspection [488], loudspeaker cones surface inspection [80] and semiconductor surface inspection [489], [490].

Appendix B (Appendix for Chapter 3)

This appendix shows the proposed algorithms and functions and additional experimental results in Chapter 3.

Algorithm B.1 PA-QN with Wolfe Conditions

Parameters: Choose $\gamma > 0$; an initial point, x_0 ; an initial Hessian matrix, B_0 ; parameters of first and second Wolfe conditions, m_1 and m_2 ; an initial step size, α'_0 ; a backtracking shrinking factor, τ ; maximums of iterations of the main for loop and backtracking line search, m_n and b_n .

$Prox_{\bar{F}_\gamma}^\gamma(x_{-1}) = 0_n$ # n -dimensional zero vector, n is the dimensional of x

$\eta_{-1} = 1$

for $k = 0, 1, \dots, m_n$ **do**

$Prox_{\bar{F}_\gamma}^\gamma(x_k) = \sum_{i=1}^N Prox_{f_i}^\gamma(x_k)$

$\nabla \bar{F}_\gamma(x) = \frac{1}{\gamma} (Prox_{\bar{F}_\gamma}^\gamma(x) - x)$

$W_C = 0$ # variable which shows whether the Wolfe Conditions have been fulfilled

$x_{k+1} = x_k$

Quasi-Newton method

if $k > 0$

$s_{k-1} = \nabla \bar{F}_\gamma(x_k) - \nabla \bar{F}_\gamma(x_{k-1})$

$B_k = B_{k-1} + \frac{s_{k-1}s_{k-1}^T}{s_{k-1}^T \Delta x_{k-1}} - \frac{B_{k-1} \Delta x_{k-1} (B_{k-1} \Delta x_{k-1})^T}{\Delta x_{k-1}^T B_{k-1} \Delta x_{k-1}}$

$p_{1,k} = -B_k^{-1} \nabla \bar{F}_\gamma(x_k)$ # descent direction

end if

$[x_{k+1}, W_C] = \text{WolfeConditionsCheck}(x_{k+1}, p_{1,k}, \alpha'_0, \tau, b_n, m_1, m_2)$

Steepest gradient descent in L_2 -norm

if $W_C = 0$

$p_{2,k} = \nabla \bar{F}_\gamma(x)$ # descent direction

$[x_{k+1}, W_C] = \text{WolfeConditionsCheck}(x_{k+1}, p_{2,k}, \alpha'_0, \tau, b_n, m_1, m_2)$

end if

Algorithm B.1 PA-QN with Wolfe Conditions (Continue)

```

## Steepest gradient descent in  $L_1$ -norm ##
if  $W_C = 0$ 
     $p_{3,k} = -\text{sign}(\frac{\partial \bar{F}_Y(x)}{\partial x_i})e_i$       # steepest gradient descent for  $L_1$ -norm,
                                                # where  $e_i$  is the  $i^{th}$  standard basis vector
                                                # and  $i$  is the index for which  $\|\nabla \bar{F}_Y(x_k)\|_\infty =$ 
                                                #  $|(\nabla \bar{F}_Y(x_k))_i|$ 
     $[x_{k+1}, W_C] = \text{WolfeConditionsCheck}(x_{k+1}, p_{3,k}, \alpha'_0, \tau, b_n, m_1, m_2)$ 
end if

## Nesterov's accelerated gradient descent ##
if  $W_C = 0$ 
     $\eta_t = \frac{1 + \sqrt{1 + 4\eta_t^2}}{2}$ 
    # descent direction
     $p_{4,k} = \text{Prox}_{\bar{F}_Y}^\gamma(x_k) + \frac{\eta_t - 1}{\eta_{t+1}}(\text{Prox}_{\bar{F}_Y}^\gamma(x_k) - \text{Prox}_{\bar{F}_Y}^\gamma(x_{k-1}))$ 
     $[x_{k+1}, \sim] = \text{WolfeConditionsCheck}(x_{k+1}, p_{4,k}, \alpha'_0, \tau, b_n, m_1, m_2)$ 
end if
end for

```

Function B.1 Backtracking Inexact Line Search with Wolfe Conditions

```

Function WolfeConditionsCheck( $x_{k+1}, p_k, \alpha'_0, \tau, b_n, m_1, m_2$ )
 $W_C = 0; \alpha'_k = \alpha'_0$ 
for  $j = 0, 1, \dots$  do
     $\hat{x}_{k+1} = x_k + \alpha'_k p_k$ 
    if  $F(\hat{x}_{k+1}) \leq F(x_k) + m_1 \alpha'_k p_k^T \nabla \bar{F}_Y(x_k)$  and
         $-p_k^T \nabla \bar{F}_Y(\hat{x}_{k+1}) \leq -m_2 p_k^T \nabla \bar{F}_Y(x_k)$ 
         $x_{k+1} = \hat{x}_{k+1}$ 
         $W_C = 1$ 
        break
    else if  $j > b_n$ 
        break
    else
         $\alpha'_k = \alpha'_k * \tau$ 
    end if
end for
Return  $x_{k+1}, W_C$ 

```

Algorithm B.2 Greedy PA-QN

Parameters: Choose $\gamma > 0$, an initial point, x_0 ; an initial Hessian matrix, B_0 ; an initial step size, α'_0 ; a backtracking shrinking factor, τ ; maximums of iterations of the main for loop and backtracking line search, m_n and b_n .

```

 $Prox_{\bar{F}_\gamma}^\gamma(x_{-1}) = 0_n$  #  $n$ -dimensional zero vector,  $n$  is the dimensional of  $x$ 
 $\eta_{-1} = 1$ 
 $F_0 = F(x_0)$ 
for  $k = 0, 1, \dots, m_n$  do
     $Prox_{\bar{F}_\gamma}^\gamma(x_k) = \sum_{i=1}^N Prox_{f_i}^\gamma(x_k)$ 
     $\nabla \bar{F}_\gamma(x) = \frac{1}{\gamma} (Prox_{\bar{F}_\gamma}^\gamma(x) - x)$ 
     $x_{k+1} = x_k$ 
     $F_{k+1} = F_k$ 

    ## Quasi-Newton method ##
    if  $k > 0$ 
         $s_{k-1} = \nabla \bar{F}_\gamma(x_k) - \nabla \bar{F}_\gamma(x_{k-1})$ 
         $B_k = B_{k-1} + \frac{s_{k-1}s_{k-1}^T}{s_{k-1}^T \Delta x_{k-1}} - \frac{B_{k-1} \Delta x_{k-1} (B_{k-1} \Delta x_{k-1})^T}{\Delta x_{k-1}^T B_{k-1} \Delta x_{k-1}}$ 
         $p_{1,k} = -B_k^{-1} \nabla \bar{F}_\gamma(x_k)$  # descent direction
    end if
     $[x_{k+1}, F_{k+1}] = \text{Backtracking}(x_{k+1}, F_{k+1}, p_{1,k}, \alpha'_0, \tau, b_n)$ 

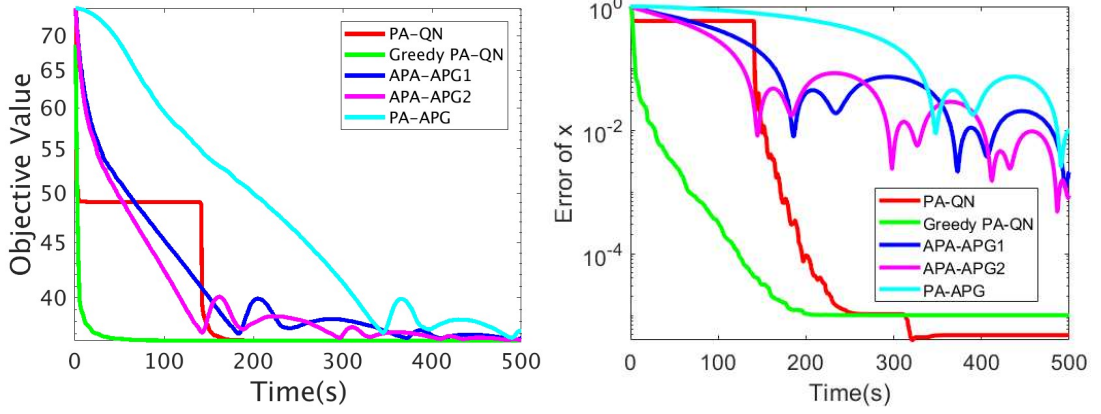
    ## Steepest gradient descent in  $L_2$ -norm ##
     $p_{2,k} = -\nabla \bar{F}_\gamma(x)$  # descent direction
     $[x_{k+1}, F_{k+1}] = \text{Backtracking}(x_{k+1}, F_{k+1}, p_{2,k}, \alpha'_0, \tau, b_n)$ 

    ## Steepest gradient descent in  $L_1$ -norm ##
     $p_{3,k} = -\text{sign}(\frac{\partial \bar{F}_\gamma(x)}{\partial x_i}) e_i$  # steepest gradient descent in  $L_1$ -norm, where  $e_i$ 
    # is the  $i^{th}$  standard basis vector and  $i$  is the
    # index for which  $\|\nabla \bar{F}_\gamma(x_k)\|_\infty = |(\nabla \bar{F}_\gamma(x_k))_i|$ 
     $[x_{k+1}, F_{k+1}] = \text{Backtracking}(x_{k+1}, F_{k+1}, p_{3,k}, \alpha'_0, \tau, b_n)$ 

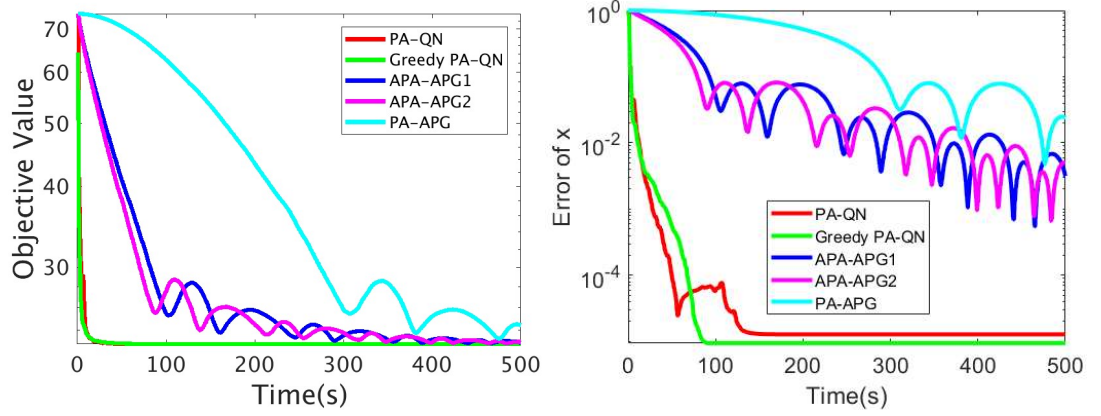
    ## Nesterov's accelerated gradient descent ##
     $\eta_k = \frac{1 + \sqrt{1 + 4\eta_k^2}}{2}$ 
    # descent direction
     $p_{4,k} = Prox_{\bar{F}_\gamma}^\gamma(x_k) + \frac{\eta_k - 1}{\eta_{k+1}} (Prox_{\bar{F}_\gamma}^\gamma(x_k) - Prox_{\bar{F}_\gamma}^\gamma(x_{k-1}))$ 
     $[x_{k+1}, \sim] = \text{Backtracking}(x_{k+1}, F_{k+1}, p_{4,k}, \alpha'_0, \tau, b_n)$ 
end for

```

When $(m, n, k_1, k_2) = (50, 100, 5, 3)$



When $(m, n, k_1, k_2) = (100, 100, 5, 5)$



When $(m, n, k_1, k_2) = (1000, 100, 5, 50)$

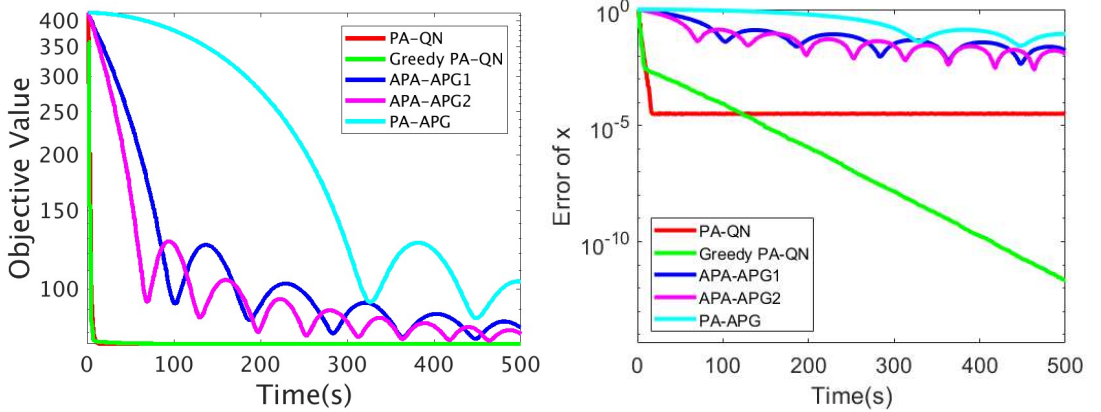


Figure B.1 Objective function value $F(x_k)$ and error vs time t (up to 500 seconds) on the sum of multiple sparsities with $\alpha_1 = 0.5$ and $\alpha_2 = 0.5$

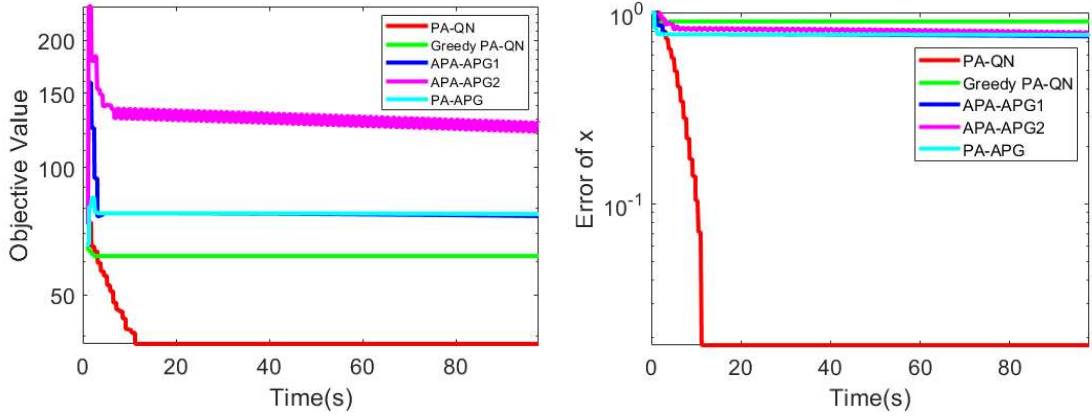
Table B.1 Objective function value $F(x_k)$ versus time t (seconds) on sum of multiple sparsities with $\alpha_1 = 0.5$ and $\alpha_2 = 0.5$

Cases (m, n, k_1, k_2)	(50, 100, 5, 3)	(100, 100, 5, 5)	(1000, 100, 5, 50)
$F(x_k)$	70 60 50	50 40 30	200 150 100
PA-QN	1 2 3	2 2 4	4 4 6
Greedy PA-QN	1 2 3	2 2 3	2 4 4
APA-APG1	3 15 60	31 52 82	62 78 96
APA-APG2	3 12 47	27 45 70	43 53 65
PA-APG	45 96 200	167 218 274	254 285 317

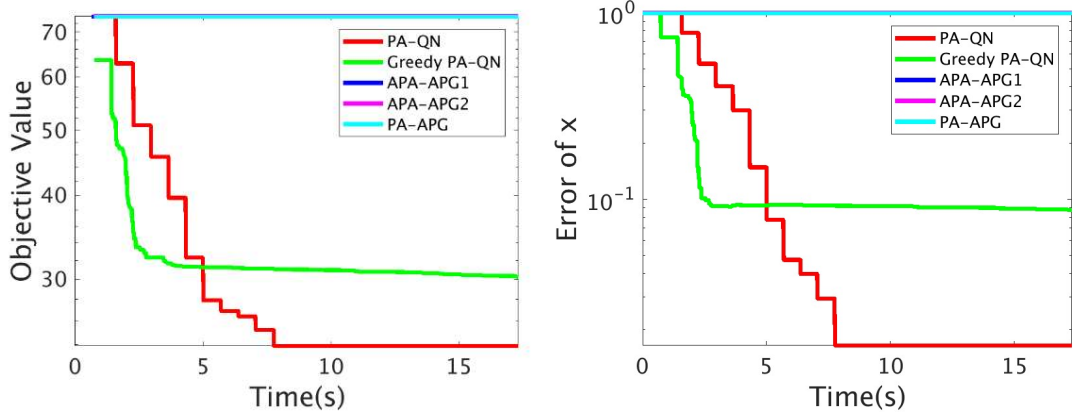
Table B.2 Precision ϵ_p versus time t (seconds) on sum of multiple sparsities with $\alpha_1 = 0.5$ and $\alpha_2 = 0.5$

Cases (m, n, k_1, k_2)	(50, 100, 5, 3)	(100, 100, 5, 5)	(1000, 100, 5, 50)
ϵ_p	10^{-3} 10^{-4} 10^{-5}	10^{-3} 10^{-4} 10^{-5}	10^{-5} 10^{-8} 10^{-14}
PA-QN	171 196 313	28 50 -	114 187 -
Greedy PA-QN	66 123 -	43 67 87	2 136 473
APA-APG1	515 602 -	440 601 -	3 285 -
APA-APG2	485 560 -	399 592 -	887 - -
PA-APG	561 - -	625 685 -	97 99 -

When $(m, n, k_1, k_2) = (50, 100, 5, 3)$



When $(m, n, k_1, k_2) = (100, 100, 5, 5)$



When $(m, n, k_1, k_2) = (1000, 100, 5, 50)$

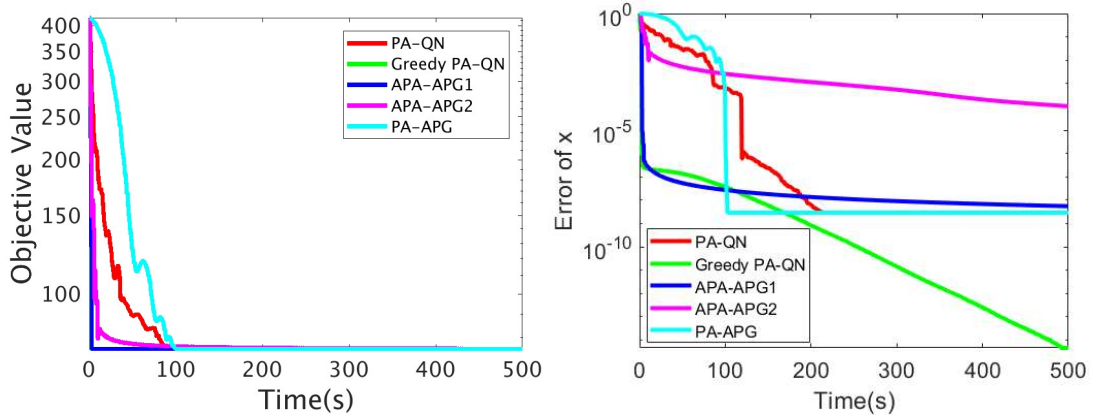


Figure B.2 Objective function value and error vs time (up to 500 seconds) on the sum of multiple sparsities with $\alpha_1 = \frac{1}{1+10^5}$ and $\alpha_2 = \frac{10^5}{1+10^5}$

Table B.3 Objective function value $F(x_k)$ versus time (seconds) on the sum of multiple sparsities with $\alpha_1 = \frac{1}{1+10^5}$ and $\alpha_2 = \frac{10^5}{1+10^5}$

Cases (n, d, k, m)	(50, 100, 5, 3)	(100, 100, 5, 5)	(1000, 100, 5, 50)
$F(x_k)$	100 70 50	35 30 25	200 150 100
PA-QN	1 2 7	4 5 8	10 17 34
Greedy PA-QN	1 1 -	2 21 -	1 1 2
APA-APG1	1 279 -	- - -	3 3 3
APA-APG2	1 - -	- - -	3 3 6
PA-APG	1 1 -	- - -	39 45 70

Table B.4 Precision ϵ_p versus time (seconds) on the sum of multiple sparsities with $\alpha_1 = \frac{1}{1+10^5}$ and $\alpha_2 = \frac{10^5}{1+10^5}$

Cases (n, d, k, m)	(50, 100, 5, 3)	(100, 100, 5, 5)	(1000, 100, 5, 50)
ϵ_p	0.9 0.3 0.02	0.1 0.09 0.02	1e-4 1e-7 1e-13
PA-QN	2 7 11	5 5 8	112 190 -
Greedy PA-QN	1 - -	3 14 -	2 126 414
APA-APG1	1 - -	- - -	4 445 -
APA-APG2	3 609 -	- - -	907 - -
PA-APG	1 - -	- - -	97 99 -

Appendix C (Appendix for Chapter 4)

This appendix shows the tools, lemmas and the proofs of those lemmas which are used in Chapter 4.

Section C.1 introduces the tools for probability analyses. Section C.2 presents some supporting technical lemmas. Section C.3 presents proof of each lemma.

C.1. Tools for Probability Analyses

All probabilistic analysis in Chapter 4 depends on the measure concentration results for Lipschitz functions of random matrices, especially in their norms and singular values. All the following probabilistic bounds are used to bound the norms in eqn. (4.21).

C.1.1. Preliminaries

In this subsection, the preliminary probability bounds which related to a random matrix and the definition of the preliminary singular values of a random matrix which will be used in probability analyses are presented.

Let M be an $am \times bm$, $a > b$, random matrix with entries i.i.d. $N\left(0, \frac{1}{m}\right)$. Then (see [491] and the references therein) the following probabilistic bounds in eqns (C.1) and (C.2) are valid.

$$\mathbb{P}[\|M\| \geq \sqrt{a} + \sqrt{b} + \sqrt{2\varepsilon}] < \exp(-\varepsilon m) \quad (\text{C.1})$$

$$\mathbb{P}[\sigma_{\min}(M) \geq \sqrt{a} - \sqrt{b} - \sqrt{2\varepsilon}] < \exp(-\varepsilon m) \quad (\text{C.2})$$

Similarly (e.g., via [[492], eq. (2.35)])

$$\mathbb{P}[\|M\|_F \geq \sqrt{abm} + \sqrt{2\varepsilon}] < \exp(-\varepsilon m) \quad (\text{C.3})$$

The extremal submatrix singular values¹⁰ which are shown in eqns (C.4)-(C.6) are used to study the action of random matrices on sparse vectors.

$$\beta_s(M) = \sup_{\|x\|_2 \leq 1, \|x\|_0 \leq s} \|Mx\|_2 \quad (C.4)$$

$$\gamma_s(M) = \inf_{\|x\|_2 = 1, \|x\|_0 \leq s} \|Mx\|_2 \quad (C.5)$$

$$\emptyset_s(M) = \sup_{\substack{\|x\|_2 \leq 1, \|x\|_0 \leq s \\ \|y\|_2 \leq 1, \|y\|_0 \leq s}} x^* M y \quad (C.6)$$

Let $a > 0$ be arbitrary and suppose that $c < \min(a, b)$. Since $\binom{m}{k} < \left(\frac{me}{k}\right)^k$, the $am \times bm$ matrix M has $\binom{bm}{cm} \leq \exp\left(m c \log \frac{b}{c}\right)$ submatrices $M_{[am], I}$ of size $am \times cm$. The union bounds in eqn. (C.7) is formed.

$$\mathbb{P} \left[\beta_{cm}(M) \geq \sqrt{a} + \sqrt{c} + \sqrt{2c \log \left(\frac{b}{c}\right) + 2\varepsilon} \right] < \exp(-\varepsilon m) \quad (C.7)$$

Note that due to the rotational invariance of the Gaussian distribution, if S is a (possibly random) subspace of dimension $a'm < am$ and it is probabilistically independent of M , the above bounds hold for $P_S M$, with replacing a by a' .

C.1.2. Observations on Submatrix Singular Values

In this subsection, four lemmas, **Lemmas 5 - 8**, which shows the bound relationships between singular values of a random matrix and its submatrices and a corresponding proof are presented. The proof of **Lemma 5** is shown in Section C.3.4 and **Lemmas 6 - 8** are borrowed from **Lemmas 7 - 9** in [4].

Lemma 5

For any block matrix $[M \tilde{N}_1 \dots \tilde{N}_n]$, scalar t_i with $|t_i| \leq 1, \forall i = 1 \dots n$ and $\tilde{t} = \min(1, t_1, \dots, t_n)$, the following bound in eqn. (C.8) can be formed.

¹⁰ The quantities $\beta, \gamma, \emptyset$ have a strong relationship to the restricted isometry property of the matrices in question. In particular, argument suggests the possibility of a deterministic analogue of **Theorem 1** under RIP-like conditions on the matrix Z

$$\gamma_s([M|t_1\tilde{N}_1| \dots |t_n\tilde{N}_n]) \geq |\tilde{t}|\gamma_s([M \tilde{N}_1 \dots \tilde{N}_n]) \quad (C.8)$$

where function γ_s is defined in eqn. (C.5).

Lemma 6

For any matrices $M, \tilde{N}$, the following bounds in eqns (C.9) and (C.10) can be formed.

$$\gamma_s(M + \tilde{N}) \geq \gamma_s(M) - \beta_s(\tilde{N}) \quad (C.9)$$

and

$$\beta_s(M + \tilde{N}) \leq \beta_s(M) + \beta_s(\tilde{N}) \quad (C.10)$$

where functions γ_s and β_s are defined in eqns (C.5) and (C.4) respectively.

Lemma 7

Suppose $\tilde{W} \in \mathbb{R}^{m \times n}$, ($m < n$) be any matrix with full row rank m , and let $P_{\tilde{W}} \in \mathbb{R}^{n \times n}$ denotes the projection operator onto the range of $\tilde{W}^*$. The following bound in eqn. (C.11) can be formed.

$$\beta_s(P_{\tilde{W}}) \leq \frac{\beta_s(\tilde{W})}{\sigma_{\min}(\tilde{W})} \quad (C.11)$$

where β_s is defined in eqn. (C.4).

Lemma 8

For any matrix M , the following bound in eqn. (C.12) can be formed.

$$\gamma_s^2([IM]) \geq \min\{1, \gamma_s^2(M)\} - \Phi_s(M) \quad (C.12)$$

where

γ_s and Φ_s are defined in eqns (C.5) and (C.6), and

I is defined in Section 4.1.

C.1.3. Two Concentration Results

In this subsection, two lemmas, **Lemmas 9 and 10**, which show two concentration results respectively and corresponding proofs are presented. Sections C.3.5 and C.3.6 show the proofs of **Lemmas 9 and 10** respectively.

Lemma 9

Let $\vartheta = \frac{\hat{\mu}_{J^c}}{N\sqrt{1-\rho}}$ where $\hat{\mu}_{J^c} = [\mu_{J^c}^1, \dots, \mu_{J^c}^N]$, $\mu_i \in \mathbb{R}^m$, $\|\mu_i\|_\infty \leq C_\mu m^{-\frac{1}{2}}$, $\|\mu_i\| = 1$, $i = 1, \dots, N$, and let $J \subset [m]$ be uniformly distributed among all subsets of size pm . Then, for any $\varepsilon > 0$ and $\forall i$, the bound in eqn. (C.13) holds with probability at least $1 - \exp(-\varepsilon m)$ in J

$$\|\mu_{J^c}^i\|^2 > 1 - \rho - 8C_\mu^2\sqrt{2\varepsilon} \quad (\text{C.13})$$

where

μ_i , C_μ , m and N are defined in **Assumption 1**,

$\mu_{J^c}^i$ is defined in eqn. (C.55), and

δ and ρ are defined in eqn. (4.7)

Furthermore, if $\delta \geq 1$, then with probability the most $N \exp(-2\varepsilon^* m)$

$$\|\vartheta\|^2 \leq \frac{1}{4} \quad (\text{C.14})$$

where

N is defined in **Assumption 1**, and

ε^* is the short forms of $\varepsilon_{\delta, \rho, v_{\max}}^*$ and is defined in eqn. (4.15).

Lemma 10

Suppose any $\delta \geq 1$ and $\rho < 1$ in where δ and ρ are defined in eqn. (4.7). Let $v_{\max} \leq v^*$, and ε^* , m^* , α^* are following (4.15), and $m \geq m^*$ in where

$v_{\max}$ is defined in eqn. (4.9)

m is defined in **Assumption 1**,

v^* , m^* , ε^* and α^* are the short forms of $v_{\delta, \rho}^*$, $m_{\delta, \rho, v_{\max}, v_{\min}}^*$, $\varepsilon_{\delta, \rho, v_{\max}}^*$ and $\alpha_{\delta, \rho, v_{\max}}^*$ and are defined in eqn. (4.15).

With probability at least $1 - (2 + N)\exp(-2\varepsilon^* m)$, the matrix W defined by eqn. (C.15) satisfies eqn. (C.16) in where

N is defined in **Assumption 1**,

$\hat{\mu}_{J^c}$ is a matrix equal to $[\mu_{J^c}^1, \dots, \mu_{J^c}^N]$, and

μ_{jc}^i is defined in **Lemma 1**.

$$W^* = \begin{bmatrix} \frac{Z_{jc,I}}{\sqrt{1-\rho}} & \frac{\hat{\mu}_{jc}}{N\sqrt{1-\rho}} \end{bmatrix} \in \mathbb{R}^{(m-k_2) \times (k_1+N)} \quad (C.15)$$

$$\sigma_{\min}(W) \geq \frac{1}{4} \quad (C.16)$$

where $Z_{jc,I}$ is defined in **Lemma 1**.

C.1.4. Concentration of Measure

This subsection shows some probability bounds which will be used in the probability analyses when C_s defined in eqn. (4.10) is used. The proofs of them except eqns. (C.19) and (C.20) which have been proven in (iii) and (iv) of Section B.b in [4] are also presented.

$$\|U\| \leq 3\nu_{\max}\sqrt{\delta} \quad (C.17)$$

with probability (w.p.) $\geq 1 - \exp(-2\varepsilon^*m)$

$$\sigma_{\min}(U) \geq \frac{\nu_{\min}\sqrt{\delta}}{2} \quad (C.18)$$

w.p. $\geq 1 - \exp(-2\varepsilon^*m)$

$$\gamma_s(\hat{U}^*) \geq \sqrt{\frac{(1-\rho)}{2}} \quad (C.19)$$

w.p. $\geq 1 - \exp(-2\varepsilon^*m)$

$$\phi_s(\hat{U}^*) \leq \frac{(1-\rho)}{4} \quad (C.20)$$

w.p. $\geq 1 - \exp(-2\varepsilon^*m)$

$$\beta_s(P_W \hat{U}^*) \leq \frac{\sqrt{1-\rho}}{8} \quad (C.21)$$

w.p. $\geq 1 - \exp(-2\varepsilon^*m)$

$$\beta_s(P_W) \leq \frac{\sqrt{1-\rho}}{8} \quad (C.22)$$

w.p. $\geq 1 - (3 + N)\exp(-2\varepsilon^*m)$

where

U	is defined in eqn. (C.53),
$\nu_{\max}$	is defined in eqn. (4.9),
δ and ρ	are defined in Assumption 3 ,
ε^*	is the short form of $\varepsilon_{\delta,\rho,\nu_{\max}}^*$ and is defined in eqn. (4.15),

m and N	are defined in Assumption 1 ,
γ_s , $\emptyset_s$ and β_s	are defined in eqns (C.4)-(C.6),
s	is defined in eqn. (4.23), and
P_W	is defined in eqns (C.63)-(C.65).

Before the proofs of eqns (C.17), (C.18), (C.21) and (C.22) are shown, some bounds of the constants which will be used in the proofs are derived as follows. The proof is immediately after the derivation.

Note that since $Y[t] \leq t$, the choice of C_s in eqn. (4.10) and the fact that $C_\mu \geq 1$ (defined in **Assumption 1**), $\delta \geq 1$ (assumption stated in **Lemma 3**) imply that

$$\sqrt{C_s} \leq C_\mu \sqrt{C_s} < 2^{-8}(1 - \rho) \quad (\text{C.23})$$

Moreover, using the fact that $C_\mu \geq 1$, $\delta \geq 1$ and eqn. (4.14), the following inequality in eqn. (C.24) holds

$$C_s \log\left(\frac{\delta}{C_s}\right) \leq 2^{-17} C_\mu^{-4} \delta^{-1} (1 - \rho)^2 \leq 2^{-17} (1 - \rho)^2 \quad (\text{C.24})$$

and hence eqn. (C.25) holds too.

$$\sqrt{2C_s \log\left(\frac{\delta}{C_s}\right)} \leq 2^{-8}(1 - \rho) \quad (\text{C.25})$$

There are three other bounds of interest are noted. First, the following bound is got by using

the constant $m^* = \max\left(\frac{2^{17}}{(1-\rho)C_s}, \frac{2^{16} \times 3}{\lambda^8 v_{\min}^8 \delta^2 (1-\rho)^3}\right)$ stated in eqn. (4.15) and eqn. (C.23),

$$m^* \geq 2^{33}(1 - \rho)^{-3} > 2^{33}(1 - \rho)^{-2} \quad (\text{C.26})$$

where m^* is a short form of $m_{\delta, \rho, v_{\max}, v_{\min}}^*$ in eqn. (4.15).

Moreover, the following bound is got from eqn. (4.15) and (C.23)

$$v^* \leq 2^{-12} \sqrt{\frac{C_s}{\delta}} \leq 2^{-12} \sqrt{C_s} \leq 2^{-20}(1 - \rho) \quad (\text{C.27})$$

From eqn. (4.15), $\varepsilon^* < \nu^2$, $\varepsilon^* < 2^{-4} (1 - \rho)^2$ can be derived. Furthermore, $\alpha^* = Y \left[\frac{\varepsilon^*}{\delta} \right] \leq \frac{\varepsilon^*}{\delta} \leq \varepsilon^*$ can be got from (4.15). As the result, $\alpha^* \leq 2^{-40} (1 - \rho)^2$, and for any $m \geq m^*$, eqn. (C.28) holds.

$$\begin{aligned} \sqrt{\alpha^* + \frac{1}{m}} &\leq \sqrt{\alpha^*} + \sqrt{\frac{1}{m^*}} \\ &\leq 2^{-20} (1 - \rho) + 2^{-\frac{33}{2}} (1 - \rho) \\ &\leq 2^{-8} (1 - \rho) \end{aligned} \tag{C.28}$$

where

α^* is a short form of $\alpha_{\delta, \rho, \nu_{max}}^*$ in eqn. (4.15), and

m is defined in **Assumption 1**.

Finally, the following weak bound in eqn. (C.29) is noted

$$\sqrt{4\varepsilon^*} < 2^{-8} (1 - \rho) \tag{C.29}$$

(i) Proof of eqn. (C.17):

The following bound can be easily derived

$$\begin{aligned} \|U\| &\leq \left\| \begin{bmatrix} v_{max}(Z_{jc}^1)^* \\ \vdots \\ v_{max}(Z_{jc}^N)^* \end{bmatrix} \right\| \\ &\leq v_{max} \|\hat{U}\| \end{aligned} \tag{C.30}$$

where

U is defined in eqn. (C.53),

v_{max} is defined in eqn. (4.9),

Z_{jc}^i is defined in **Lemma 1**, and

$\hat{U}$ is an $(n - k_1) \times (m - k_2)$ i.i.d. $N\left(0, \frac{1}{m}\right)$ matrix in where m and n are defined in **Assumption 1**.

Notice that $\hat{U}$ is an $(n - k_1) \times (m - k_2)$ i.i.d. $N\left(0, \frac{1}{m}\right)$ matrix. Applying eqn. (C.1), eqn. (C.31) holds w.p. at least $1 - \exp(-2\varepsilon^* m)$.

$$\|U\| \leq v_{\max}(\sqrt{\delta - \alpha^*} + \sqrt{1 - \rho} + \sqrt{4\varepsilon^*}) \quad (\text{C.31})$$

where

δ and ρ are defined in **Assumption 3**,
 ε^* and α^* are the short forms of $\varepsilon_{\delta, \rho, v_{\max}}^*$ and $\alpha_{\delta, \rho, v_{\max}}^*$ and are defined in eqn. (4.15).

Since $\sqrt{1 - \rho} < 1 \leq \sqrt{\delta}$ and [applying (C.29)] $\sqrt{4\varepsilon^*} \leq 2^{-8}(1 - \rho) \ll 1 \leq \sqrt{\delta}$, the good event in eqn. (C.32) holds w.p. at least $1 - \exp(-2\varepsilon^*m)$.

$$\|U\| \leq 3v_{\max}\sqrt{\delta} \quad (\text{C.32})$$

Eqn. (C.17) is proven.

(ii) Proof of eqn. (C.18):

Since the following inequality in eqn. (C.33) holds,

$$\begin{aligned} \sigma_{\min}(U) &\geq \sigma_{\min}\left(\begin{bmatrix} v_{\min}(Z_{jc}^1)^* \\ \vdots \\ v_{\min}(Z_{jc}^N)^* \end{bmatrix}\right) \\ &\geq \sigma_{\min}(v_{\min}\hat{U}) \\ &\geq v_{\min}\sigma_{\min}(\hat{U}) \end{aligned} \quad (\text{C.33})$$

where

U is defined in eqn. (C.53),

$v_{\min}$ is defined in eqn. (4.15),

Z_{jc}^i is defined in **Lemma 1**,

$\hat{U}$ is an $(n - k_1) \times (m - k_2)$ i.i.d. $N\left(0, \frac{1}{m}\right)$ matrix in where m and n are defined in **Assumption 1**.

The following bound in eqn. (C.34) holds w.p. at least $1 - \exp(-2\varepsilon^*m)$ by applying eqn. (C.2).

$$\begin{aligned} \sigma_{\min}(\hat{U}) &\geq \sqrt{\delta - \alpha^*} - \sqrt{1 - \rho} - \sqrt{4\varepsilon^*} \\ &\geq \sqrt{\delta} - \sqrt{\alpha^*} - \sqrt{1 - \rho} - \sqrt{4\varepsilon^*} \end{aligned} \quad (\text{C.34})$$

where

δ and ρ are defined in **Assumption 3**,
 ε^* and α^* are the short forms of $\varepsilon_{\delta,\rho,v_{max}}^*$ and $\alpha_{\delta,\rho,v_{max}}^*$ and are defined in eqn. (4.15).

Since $\rho \geq \frac{15}{16}$, $\sqrt{1-\rho} \leq \frac{1}{4} \leq \frac{\sqrt{\delta}}{4}$, $\sqrt{4\varepsilon^*} \leq 2^{-8}\sqrt{\delta}$ from eqn. (C.29) and $\sqrt{\alpha^*} \leq 2^{-8} \leq 2^{-8}\sqrt{\delta}$ from eqn. (C.28), the good event in eqn. (C.39) holds w.p. $\geq 1 - \exp(-2\varepsilon^*m)$.

$$\sigma_{\min}(U) \geq v_{\min}\sqrt{\delta} \times \left(1 - 2^{-8} - \frac{1}{4} - 2^{-8}\right) > \frac{v_{\min}\sqrt{\delta}}{2} \quad (\text{C.35})$$

Eqn. (C.18) is proven.

(iii) Proof of eqn. (C.21):

The columns of $\widehat{U}^*$ ($\widehat{U}$ is defined in eqn. (C.74)) are projected by the matrix P_W onto the independent $k_1 + N$ -dimensional subspace $\text{range}(W^*)$ (W is defined in eqn. (C.56)) in where

P_W is defined in eqns (C.63)-(C.65),
 k_1 is defined in Section 4.1,
 N is defined in **Assumption 1**, and
 W is defined in eqn. (C.56).

As W and $\widehat{U}$ are probabilistically independent, $\beta_s(P_W\widehat{U}^*)$ is distributed as $\beta_s(\widetilde{M})$, where $\widetilde{M}$ is a $(k_1 + N) \times (n - k_1)$ i.i.d. $N\left(0, \frac{1}{m}\right)$ matrix (in where m and n are defined in **Assumption 1**). By using eqns (C.7), (C.23), (C.25), (C.28) and (C.29), the following bound in eqn. (C.36) holds w.p. at least $1 - \exp(-2\varepsilon^*m)$.

$$\begin{aligned}
\beta_s(P_W \hat{U}^*) &\leq \sqrt{\frac{k_1 + 1}{m}} + \sqrt{\frac{s}{m}} + \sqrt{2 \frac{s}{m} \log \left(\frac{n - k_1}{s} \right)} + 4\varepsilon^* \\
&\leq \sqrt{\alpha^* + \frac{1}{m}} + \sqrt{C_s} + \sqrt{2C_s \log \left(\frac{\delta}{C_s} \right)} + \sqrt{4\varepsilon^*} \\
&\leq 4 \times 2^{-8}(1 - \rho) \\
&= 2^{-6}(1 - \rho) \\
&< 2^{-3} \sqrt{1 - \rho}
\end{aligned} \tag{C.36}$$

where

- β_s is defined in eqn. (C.4),
- s is defined in eqn. (4.23),
- n is defined in **Assumption 1**,
- α^* and ε^* are the short form of $\alpha_{\delta, \rho, v_{\max}}^*$ and $\varepsilon_{\delta, \rho, v_{\max}}^*$ and are defined in eqn. (4.15),
- ρ is defined in **Assumption 3**, and
- C_s is defined in eqn. (4.10).

Hence, eqn. (C.44) holds w.p. at least $1 - \exp(-2\varepsilon^* m)$

$$\beta_s(P_W \hat{U}^*) < 2^{-3} v_{\max} \sqrt{1 - \rho} \tag{C.37}$$

Eqn. (C.21) is proven.

(iv) Proof of eqn. (C.22):

Lemma 10 shows that eqn. (C.38) holds.

$$\begin{aligned}
\sigma_{\min}(W) &> \frac{1}{4} \\
\text{w.p. } &\geq 1 - (2 + N) \exp(-2\varepsilon^* m)
\end{aligned} \tag{C.38}$$

where

- W is defined in eqn. (C.56),
- m and N are defined in **Assumption 1**, and
- ε^* is the short form of $\varepsilon_{\delta, \rho, v_{\max}}^*$ and is defined in eqn. (4.15).

Moreover, by **Lemma 7**, whenever $\sigma_{\min}(W) > 0$ has full row rank $k_1 + N$, eqn. (C.11) holds.

$$\beta_s(P_W) \leq \frac{\beta_s(W)}{\sigma_{\min}(W)} \quad \text{Recap (C.11)}$$

where

- k_1 is defined in Section 4.1,
- β_s is defined in eqn. (C.4),
- s is defined in eqn. (4.23), and
- P_W is defined in eqns (C.63)-(C.65),

Eqn. (C.39) holds with the definition of W and eqn. (C.8).

$$\beta_s(W) \leq \frac{\beta_s((Z_{J^c, I})^*)}{\sqrt{1-\rho}} + \frac{\sum_i^N \beta_s((\mu_{J^c}^i)^*)}{N\sqrt{1-\rho}} \quad (\text{C.39})$$

where

- $Z_{J^c}^i$ and $\mu_{J^c}^i$ is defined in **Lemma 1**, and
- ρ is defined in **Assumption 3**.

Using eqn. (C.7) , eqn. (C.40) holds w.p. at least $1 - \exp(-2\varepsilon^*m)$.

$$\begin{aligned} \beta_s((Z_{J^c, I})^*) &\leq \sqrt{\frac{k_1}{m}} + \sqrt{\frac{s}{m}} + \sqrt{2\frac{s}{m} \log\left(\frac{(1-\rho)m}{s}\right) + 4\varepsilon^*} \\ &\leq \sqrt{\alpha^*} + \sqrt{C_s} + \sqrt{2C_s \log\left(\frac{(1-\rho)m}{s}\right) + 4\varepsilon^*} \\ &\quad + \sqrt{4\varepsilon^*} \\ &\leq 4 \times 2^{-8}(1-\rho) \\ &= 2^{-6}(1-\rho) \end{aligned} \quad (\text{C.40})$$

where

- α^* is the short form of $\alpha_{\delta, \rho, v_{\max}}^*$ and is defined in eqn. (4.15),
- C_s is defined in eqn. (4.10).

Regarding the second term in (C.39), as μ_{jc}^{i*} is simply a row vector of length $m - k_2$, and so $\beta_s(\mu_{jc}^{i*})$ is simply the largest Euclidean norm of any vector formed from s elements of μ_{jc} , eqn. (C.41) holds when $\|\mu_i\|_\infty \leq C_\mu m^{-\frac{1}{2}}$ is used, and eqn. (C.23) is invoked to bound $C_\mu \sqrt{C_s}$.

$$\beta_s(\mu_{jc}^{i*}) \leq \beta_s(\mu_i^*) = \sup_{\|x\|_2 \leq 1, \|x\|_0 \leq s} \|\mu_i^* x\|_2 = C_\mu \sqrt{\frac{s}{m}} = C_\mu \sqrt{C_s} \leq 2^{-8}(1 - \rho) \quad (\text{C.41})$$

where

μ_i and C_μ is defined in **Assumption 1**, and

As a result, eqn. (C.42) holds by applying eqns (C.40) and (C.41) to eqn. (C.39).

$$\begin{aligned} \beta_s(W) &\leq 2^{-6}\sqrt{1-\rho} + 2^{-8}\sqrt{1-\rho} < 2^{-5}\sqrt{1-\rho} \\ \text{w.p.} &\geq 1 - \exp(-2\varepsilon^*m) \end{aligned} \quad (\text{C.42})$$

Furthermore, eqn. (C.43) holds by applying eqns (C.38) to (C.42),

$$\begin{aligned} \beta_s(P_W) &\leq \frac{\sqrt{1-\rho}}{8} \\ \text{w.p.} &\geq 1 - (3 + N)\exp(-2\varepsilon^*m) \end{aligned} \quad (\text{C.43})$$

The proof for eqn. (C.22) is completed.

C.1.5. Concentration for Functions of i.i.d. Rademacher Sequences

The following general concentration lemma for functions of i.i.d. Rademacher sequences are largely used in the proof of **Lemma 4**.

Lemma 11 (Lemma 5 in [4])

Let any matrix $M \in \mathbb{R}^{m \times n}$ and $\sigma = (\sigma_1, \dots, \sigma_n)$ be a sequence of i.i.d. Rademacher (symmetric ± 1) random variables. Moreover, suppose S denotes the shrinkage operator with a threshold of $\frac{1}{2}$, which is defined in Section 4.1, the following two bounds in eqns can be formed.

$$\mathbb{P}[\|M\sigma\| > 2\|M\|_F + 4\|M\|\sqrt{\varepsilon n}] < 4\exp(-\varepsilon n) \quad (C.44)$$

$$\mathbb{P}\left[\|S[M\sigma]\| > 2\|M\|_F \exp\left(-\frac{1}{4}\|M\|_{2,\infty}^2\right) + 4\|M\|\sqrt{\varepsilon n}\right] < 4\exp(-\varepsilon n) \quad (C.45)$$

C.2. Technical Lemmas

Lemmas 12 - 13 which shows the reductions to large error fractions mimics the **Lemmas 12 - 13** in [4].

Lemma 12

Suppose $A \in \mathbb{R}^{m \times n}$, (x_o, e_o) is the unique optimal solution to eqn. (C.46)

$$\begin{aligned} & \min \|x\|_1 + \lambda \|e\|_1 \\ \text{s.t. (such that)} \quad & Ax + e = Ax_o + e_o \end{aligned} \quad (C.46)$$

If $\text{supp}(x_o) \subseteq I$, $(x_o(I), e_o)$ is the unique optimal solution to eqn. (C.47).

$$\begin{aligned} & \min \|\dot{x}\|_1 + \lambda \|e\|_1 \\ \text{subject to } & A_{[m],I} \dot{x} + e = A_{[m],I} x_o(I) + e_o \end{aligned} \quad (C.47)$$

Lemma 13

Let $A \in \mathbb{R}^{m \times n}$ and (x_o, e_o) is the unique optimal solution to eqn. (C.48).

$$\min \|x\|_1 + \lambda \|e\|_1 \text{ subject to } Ax + e = Ax_o + e_o \quad (C.48)$$

Suppose $J = \text{supp}(e_o)$, $J' \subset J$, and e'_o is constructed by letting $e'_o(J') = e_o(J')$, $e'_o(J'^c) = 0$. (x_o, e'_o) is then the unique optimal solution to eqn. (C.49).

$$\min \|x\|_1 + \lambda \|e\|_1 \text{ subject to } Ax + e = Ax_o + e'_o \quad (C.49)$$

C.3. Proofs of Lemmas

These sections show the proof of various lemmas which are used in Chapter 3 and this appendix. Some lemmas which are borrowed from the other references are stated in the sections corresponding to lemmas.

C.3.1. Proof of Lemma 1

Based on the duality principle (e.g., **Theorem 4** in [83]), if there is any $h \in \mathbb{R}^m$ which can satisfy eqns (C.50) and (C.51),

$$\begin{cases} (A^*)_I h = \sigma_x, h_J = \lambda \sigma \\ \|(A^*)_{I^c} h\|_\infty < 1, \|h_{J^c}\|_\infty < \lambda \end{cases} \quad (\text{C.50})$$

$$\tilde{B} = [A_{[m],I} (I)_{[m],J}] \text{ has full rank } k_1 + k_2 \quad (\text{C.51})$$

where

$$I = \text{supp}(x_o) \subset [n],$$

$$J = \text{supp}(e_o) \subset [m],$$

$$\sigma_x = \text{sign}(x_o(I)) \in \{\pm 1\}^{k_1},$$

$$\sigma = \text{sign}(e_o(J)) \in \{\pm 1\}^{k_2}$$

$$k_1 = |I|,$$

$$k_2 = |J|,$$

x_o is original/ground truth sparse coefficient vector,

e_o is original/ground truth corrupting error vector,

A is defined in **Assumption 1**, and

λ is the tradeoff parameter in eqn. (4.19).

(x_o, e_o) is then the unique optimal solution to eqn. (4.19).

Eqn. (4.18) implies that eqn. (C.51) has $\text{rank}(\tilde{B}) = k_2 + \text{rank}(A_{J^c,I})$. By reformulating the system into the eqn. (C.52),

$$\begin{bmatrix} (A_{J^c,I^c})^* & \frac{1}{\lambda} I \\ (A_{J^c,I})^* & 0 \end{bmatrix} q = \begin{bmatrix} -(A_{J,I^c})^* \sigma \\ \frac{1}{\lambda} \sigma_x - (A_{J,I})^* \sigma \end{bmatrix}, \|q\|_\infty < 1 \quad (\text{C.52})$$

with unknown $q \in \mathbb{R}^{m+n-k_1-k_2}$ in which I is an identity matrix with the corresponding size. There is a dual vector h satisfying eqn. (C.50) with $h_J = \lambda \sigma$ and $h_{J^c} = q_{1\dots m-k_2}$ if there exist q satisfying eqn. (C.52). Hence, any q satisfying eqn. (4.17) also satisfying eqn. (C.52) and it completes the proof.

As the result, only a vector q of small L_∞ -norm satisfying eqn. (4.17) is needed to be constructed. Eqn. (4.17) implies that the $n + N$ equality constraints in eqn. (4.17) may have very different scale if v_i is small for $\forall i$. Hence, an equivalent set of constraints that is better

conditioned and the notation for the various components will be introduced for replacing the original equality constraints.

The left top component $\begin{bmatrix} v_1(Z_{jc}^1)^* \\ \vdots \\ v_N(Z_{jc}^N)^* \end{bmatrix} \frac{1}{\lambda} \mathbf{I} q$ of eqn. (4.17) has norm ≈ 1 for each row. Two

new components are created for the top component of eqn. (4.17), let eqns (C.53) and (C.54).

$$U = \begin{bmatrix} v_1(Z_{jc}^1)^* \\ \vdots \\ v_N(Z_{jc}^N)^* \end{bmatrix} \in \mathbb{R}^{(n-k_1) \times (m-k_2)} \quad (\text{C.53})$$

$$R = \begin{bmatrix} -v_1(Z_j^1)^* \\ \vdots \\ -v_N(Z_j^N)^* \\ \mathbf{0} \end{bmatrix} \in \mathbb{R}^{n+N \times k_2} \quad (\text{C.54})$$

where

- v_i is a deviation amplifier and is defined in **Assumption 1**,
- Z_j^i is a submatrix of Z_i containing the rows indexed by J , and
- Z_i is defined in eqn. (4.16).

The middle left component $[v_t(Z_{jc,t}^t)^* \quad \mathbf{0}] q$ of eqn. (4.17) has a norm $\approx v_t \sqrt{1-\rho}$ for each row. Each row is rescaled by $v_t^{-1}(1-\rho)^{-\frac{1}{2}}$.

The last equalities in eqn. (4.17) can be written as eqn. (C.55),

$$\begin{bmatrix} \mu_{jc}^1 \\ \vdots \\ \mu_{jc}^N \end{bmatrix} q = - \begin{bmatrix} \mu_j^1 \\ \vdots \\ \mu_j^N \end{bmatrix} \sigma \quad (\text{C.55})$$

where

- μ_j^1 is subvector of μ_i containing rows indexed by J , and
- μ_i is defined in (4.6).

and each row has norm $\approx \sqrt{1-\rho}$ in which ρ is defined in **Assumption 3** such that $\rho <$

1. The new equality is rescaled by $(1-\rho)^{-\frac{1}{2}}$. Write eqns (C.56)-(C.58).

$$W = (1 - \rho)^{-\frac{1}{2}} \begin{bmatrix} (Z_{j^c, I_t}^t)^* \\ \frac{1}{N} (\mu_{j^c}^1)^* \\ \vdots \\ \frac{1}{N} (\mu_{j^c}^N)^* \end{bmatrix} \in \mathbb{R}^{(k_1+N) \times (m-k_2)} \quad (\text{C.56})$$

$$T = (1 - \rho)^{-\frac{1}{2}} \begin{bmatrix} \mathbf{0} \\ -(Z_{j, I_t}^t)^* \\ -\frac{1}{N} (\mu_j^1)^* \\ \vdots \\ -\frac{1}{N} (\mu_j^N)^* \end{bmatrix} \in \mathbb{R}^{(n+N) \times k_2} \quad (\text{C.57})$$

$$z = (1 - \rho)^{-\frac{1}{2}} v_t^{-1} \begin{bmatrix} \mathbf{0} \\ \frac{1}{\lambda} \sigma_x \\ \mathbf{0} \end{bmatrix} \in \mathbb{R}^{n+N} \quad (\text{C.58})$$

Where I_t , Z_{j, I_t}^t and v_t are defined in eqn. (4.17) and $\mathbf{0}$ is either a zero vector or matrix with corresponding size.

Further, set eqn. (C.59).

$$\Phi = \begin{bmatrix} U & \frac{1}{\lambda} \mathbf{I} \\ W & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{(n+N) \times (m+n-k_1-k_2)} \quad (\text{C.59})$$

Hence, if and only if q satisfies eqn. (C.60),

$$\Phi q = R\sigma + T\sigma + z \quad (\text{C.60})$$

q is a solution to eqn. (4.17).

C.3.2. Proof of Lemma 3

The proof has two parts:

1. the quantity of interest is first reduced to a function of the singular values of several submatrices of U and W , and then
2. apply Gaussian measure concentration to bound these quantities.

A. Lower Bounding $1 - \xi$

The lower bounding $1 - \xi_s(\Phi)$ can be expressed as eqn. (C.61).

$$\begin{aligned}
1 - \xi_s(\Phi) &\geq \frac{1}{2} (1 - \xi_s(\Phi)^2) \\
&= \frac{1}{2} \left(1 - \sup_{\substack{\|w\| \leq 1, \\ \|w\|_0 \leq s}} \|P_\Phi w\|^2 \right) \\
&= \frac{1}{2} \left(1 - \sup_{\substack{\|w\| \leq 1, \\ \|w\|_0 \leq s}} \|w\|^2 - \|w - P_\Phi w\|^2 \right) \\
&= \frac{1}{2} \left(1 - \sup_{\substack{\|w\| = 1, \\ \|w\|_0 \leq s}} 1 - \|w - P_\Phi w\|^2 \right) \\
&= \frac{1}{2} \inf_{\substack{\|w\| = 1, \\ \|w\|_0 \leq s}} \|w - P_\Phi w\|^2 \\
&= \frac{1}{2} \inf_{\substack{\|w\| = 1, \\ \|w\|_0 \leq s}} \min_r \|w - \Phi^* r\|^2
\end{aligned} \tag{C.61}$$

where $\xi_s(\Phi)$ is defined in eqn. (4.20).

The bound can be written more explicitly as eqn. (C.62) using eqn. (C.59).

$$\begin{aligned}
&\frac{1}{2} \inf_{\substack{\|w_1\|^2 + \|w_2\|^2 = 1, \\ \|w_1\|_0 + \|w_2\|_0 \leq s}} \min_r \left\| \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} - \begin{bmatrix} U^* & W^* \\ \frac{1}{\lambda} \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} \right\|^2 \\
&= \frac{1}{2} \inf_{w_1, w_2} \min_{u_1, u_2} \left\| \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} - \begin{bmatrix} U^* & W^* \\ \frac{1}{\lambda} \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} u_1 + w_2 \\ u_2 \end{bmatrix} \right\|^2 \\
&= \frac{1}{2} \inf_w \min_u \left\| \frac{1}{\lambda} w_1 - U^* u_1 - U^* w_2 - W^* u_2 \right\|^2 + \left\| \frac{1}{\lambda} u_1 \right\|^2
\end{aligned} \tag{C.62}$$

where

s is defined in eqn. (4.23),

λ is defined in eqn. (4.8),

$\mathbf{0}$ is a zero matrix with the corresponding size,

U and W are defined in eqns (C.53) and (C.56).

Notice that $W \in \mathbb{R}^{(k_1+N) \times (1-\rho)m}$ has a nontrivial null space; suppose $\Gamma \subset \mathbb{R}^{(1-\rho)m} = \text{null}(W)$, P_Γ denotes the projection operator onto Γ , and P_W denotes the projection operator onto the range of W^* , so that it fulfils eqns (C.63)-(C.65).

$$P_W + P_\Gamma = I \quad (\text{C.63})$$

$$P_W P_\Gamma = 0 \quad (\text{C.64})$$

$$P_\Gamma W^* = 0 \quad (\text{C.65})$$

where I is an identity matrix with the corresponding size.

Then, the first term on the right side of eqn. (C.62) can be written in terms of these operators as in eqn. (C.66).

$$\begin{aligned} & \left\| \frac{1}{\lambda} w_1 - U^* u_1 - U^* w_2 - W^* u_2 \right\|^2 \\ &= \left\| P_\Gamma \left(\frac{1}{\lambda} w_1 - U^* u_1 - U^* w_2 - W^* u_2 \right) \right\|^2 \\ &+ \left\| P_W \left(\frac{1}{\lambda} w_1 - U^* u_1 - U^* w_2 - W^* u_2 \right) \right\|^2 \\ &\geq \left\| P_\Gamma \left(\frac{1}{\lambda} w_1 - U^* u_1 - U^* w_2 - W^* u_2 \right) \right\|^2 \\ &= \left\| P_\Gamma U^* u_1 - \left(\frac{1}{\lambda} P_\Gamma w_1 - P_\Gamma U^* w_2 \right) \right\|^2 \end{aligned} \quad (\text{C.66})$$

Plugging eqn. (C.66) into eqn. (C.62), $1 - \xi_s(\Phi)$ is bounded by eqn. (C.67)

$$\begin{aligned} & \frac{1}{2} \inf_{w_1, w_2} \min_{u_1} \left\| P_\Gamma U^* u_1 - \left(\frac{1}{\lambda} P_\Gamma w_1 - P_\Gamma U^* w_2 \right) \right\|^2 + \left\| \frac{1}{\lambda} u_1 \right\|^2 \\ &= \frac{1}{2\lambda^2} \inf_{w_1, w_2} \min_{u_1} \left\| \lambda P_\Gamma U^* u_1 - (P_\Gamma w_1 - \lambda P_\Gamma U^* w_2) \right\|^2 + \|u_1\|^2 \end{aligned} \quad (\text{C.67})$$

For any matrix M and vector x and y , the unique solution to eqn. (C.68) is given by $x^*(\tilde{y}) = (I + M^* M)^{-1} M^* \tilde{y}$.

$$\min_x \|Mx - \tilde{y}\|^2 + \|x\|^2 \quad (\text{C.68})$$

Hence, in eqn. (C.67), for any $w = [w_1^*, w_2^*]^*$, the optimizer $u_1^*(w)$ has the explicit form as in eqn. (C.69).

$$\begin{aligned} u_1^*(w) &= \lambda(I + \lambda^2 U P_\Gamma U^*)^{-1} U P_\Gamma (w_1 - \lambda U^* w_2) \\ &= \lambda(I + \lambda^2 U P_\Gamma U^*)^{-1} U \Psi w \end{aligned} \quad (\text{C.69})$$

where Ψ denotes the matrix $[P_\Gamma | -\lambda P_\Gamma U^*]$.

Notice that eqn. (C.70) is valid for any w .

$$\begin{aligned} \|u^*(w)\| &\geq \frac{\lambda \sigma_{\min}(U)}{\|I + \lambda^2 U P_\Gamma U^*\|} \|\Psi w\| \\ &\geq \frac{\lambda \sigma_{\min}(U)}{1 + \lambda^2 \|U\|^2} \|\Psi w\| \end{aligned} \quad (\text{C.70})$$

Hence, eqn. (C.71) is valid for any w satisfying $\|w\| = 1$ and $\|w\|_0 \leq s$

$$\|u^*(w)\| \geq \frac{\lambda \sigma_{\min}(U)}{1 + \lambda^2 \|U\|^2} \gamma_s(\Psi) \quad (\text{C.71})$$

where function γ_s is defined in eqn. (C.5).

Applying eqns (C.67)-(C.71) to eqns (C.61) and (C.62), eqn. (C.72) is established.

$$\begin{aligned} 1 - \xi_s(\Phi) &\geq \frac{1}{2} \inf_{\substack{\|w\|=1 \\ \|w\|_0 \leq s}} \|\mu^*(w)\|^2 \\ &\geq \frac{\lambda^2 \sigma_{\min}^2(U)}{2(1 + \lambda^2 \|U\|^2)^2} \gamma_s^2(\Psi) \end{aligned} \quad (\text{C.72})$$

Apply a sequence of straightforward algebraic manipulations to lower bound $\gamma_s(\Psi)$ in terms of the singular values of submatrices of U and W , the inequality in eqn. (C.73) holds. Each step below should be intuitive and the correctness of them is proven in **Lemmas 5 - 8** of the appendix.

$$\begin{aligned} \gamma_s(\Psi) &\geq \lambda v_{\min} \gamma_s([P_\Gamma | P_\Gamma \hat{U}^*]) && \text{by (C.8)} \\ &= \lambda v_{\min} \gamma_s([I - P_W | (I - P_W) \hat{U}^*]) && (\text{C.73}) \\ &\geq \lambda v_{\min} (\gamma_s([I \hat{U}^*]) - \beta_s(P_W) - \beta_s(P_W \hat{U}^*)) && \text{by (C.9)} \end{aligned}$$

where

$v_{\min}$ is defined in eqn. (4.15),

$\mathbf{I}$ is defined in Section 4.1, and

$\hat{\mathbf{U}}$ is defined in eqn. (C.74).

$$\hat{\mathbf{U}} = \begin{bmatrix} (Z_{jc}^1)^* \\ \vdots \\ (Z_{jc}^N)^* \end{bmatrix} \in \mathbb{R}^{(n-k_1) \times (m-k_2)} \quad (\text{C.74})$$

where Z_j^i is defined in eqn. (4.16).

When eqn. (C.8) is applied in eqn. (C.73), the following bound in eqn. (C.75) is used,

$$\lambda v_{\min} \leq \lambda v^* = 2^{-12} \sqrt{\frac{C_s}{\delta}} \ll 1 \quad (\text{C.75})$$

where

v^* is the short form of $v_{\delta, \rho}^*$ and is defined in eqn. (4.15),

C_s is defined in eqn. (4.10), and

δ is defined in **Assumption 3**.

Finally, with (C.12) in **Lemma 8**, the following bound in eqn. (C.76) is established.

$$\gamma_s([\mathbf{I}\hat{\mathbf{U}}^*]) \geq \sqrt{\min\{1, \gamma_s^2(\hat{\mathbf{U}}^*)\} - \phi_s(\hat{\mathbf{U}}^*)} \quad (\text{C.76})$$

where function ϕ_s is defined in eqn. (C.6).

When all the above bounds in eqns (C.17)-(C.22) hold, then the following eqn. (C.77) is formed.

$$\gamma_s(\Psi) \geq \frac{\lambda v_{\min} \sqrt{1 - \rho}}{4} \quad (\text{C.77})$$

where

Ψ is defined in eqn. (C.69),

λ is defined in eqn. (4.8),

$v_{\min}$ is defined in eqn. (4.15).

By eqn. (C.72) and the bounds in eqns (C.17)-(C.22), the following bound in eqn. (C.78) is established,

$$1 - \xi_s(\Phi) \geq \frac{\lambda^2 v_{min}^2 \delta}{8(1 + 9\lambda^2 v_{max}^2 \delta)^2} \gamma_s^2(\Psi) \geq \frac{\lambda^2 v_{min}^2 \delta}{32} \gamma_s^2(\Psi) \quad (C.78)$$

where the bound in eqn. (C.79) is used.

$$\lambda^2 v_{min}^2 \leq \lambda^2 v_{max}^2 \leq \lambda^2 v^* = 2^{-12} \sqrt{\frac{C_s}{\delta}} < \frac{2^{-12}}{\sqrt{\delta}} \ll \frac{1}{3\sqrt{\delta}} \quad (C.79)$$

where

v^* is the short form of $v_{\delta, \rho}^*$ and is defined in eqn. (4.15), and

C_s is defined in eqn. (4.10).

The bound in eqn. (C.80) can be obtained by combining the two bounds in eqns (C.77) and (C.78) on the intersection of the events in eqns (C.17)-(C.22); the bound holds with probability at least $1 - (8 + N)\exp(-2\varepsilon^* m)$ by summing the failure probabilities.

$$1 - \xi_s(\Phi) \geq \frac{\lambda^4 v_{min}^4 \delta (1 - \rho)}{512} \quad (C.80)$$

The proof is concluded by establishing the probabilistic bounds in eqns (C.17)-(C.22).

C.3.3. Proof of Lemma 4

By eqn. (4.15), $\alpha^* = \lambda^2 v_{max}^2 \Upsilon\left[\frac{\varepsilon^*}{\delta}\right] \leq \frac{\lambda^2 v_{max}^2 \varepsilon^*}{\delta} \leq \frac{\lambda^4 v_{max}^4 \tau^2 (1 - \rho)}{2\delta}$, where $\Upsilon[t] \leq t$ for all t is used. With $\delta \geq 1$, the following slightly weaker bound is formed,

$$\alpha^* \leq \lambda^4 v_{max}^4 \tau^2 (1 - \rho) \quad (C.81)$$

where

α^* and ε^* are the short forms of $\alpha_{\delta, \rho, v_{max}}^*$ and $\varepsilon_{\delta, \rho, v_{max}}^*$ and are defined in eqn. (4.15),

λ is defined in eqn. (4.8),

v_{max} and τ is defined in eqn. (4.9),

Υ is defined in eqn. (4.12), and

δ and ρ are defined in **Assumption 3**.

Furthermore, $\alpha^* \ll 1$ is seen.

Conditioning of Φ (Φ is defined in eqn. (C.59)). Whenever $\Phi\Phi^*$ is invertible, the pseudoinverse operator defined in eqn. (C.82) is well defined, and eqn. (C.83) holds. Let $\mathcal{E}_\Phi$ denotes the event in eqn. (C.84). Recall the definition of Φ in eqn. (C.59), eqn. (C.85) is established.

$$\Phi^\dagger = \Phi^*(\Phi\Phi^*)^{-1} \quad (\text{C.82})$$

$$\|\Phi^\dagger\| = \frac{1}{\sigma_{\min}(\Phi)} \quad (\text{C.83})$$

$$\mathcal{E}_\Phi = \{\Phi\Phi^* \text{ is invertible, and } \|\Phi^\dagger\| \leq 8\} \quad (\text{C.84})$$

$$\sigma_{\min}(\Phi) \geq \min(1, \sigma_{\min}(W)) - \|U\| \quad (\text{C.85})$$

where U and W are defined in eqns (C.53) and (C.56) respectively.

Lemma 10 presents that $\sigma_{\min}(W) \geq \frac{1}{4}$ holds w.p. at least $1 - (2 + N)\exp(-2\varepsilon^*m)$ By eqn. (C.1) to get bound of $\|\hat{U}\|$, eqn. (C.86) holds w.p. $1 - \exp(-2\varepsilon^*m)$ where the bound in eqn. (C.87) is used.

$$\begin{aligned} \|U\| &\leq v_{\max}\|\hat{U}\| \\ &\leq v_{\max}(\sqrt{\delta} + \sqrt{1-\rho} + \sqrt{4\varepsilon^*}) \\ &\leq 3v_{\max}\sqrt{\delta} \\ &\leq \frac{1}{8} \end{aligned} \quad (\text{C.86})$$

$$v^* < \frac{1}{24\lambda\sqrt{\delta}} \quad (\text{C.87})$$

where

m	is defined in Assumption 1 ,
N	is defined in Assumption 1 ,
U and W	are defined in eqns (C.53) and (C.56) respectively,
$\hat{U}$	is defined in eqn. (C.74).
$v_{\max}$	is defined in eqn. (4.9),
ε^* and v^*	are the short forms of $\varepsilon_{\delta,\rho,v_{\max}}^*$ and $v_{\delta,\rho}^*$ and are defined in eqn. (4.15),
δ and ρ	are defined in Assumption 3 , and
λ	is defined in eqn. (4.8).

Hence, eqns (C.90) and (C.89) hold.

$$\sigma_{\min}(\Phi) \geq \frac{1}{8} \quad (\text{C.88})$$

$$w.p. \geq 1 - (3 + N)\exp(-2\varepsilon^*m)$$

$$\mathbb{P}[\mathcal{E}_\Phi] \geq 1 - (3 + N)\exp(-2\varepsilon^*m) \quad (\text{C.89})$$

where

Φ is defined in eqn. (C.59), and

$\mathcal{E}_\Phi$ denotes the event in eqn. (C.84).

Bounding the solution. So, the minimum L_2 -norm solution q_0 to eqn. (C.60) is given by eqn. (C.90) on $\mathcal{E}_\Phi$.

$$q_0 = \Phi^\dagger R\sigma + \Phi^\dagger T\sigma + \Phi^\dagger z \quad (\text{C.90})$$

where

R, T and z are defined in eqns (C.54), (C.57) and (C.58),

σ is defined in Section 4.1,

Φ is defined in eqn. (C.59), and

$\mathcal{E}_\Phi$ denotes the event in eqn. (C.84).

The following four bad events in eqns (C.91)-(C.94) are defined, three of them involve the deviation of norms of the elements of q_0 from their expectations, and the other one involves the norm of a soft-thresholded version of the first component deviating above its expectation.

$$\mathcal{E}_1 = \{\sigma_{\min}(\Phi) = 0, \text{ or } \|\Phi^\dagger R\sigma\| > 128\nu_{\max}\sqrt{\delta m}\} \quad (\text{C.91})$$

$$\mathcal{E}_2 = \{\sigma_{\min}(\Phi) = 0, \text{ or } \|S[\Phi^\dagger R\sigma]\| > 128\nu_{\max}\tau\sqrt{\delta m}\} \quad (\text{C.92})$$

$$\mathcal{E}_3 = \left\{ \sigma_{\min}(\Phi) = 0, \text{ or } \|\Phi^\dagger T\sigma\| > 176\nu_{\max}\tau\sqrt{\delta m} + \frac{48}{\sqrt{1-\rho}} \right\} \quad (\text{C.93})$$

$$\mathcal{E}_4 = \{\sigma_{\min}(\Phi) = 0, \text{ or } \|\Phi^\dagger z\| > 8\lambda\nu'\nu_{\max}\tau\sqrt{\delta m}\} \quad (\text{C.94})$$

where

m is defined in **Assumption 1**,

$\nu_{\max}$ and τ are defined in eqn. (4.9),

v_{min}	is defined in eqn. (4.15),
v'	is a constant equal to $\frac{v_{max}}{v_{min}}$,
δ and ρ	are defined in Assumption 3 , and
S	is defined in eqn. (4.2).

Noticing that all forms of the bounds in these events arise from the structure of eqn. (C.60) and the numerical constants are pessimistic. The following bound will be got with using $\tau \leq 1$ and when events of $\cap_i \mathcal{E}_i^c$ happen, eqn. (C.95) holds.

$$\begin{aligned} \|q_0\| &\leq \|\Phi^\dagger R \sigma\| + \|\Phi^\dagger T \sigma\| + \|\Phi^\dagger z\| \\ &\leq (128 + 176\lambda + 8\lambda v') v_{max} \sqrt{\delta m} + O(1) \end{aligned} \quad (C.95)$$

where

q_0	is defined in Section 4.7.2, and
λ	is defined in eqn. (4.8),

As for any vectors x, y $\|S[x - y]\| \leq \|S[x]\| + \|y\|$, on $\cap_i \mathcal{E}_i^c$, hence, the following bound in eqn. (C.96) is formed.

$$\begin{aligned} \|S[q_0]\| &\leq \|S[\Phi^\dagger R \sigma]\| + \|\Phi^\dagger T \sigma\| + \|\Phi^\dagger z\| \\ &\leq \left(128 + 176\lambda + 8\lambda \frac{v_{max}}{v_{min}}\right) v_{max} \tau \sqrt{\delta m} + O(1) \end{aligned} \quad (C.96)$$

The probabilities of the four bad events in eqns (C.91)-(C.94) are bounded as following,

$\mathcal{E}_1$ (eqn. (C.91)): bounding $\|\Phi^\dagger R \sigma\|$.

Applying eqns (C.1) and (C.3) on R , each of the following two bounds in eqns (C.97) and (C.98) holds w.p. at least $1 - \exp(-2\varepsilon^* m)$ and hence the two bounds hold simultaneously w.p. at least $1 - 2\exp(-2\varepsilon^* m)$. Note, $\|\hat{R}\|$ in eqns (C.97) and (C.98) is defined in eqn. (C.99).

$$\|R\| \leq v_{max} \|\hat{R}\| \leq v_{max} (\sqrt{\delta} + \sqrt{\rho} + \sqrt{4\varepsilon^*}) \leq 3v_{max} \sqrt{\delta} \quad (C.97)$$

$$\|R\|_F \leq v_{max} \|\hat{R}\|_F \leq v_{max} (\sqrt{\delta \rho m} + \sqrt{4\varepsilon^*}) \leq 2v_{max} \sqrt{\delta m} \quad (C.98)$$

$$\hat{R} = \begin{bmatrix} (Z_j^1)^* \\ \vdots \\ (Z_j^N)^* \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{n+N \times k_2} \quad (\text{C.99})$$

where

- R is defined in eqn. (C.54),
- v_{max} is defined in eqn. (4.9),
- Z_j^i is defined in **Lemma 1**,
- $\hat{U}$ is defined in eqn. (C.74).
- δ and ρ are defined in **Assumption 3**,
- ε^* is the short forms of $\varepsilon_{\delta, \rho, v_{max}}^*$ and is defined in eqn. (4.15), and
- m is defined in **Assumption 1**,

and the weak bound $\sqrt{4\varepsilon^*} < 1 \leq \sqrt{\delta}$ [established, e.g., in eqn. (C.29)] and $\sqrt{\rho} < 1 \leq \sqrt{\delta}$ (derived from the assumptions in **Lemma 4**) are used. Suppose $\mathcal{E}_R$ denotes the event that eqns (C.97)–(C.98) hold. On $\mathcal{E}_\Phi \cap \mathcal{E}_R$ in where $\mathcal{E}_\Phi$ denotes the event in eqn. (C.101), the following bound in eqn. (C.100) forms.

$$\begin{aligned} & 2\|\Phi^\dagger R\|_F + 4\|\Phi^\dagger R\|\sqrt{2\varepsilon^*m} \\ & \leq 2\|\Phi^\dagger\|\|R\|_F + 4\|\Phi^\dagger\|\|R\|\sqrt{2\varepsilon^*m} \\ & \leq 32 v_{max}\sqrt{\delta m} + 96v_{max}\sqrt{2\delta\varepsilon^*m} \\ & < 128v_{max}\sqrt{\delta m} \end{aligned} \quad (\text{C.100})$$

where $\sqrt{2\varepsilon^*} < 1$ is used again. Then the probability bound in eqn. (C.101) holds.

$$\begin{aligned} \mathbb{P}[\mathcal{E}_1] &= \mathbb{P}[\mathcal{E}_1|\mathcal{E}_\Phi \cap \mathcal{E}_R]\mathbb{P}[\mathcal{E}_\Phi \cap \mathcal{E}_R] \\ &+ \mathbb{P}[\mathcal{E}_1|(\mathcal{E}_\Phi \cap \mathcal{E}_R)^c]\mathbb{P}[(\mathcal{E}_\Phi \cap \mathcal{E}_R)^c] \\ &\leq \mathbb{P}[\mathcal{E}_1|\mathcal{E}_\Phi \cap \mathcal{E}_R] + \mathbb{P}[\mathcal{E}_\Phi^c \cap \mathcal{E}_R^c] \\ &\leq \sup_{Z_0, J_0 \in \mathcal{E}_\Phi \cap \mathcal{E}_R} \mathbb{P}[\mathcal{E}_1|Z_0, J_0] + \mathbb{P}[\mathcal{E}_\Phi^c] + \mathbb{P}[\mathcal{E}_R^c] \\ &\leq \sup_{Z_0, J_0 \in \mathcal{E}_\Phi \cap \mathcal{E}_R} \mathbb{P}[\mathcal{E}_1|Z_0, J_0] + (5 + N)\exp(-2\varepsilon^*m) \end{aligned} \quad (\text{C.101})$$

where N is defined in **Assumption 1**.

For any $(Z_0, J_0) \in \mathcal{E}_\Phi \cap \mathcal{E}_R$, applying **Lemma 11** with $\Phi^\dagger R$ fixed establishes the probability bounds in eqns (C.102) and (C.103).

$$\begin{aligned} \mathbb{P}[\mathcal{E}_1|Z_0, J_0] &= \mathbb{P}\left[\|\Phi^\dagger R\sigma\| > 128v_{\max}\sqrt{\delta m}|Z_0, J_0\right] \\ &\leq \mathbb{P}\left[\|\Phi^\dagger R\sigma\| > 2\|\Phi^\dagger R\|_F + 4\|\Phi^\dagger R\|\sqrt{2\varepsilon^* m}|Z_0, J_0\right] \\ &\leq 4\exp(-2\varepsilon^* m) \end{aligned} \quad (\text{C.102})$$

$$\mathbb{P}[\mathcal{E}_1] \leq (9 + N)\exp(-2\varepsilon^* m) \quad (\text{C.103})$$

$\mathcal{E}_2$ (eqn. (C.92)): bounding $\|\mathcal{S}[\Phi^\dagger R\sigma]\|$.

Applying the same calculation as in eqn. (C.101), probability bound in eqn. (C.104) forms.

$$\mathbb{P}[\mathcal{E}_2] \leq \sup_{Z_0, J_0 \in \mathcal{E}_\Phi \cap \mathcal{E}_R} \mathbb{P}[\mathcal{E}_2|Z_0, J_0] + (5 + N)\exp(-2\varepsilon^* m) \quad (\text{C.104})$$

where

- N is defined in **Assumption 1**,
- ε^* is the short forms of $\varepsilon_{\delta, \rho, v_{\max}}^*$ and is defined in eqn. (4.15),
- m is defined in **Assumption 1**,
- $\mathcal{E}_\Phi$ denotes the event in eqn. (C.84), and
- $\mathcal{E}_R$ denotes the event that eqns (C.97)–(C.98).

As the bounds in eqn. (C.105) holds, the bounds in eqns (C.106) and (C.107) hold for any fixed $Z_0, J_0 \in \mathcal{E}_\Phi \cap \mathcal{E}_R$.

$$\|\Phi^\dagger R\|_{2, \infty} \leq \|\Phi^\dagger R\| \leq \|\Phi^\dagger\| \|R\| \quad (\text{C.105})$$

$$\begin{aligned} \exp\left(-\frac{1}{4\|\Phi^\dagger R\|_{2, \infty}^2}\right) &\leq \exp\left(-\frac{1}{4\|\Phi\|^2\|R\|^2}\right) \\ &\leq \exp\left(-\frac{1}{48^2\delta v_{\max}^2}\right) \\ &= \tau \end{aligned} \quad (\text{C.106})$$

$$\begin{aligned}
& 2\|\Phi^\dagger R\|_F \exp\left(-\frac{1}{4\|\Phi^\dagger R\|_{2,\infty}^2}\right) + 4\|\Phi^\dagger R\|\sqrt{2\varepsilon^* m} \\
& \leq 32v_{\max}\tau\sqrt{\delta m} + 96v_{\max}\sqrt{2\varepsilon^* m} \\
& \leq 128v_{\max}\tau\sqrt{\delta m}
\end{aligned} \tag{C.107}$$

where

- Φ is defined in eqn. (C.59),
- R is defined in eqn. (C.54),
- δ is defined in **Assumption 3**,
- $v_{\max}$ and τ are defined in eqn. (4.9), and
- m is defined in **Assumption 1**.

and $\delta \geq 1$ (assumption in **Lemma 4**) and $\varepsilon^* < \frac{\tau^2}{2}$ derived from eqn. (4.15) are used in the last step of eqn. (C.107).

By **Lemma 11**, probability bound in eqn. (C.108) holds.

$$\begin{aligned}
\mathbb{P}[\mathcal{E}_2|Z_0, J_0] &= \mathbb{P}[\|S[\Phi^\dagger R\sigma]\| > 128v_{\max}\tau\sqrt{\delta m}|Z_0, J_0] \\
&\leq \mathbb{P}\left[\|S[\Phi^\dagger R\sigma]\| \right. \\
&\quad \left. > \left(2\|\Phi^\dagger R\|_F e^{-\frac{1}{4\|\Phi^\dagger R\|_{2,\infty}^2}} \right. \right. \\
&\quad \left. \left. + 4\|\Phi^\dagger R\|\sqrt{2\varepsilon^* m}\right)|Z_0, J_0\right] + \mathbb{P}[\mathcal{E}_\Phi^c \cap \mathcal{E}_R^c] \\
&\leq 4\exp(-2\varepsilon^* m)
\end{aligned} \tag{C.108}$$

where

- S is defined in eqn. (4.2), and
- σ is defined in Section 4.1.

Hence, eqn. (C.109) is concluded.

$$\mathbb{P}[\mathcal{E}_2] \leq (9 + N)\exp(-2\varepsilon^* m) \tag{C.109}$$

$\mathcal{E}_3$ (eqn. (C.93)): bounding $\|\Phi^\dagger T \sigma\|$.

Suppose $\|\cdot\|_\diamond$ is either the Frobenius or operator norm, the bound in eqn. (C.110) holds.

$$\|T\|_\diamond \leq \frac{\|Z_{J,I}\|_\diamond}{\sqrt{1-\rho}} + \frac{\sum_{i=1}^N \|\mu_j^i\|}{N\sqrt{1-\rho}} \leq \frac{\|Z_{J,I}\|_\diamond}{\sqrt{1-\rho}} + \frac{1}{\sqrt{1-\rho}} \quad (\text{C.110})$$

where

T is defined in eqn. (C.57),

$Z_{J,I}$ and μ_j^i is defined in **Lemma 1**, and

ρ is defined in **Assumption 3**.

Applying eqns (C.1) and (C.3) on $Z_{J,I}$, with probability at least $1 - 2\exp(-2\varepsilon^*m)$, the two following bounds in eqns (C.111) and (C.112) hold.

$$\|Z_{J,I}\| \leq \sqrt{\rho} + \sqrt{\alpha^*} + \sqrt{4\varepsilon^*} \leq 3 \quad (\text{C.111})$$

$$\|Z_{J,I}\|_F \leq \sqrt{\alpha^* \rho m} + \sqrt{4\varepsilon^*} \leq \sqrt{\alpha^* m} + 1 \quad (\text{C.112})$$

where

α^* and ε^* are the short forms of $\alpha_{\delta,\rho,v_{\max}}^*$ and $\varepsilon_{\delta,\rho,v_{\max}}^*$ and are defined in eqn. (4.15), and

m is defined in **Assumption 1**.

The loose final bounds in eqns (C.111) and (4.28) above only affect the numerical constants in eqns (C.113) and (C.114) below. As eqn. (C.81) implies that $\alpha^* \leq \lambda^2 v_{\max}^2 \tau^2$, the two following bounds in eqn. eqns (C.113) and (C.114) hold with probability at least $1 - 2\exp(-2\varepsilon^*m)$

$$\|T\| \leq \frac{\|Z_{J,I}\|}{\sqrt{1-\rho}} + \frac{1}{\sqrt{1-\rho}} \leq \frac{5}{\sqrt{1-\rho}} \quad (\text{C.113})$$

$$\|T\|_F \leq \frac{\|Z_{J,I}\|_F}{\sqrt{1-\rho}} + \frac{1}{\sqrt{1-\rho}} \leq \lambda v_{\max} \tau \sqrt{m} + \frac{3}{\sqrt{1-\rho}} \quad (\text{C.114})$$

where

λ is defined in eqn. (4.8), and

$v_{\max}$ and τ is defined in eqn. (4.9),

Let $\mathcal{E}_T$ denote the event that eqns (C.113) and (C.114) hold. On $\mathcal{E}_T \cap \mathcal{E}_\Phi$, the following bound in eqn. (C.115) holds.

$$\begin{aligned}
& 2\|\Phi^\dagger T\|_F + 4\|\Phi^\dagger T\|\sqrt{2\varepsilon^*m} \\
& \leq 2\|\Phi^\dagger\|\|T\|_F + 4\|\Phi^\dagger\|\|T\|\sqrt{2\varepsilon^*m} \\
& \leq 16\lambda v_{\max}\tau\sqrt{m} + 48(1-\rho)^{-\frac{1}{2}} + 160(1-\rho)^{-\frac{1}{2}}\sqrt{2\varepsilon^*m} \\
& \leq 176\lambda v_{\max}\tau\sqrt{\delta m} + 48(1-\rho)^{-\frac{1}{2}}
\end{aligned} \tag{C.115}$$

where

Φ is defined in eqn. (C.59), and

δ is defined in **Assumption 3**.

and the bound of $2\varepsilon^* \leq \lambda^2 v_{\max}^2 (1-\rho)$ which can be derived from eqn. (4.15) is used.

Moreover, the following probabilistic in eqn. (C.116) bound holds.

$$\begin{aligned}
\mathbb{P}[\mathcal{E}_3] & \leq \sup_{Z_0, J_0 \in \mathcal{E}_\Phi \cap \mathcal{E}_R} \mathbb{P}[\mathcal{E}_3|Z_0, J_0] + \mathbb{P}[\mathcal{E}_\Phi^c] + \mathbb{P}[\mathcal{E}_T^c] \\
& \leq \sup_{Z_0, J_0 \in \mathcal{E}_\Phi \cap \mathcal{E}_R} \mathbb{P}[\mathcal{E}_3|Z_0, J_0] + (5+N)\exp(-2\varepsilon^*m)
\end{aligned} \tag{C.116}$$

where

$\mathcal{E}_\Phi$ denotes the event in eqn. (C.101),

$\mathcal{E}_R$ denotes the event that eqns (C.97)–(C.98), and

N is defined in **Assumption 1**.

By **Lemma 11** with $\Phi^\dagger T$ is fixed, the following probability bound in eqn. (C.117) holds and hence, the probability bound in eqn. (C.118) is concluded.

$$\begin{aligned}
\mathbb{P}[\mathcal{E}_3|Z_0, J_0] & \leq \mathbb{P}\left[\|\Phi^\dagger T\sigma\| \geq 2\|\Phi^\dagger T\|_F + 4\|\Phi^\dagger T\|\sqrt{\varepsilon^*m} | Z_0, J_0\right] \\
& \leq 4\exp(-2\varepsilon^*m)
\end{aligned} \tag{C.117}$$

$$\mathbb{P}[\mathcal{E}_3] \leq (9+N)\exp(-2\varepsilon^*m) \tag{C.118}$$

where σ is defined in Section 4.1.

$\mathcal{E}_4$ (eqn. (C.94)): bounding $\|\Phi^\dagger \mathbf{z}\|$.

Lastly, the bound in eqn. (C.119) holds on $\mathcal{E}_\Phi$ (denotes the event in eqn. (C.101)).

$$\|\Phi^\dagger z\| \leq \|\Phi^\dagger\| \|z\| \leq \frac{8}{\lambda v_t} \sqrt{\frac{\alpha^* m}{1-\rho}} \leq 8\lambda v' v_{\max} \tau \sqrt{\delta m} \quad (\text{C.119})$$

where

- Φ is defined in eqn. (C.59),
- z is defined in eqn. (C.58),
- v_t is defined in **Lemma 1**,
- α^* is the short forms of $\alpha_{\delta, \rho, v_{\max}}^*$ and is defined in eqn. (4.15),
- m is defined in **Assumption 1**.
- δ and ρ are defined in **Assumption 3**,
- λ is defined in eqn. (4.8),
- $v_{\max}$ and τ are defined in eqn. (4.9),
- $v_{\min}$ is defined in eqn. (4.15), and
- v' is a constant equal to $\frac{v_{\max}}{v_{\min}}$.

The result above applies eqn. (C.81), $\alpha^* \leq \lambda^4 v_{\max}^4 \tau^2 (1-\rho)$. Hence, the probability bound in eqn. (C.120) holds.

$$\mathbb{P}[\mathcal{E}_4] \leq \mathbb{P}[\mathcal{E}_\Phi^c] \leq (3+N) \exp(-2\varepsilon^* m) \quad (\text{C.120})$$

where

- $\mathcal{E}_\Phi$ denotes the event in eqn. (C.101),
- N is defined in **Assumption 1**, and
- ε^* is the short forms of $\varepsilon_{\delta, \rho, v_{\max}}^*$ and is defined in eqn. (4.15).

Summing the failure probabilities in eqns (C.103), (C.109), (C.118), (C.120) and **Lemma 10** establishes the **Lemma 4**.

C.3.4. Proof of Lemma 5

Lemma 5 is directly proven by the following eqn. (C.121).

$$\begin{aligned}
\gamma_s([M|t_1N_1| \dots |t_nN_n]) &= \inf_{\substack{\|x\|^2 + \|y_1\|^2 + \dots + \|y_n\|^2 = 1, \\ \|x\|_0 + \|y_1\|_0 + \dots + \|y_n\|_0 \leq s}} \left\| [M \ t_1N_1 \ \dots \ t_nN_n] \begin{bmatrix} x \\ y_1 \\ \vdots \\ y_n \end{bmatrix} \right\| \\
&= \inf_{\substack{\|x\|^2 + \|y_1\|^2 + \dots + \|y_n\|^2 = 1, \\ \|x\|_0 + \|y_1\|_0 + \dots + \|y_n\|_0 \leq s}} \left\| [M \ N_1 \ \dots \ N_n] \begin{bmatrix} x \\ t_1y_1 \\ \vdots \\ t_ny_n \end{bmatrix} \right\| \\
&= \inf_{\substack{\|\tilde{t}x\|^2 + \left\|\frac{t_1}{\tilde{t}}y_1\right\|^2 + \dots + \left\|\frac{t_n}{\tilde{t}}y_n\right\|^2 = 1, \\ \|\tilde{t}x\|_0 + \left\|\frac{t_1}{\tilde{t}}y_1\right\|_0 + \dots + \left\|\frac{t_n}{\tilde{t}}y_n\right\|_0 \leq s}} \left\| [M \ N_1 \ \dots \ N_n] \begin{bmatrix} \tilde{t}x \\ \tilde{t}y_1 \\ \vdots \\ \tilde{t}y_n \end{bmatrix} \right\| \quad (C.121) \\
&\geq |\tilde{t}| \inf_{\substack{\|x\|^2 + \|y_1\|^2 + \dots + \|y_n\|^2 = 1, \\ \|x\|_0 + \|y_1\|_0 + \dots + \|y_n\|_0 \leq s}} \left\| [M \ N_1 \ \dots \ N_n] \begin{bmatrix} x \\ y_1 \\ \vdots \\ y_n \end{bmatrix} \right\| \\
&= |\tilde{t}| \gamma_s([M \ N_1 \ \dots \ N_n])
\end{aligned}$$

where $\tilde{t} = \min(1, t_1, \dots, t_n)$ and function γ_s is defined in eqn. (C.5).

C.3.5. Proof of Lemma 9

Consider the group of permutations of $[m]$ with distance stated in eqn. (C.122) in where m is defined in **Assumption 1**. Suppose π be a random permutation sampled according to the uniform measure. Then, the random variable $\|\mu_{j^c}^i\|^2$ is equal in distribution to eqn. (C.123) in where $\mu_{j^c}^i$ is defined in eqn. (4.17). As $\|\mu_i\|_\infty < C_\mu m^{-\frac{1}{2}}$, eqn. (C.124) holds in where μ_i , C_μ and m are defined in **Assumption 1**.

$$d(\pi_1, \pi_2) = m^{-1} \# \{k | \pi_1(k) \neq \pi_2(k)\} \quad (C.122)$$

$$f(\pi) = \sum_{k=1}^{(1-\rho)m} \mu_i(\pi(k))^2 \quad (C.123)$$

$$\begin{aligned}
|f(\pi_1) - f(\pi_2)| &\leq 4C_\mu^2 m^{-1} \# \{\pi_1(k) \neq \pi_2(k)\} \\
&= 4C_\mu^2 d(\pi_1, \pi_2)
\end{aligned} \quad (C.124)$$

where ρ is defined in eqn. (4.7)

Corollary 4.3 of [492] presents the probability bound in eqn. (C.125).

$$\mathbb{P}[f(\pi) \leq \mathbb{E}f(\pi) - t] \leq \exp\left(-\frac{mt^2}{2^7 C_\mu^4}\right) \quad (\text{C.125})$$

When $\mathbb{E}f(\pi)$ in eqn. (C.125) equals to $1 - \rho$, eqn. (C.13) is established.

By eqn. (4.15), the following bound in eqn. (C.126) forms.

$$\begin{aligned} \sqrt{\varepsilon^*} &\leq \lambda v_{\max} \sqrt{1 - \rho} \leq 2^{-12} \sqrt{\frac{C_s(1 - \rho)}{\delta}} \\ &\leq 2^{-12} \sqrt{2^{-17} C_\mu^{-4} \delta^{-2} (1 - \rho)^3} \\ &\leq 2^{-20} C_\mu^{-2} (1 - \rho) \end{aligned} \quad (\text{C.126})$$

where

ε^* is the short forms of $\varepsilon_{\delta, \rho, v_{\max}}^*$ and is defined in eqn. (4.15).

λ is defined in eqn. (4.8),

$v_{\max}$ is defined in eqn. (4.9),

C_s is defined in eqn. (4.10), and

δ is defined in eqn. (4.7).

and $\delta \geq 1$ is used in the last step.

Hence, by the bound in eqn. (C.127) which is derived from eqn. (C.126) and eqn. (C.13), eqn. (C.128) holds with probability at least $1 - \exp(-2\varepsilon^* m)$.

$$8C_\mu^2 \sqrt{2 \times 2\varepsilon^*} \leq 2^{-16} (1 - \rho) \ll \frac{3(1 - \rho)}{4} \quad (\text{C.127})$$

$$\|\mu_{jc}^i\|^2 \geq \frac{(1 - \rho)}{4} \quad (\text{C.128})$$

By setting event $\mathcal{E}_i$ as in eqn. (C.129) and its event probability bound $\mathbb{P}[\mathcal{E}_i]$ as in eqn. (C.130), the probability bound $\mathbb{P}[\cap_{i=1}^N \mathcal{E}_i^c]$ in eqn. (C.131) holds.

$$\mathcal{E}_i = \left\{ \|\mu_{jc}^i\|^2 \leq \frac{(1 - \rho)}{4} \right\} \quad (\text{C.129})$$

$$\mathbb{P}[\mathcal{E}_i] \leq \exp(-2\varepsilon^* m) \quad (\text{C.130})$$

$$\mathbb{P} \left[\bigcap_{i=1}^N \mathcal{E}_i^c \right] > 1 - N \exp(-2\varepsilon^* m) \quad (\text{C.131})$$

Hence, the bound in eqn. (C.14) holds with probability at most $N \exp(-2\varepsilon^* m)$.

C.3.6. Proof of Lemma 10

Let $\vartheta = \frac{\hat{\mu}_{J^c}}{N\sqrt{1-\rho}}$ in where

N is defined in **Assumption 1**,

ρ is defined in **Assumption 3**,

$\hat{\mu}_{J^c}$ is a matrix equal to $[\mu_{J^c}^1, \dots, \mu_{J^c}^N]$, and

$\mu_{J^c}^i$ is defined in **Lemma 1**.

and P_ϑ be the matrix that projects onto the span of ϑ . Then eqn. (C.132) can be formed from (C.56) and hence eqn. (C.133) hold.

$$W = \begin{bmatrix} \frac{(I - P_\vartheta)Z_{J^c,I}}{\sqrt{1-\rho}} & \vartheta \end{bmatrix} + \begin{bmatrix} \frac{P_\vartheta Z_{J^c,I}}{\sqrt{1-\rho}} & \mathbf{0} \end{bmatrix} \quad (\text{C.132})$$

$$\sigma_{\min}(W) \geq \sigma_{k_1+1} \left[\frac{(I - P_\vartheta)Z_{J^c,I}}{\sqrt{1-\rho}} \quad \vartheta \right] - \left\| \frac{P_\vartheta Z_{J^c,I}}{\sqrt{1-\rho}} \right\| \quad (\text{C.133})$$

where

k_1 is defined in Section 4.1, and

$Z_{J^c,I}$ is defined in **Lemma 1**.

For any matrix $M = [\tilde{B} \tilde{C}] \in \mathbb{R}^{p \times (\tilde{q}_1 + \tilde{q}_2)}$ with $\tilde{B} \in \mathbb{R}^{p \times \tilde{q}_1}$ and $\tilde{C} \in \mathbb{R}^{p \times \tilde{q}_2}$ and $\tilde{B}^* \tilde{C} = 0$,

and so eqn. (C.134) forms, and hence the bound in eqn. (C.135) holds.

$$\sigma_{\tilde{q}_1 + \tilde{q}_2}(M) = \min\{\sigma_{\tilde{q}_1}(\tilde{B}), \sigma_{\tilde{q}_2}(\tilde{C})\} \quad (\text{C.134})$$

$$\begin{aligned} \sigma_{\min}(W) &\geq \min \left\{ \frac{\sigma_{k_1}[(I - P_\vartheta)Z_{J^c,I}]}{\sqrt{1-\rho}}, \|\vartheta\| \right\} - \left\| \frac{P_\vartheta Z_{J^c,I}}{\sqrt{1-\rho}} \right\| \\ &\geq \min \left\{ \frac{\sigma_{k_1}[Z_{J^c,I}]}{\sqrt{1-\rho}}, \|\vartheta\| \right\} - 2 \left\| \frac{P_\vartheta Z_{J^c,I}}{\sqrt{1-\rho}} \right\| \end{aligned} \quad (\text{C.135})$$

Moreover, the events in eqns (C.136)-(C.138) are defined.

$$\mathcal{E}_1 = \left\{ \|\vartheta\| \leq \frac{1}{2} \right\} \quad (\text{C.136})$$

$$\mathcal{E}_2 = \left\{ \|P_{\vartheta} Z_{J^c, I}\| \geq \sqrt{\alpha^*} + \sqrt{\frac{1}{m}} + \sqrt{4\varepsilon^*} \right\} \quad (\text{C.137})$$

$$\mathcal{E}_3 = \{ \sigma_{k_1}[Z_{J^c, I}] \leq \sqrt{1-\rho} - \sqrt{\alpha^*} - \sqrt{4\varepsilon^*} \} \quad (\text{C.138})$$

where ε^* and α^* are the short forms of $\varepsilon_{\delta, \rho, v_{\max}}^*$ and $\alpha_{\delta, \rho, v_{\max}}^*$ and are defined in eqn. (4.15).

By **Lemma 9**, the event probability bound $\mathbb{P}[\mathcal{E}_1]$ in eqn. (C.139) holds. Furthermore, by eqn. (C.1), P_{ϑ} projects $Z_{J^c, I}$ onto an independently distributed subspace of dimension at most 1, the event probability bound $\mathbb{P}[\mathcal{E}_2]$ in eqn. (C.140) holds. Lastly, by eqn. (C.2), the event probability bound $\mathbb{P}[\mathcal{E}_3]$ in eqn. (C.141) holds. Hence, the probability bound $\mathbb{P}[\bigcap_{i=1}^3 \mathcal{E}_i^c]$ in eqn. (C.142) holds.

$$\mathbb{P}[\mathcal{E}_1] \leq N \exp(-2\varepsilon^* m) \quad (\text{C.139})$$

$$\mathbb{P}[\mathcal{E}_2] < \exp(-2\varepsilon^* m) \quad (\text{C.140})$$

$$\mathbb{P}[\mathcal{E}_3] \leq \exp(-2\varepsilon^* m) \quad (\text{C.141})$$

$$\mathbb{P} \left[\bigcap_{i=1}^3 \mathcal{E}_i^c \right] > 1 - (2 + N) \exp(-2\varepsilon^* m) \quad (\text{C.142})$$

From eqn. (4.15), the following bound in eqn. (C.143) holds, and hence the bound in eqn. (C.144) forms. Similarly, as the bound in eqn. (C.145) holds, and so the bound in eqn. (C.146) holds too. Lastly, by eqn. (4.15), the bound in (C.147) holds for $m > m^*$.

$$\sqrt{\frac{\varepsilon^*}{(1-\rho)}} \leq \lambda v_{\max} \tau < \lambda v_{\max} < 2^{-12} \quad (\text{C.143})$$

$$\sqrt{\frac{4\varepsilon^*}{(1-\rho)}} < 2^{-11} \quad (\text{C.144})$$

$$\alpha^* < \Upsilon \left[\frac{\varepsilon^*}{\delta} \right] \leq \frac{\varepsilon^*}{\delta} \leq \varepsilon^* \quad (\text{C.145})$$

$$\sqrt{\frac{\alpha^*}{(1-\rho)}} < 2^{-12} \quad (\text{C.146})$$

$$\sqrt{\frac{1}{m(1-\rho)}} \leq \sqrt{2^{-17}} < 2^{-8} \quad (\text{C.147})$$

where

- λ is defined in eqn. (4.8),
- v_{max} and τ are defined in eqn. (4.9), and
- Υ is defined in eqn. (4.12).

Hence, the bound in eqn. (C.148) holds on $\cap_i \mathcal{E}_i$. **Lemma 10** is proven.

$$\begin{aligned} \sigma_{min}(W) &\geq \min\left(\frac{1}{2}, 1 - 2^{-12} - 2^{-11}\right) \\ &\quad - 2 \times (2^{-12} + 2^{-8} + 2^{-11}) \\ &> \frac{1}{4} \end{aligned} \quad (\text{C.148})$$

C.3.7. Proof of Lemma 12

The proof can be easily derived by mimicking the proof of **Lemma 12** in [147]

Suppose $A \in \mathbb{R}^{m \times n}$, (x_o, e_o) and $(x_o(I), e_o)$ are the unique optimal solutions to eqns (C.50) and (C.51) respectively and $\text{supp}(x_o) \subseteq I$.

A solution (x, e) to $Ax + e = A_{[m],I}x_o(I) + e_o$ can be corresponded by any solution $(\acute{x}, e)$ to $A_{[m],I}\acute{x} + e = A_{[m],I}x_o(I) + e_o$ by setting $x(I) = \acute{x}$ and $x(I^c) = 0$. Hence, neither (x_o, e_o) is uniquely optimal for (C.50) nor that solution is equal to $(x_o(I), e_o)$. **Lemma 12** is proven.

C.3.8. Proof of Lemma 13

The proof can be easily derived by mimicking the proof of **Lemma 13** in [147]

Let $A \in \mathbb{R}^{m \times n}$, (x_o, e_o) is the unique optimal solution to eqn. (C.48), $\text{supp}(x_o) \subseteq I, J = \text{supp}(e_o), J' \subset J$, e'_o is constructed by letting $e'_o(J') = e_o(J')$ and $e'_o(J'^C) = 0$ and $h = e_o - e'_o$.

Suppose on the contrary, that there exists a solution $(\hat{x}, \hat{e})$ to $A\hat{x} + \hat{e} = Ax_o + e'_o$ and the following inequality in eqn. (C.149) holds.

$$\|\hat{x}\|_1 + \|\hat{e}\|_1 \leq \|x_o\|_1 + \|e'_o\|_1 \quad (\text{C.149})$$

As $A\hat{x} + \hat{e} + \hat{h} = Ax_o + e_o$, $(\hat{x}, \hat{e} + \hat{h})$ is feasible for (C.52) and the following inequality in eqn. (C.150) holds.

$$\begin{aligned} \|x_o\|_1 + \|e_o\|_1 &= \|x_o\|_1 + \|e'_o\|_1 + \|\hat{h}\|_1 \\ &\geq \|\hat{x}\|_1 + \|\hat{e}\|_1 + \|\hat{h}\|_1 \\ &\geq \|\hat{x}\|_1 + \|\hat{e} + \hat{h}\|_1 \end{aligned} \quad (\text{C.150})$$

As the optimizer to (C.52) is unique, $\hat{x} = x_o$ and $\hat{e} = e_o$ are concluded. Hence, **Lemma 13** is proven.

Appendix D (Appendix for Chapter 5)

This appendix shows the pseudo-code of the proposed dimension reduction algorithms in Chapter 5.

Algorithm D.1 shows the pseudo-code of generating a new lower-dimensional dictionary A' from an existing dictionary A for eqn. (1.2).

The inputs of the **Algorithm D.1** are

1. The original dictionary (which is going to be transferred to a lower end computational unit) A
2. Number of rows of the output new dictionary A' m' ,
3. Number of clusters N_c , and
4. The larger number of nearest neighbours from the cluster centre to be chosen in each cluster N_{nn} .

Note that the numbers of the cluster and nearest neighbours which are to be selected using the recommended way which is described in Section 5.4.3 and follows eqns (5.3) and (5.4) are used in this algorithm.

The output of the **Algorithm D.1** is the lower dimensional dictionary A' .

In **Algorithm D.1**, Step 1 initiates an array F which contains m elements. Each element is a vector with the size of $n \times 1$. Step 2 to 4 is a for loop which iteratively assigns the rows (feature vectors) of dictionary A to the elements ($F_i, i = 1, \dots, m$) of the feature array F . Step 3 assigns the i^{th} row (i^{th} feature vector) of dictionary A to i^{th} element (F_i) of array F . Step 5 clusters m feature vectors into N_c clusters and returns the N_c cluster centre vectors (in $\check{C}$) and the m cluster labels (in L) of all feature vectors, F_1 to F_m . Step 6 selects m' feature vectors, the nearest neighbours of cluster centres, and assigns them as elements of a new feature array F' . The details of the nearest neighbour selection algorithm at Step 6 will be described in the following paragraph. Step 7 initiates a matrix of size $m' \times n$. Step

8 to 10 assign all feature vectors (elements) in F' to all rows of A' . Step 9 assigns i^{th} element (f'_{ij}) of array F' to i^{th} row of dictionary A' . Step 11 returns A' from **Algorithm D.1**. The final A' is formed and the dictionary transfer is done.

Algorithm D.1 Dictionary Transfer

Inputs: the original dictionary, A ; number of rows of new dictionary A' , m' ; number of clusters, N_c ; the larger number of nearest neighbors from the cluster center be chosen in each cluster, N_{nn}

Outputs: the new low dimensional dictionary, A'

1. $F = \{F_1, \dots, F_m\}$ # Initiate an array F which contains m vectors, $F_i \in \mathbb{R}^n$, $i = 1, \dots, m$
 2. **for** $i = 1, \dots, m$ **do**
 3. $F_i = [f_{i1} \dots f_{in}]^t$ # the transpose of i^{th} row of A is assigned to F_i
 - # Function *Clustering* returns C , an array contains N_c cluster center vectors, and L , an array contains m cluster labels of all F_i
 5. $\check{C}, L = \text{Clustering}(F, N_c)$
 - # Function *Nearest Neighbor Selection*($\tilde{a}, \tilde{b}, \tilde{c}, \tilde{d}, \tilde{e}$) selects either $\tilde{d}$ or $\tilde{d} + 1$ features in $\tilde{a}$ near to each cluster center in $\tilde{b}$ to form an feature array of size $\tilde{e}$ based on cluster labels of features in $\tilde{c}$ and returns the feature array. The details of algorithm of the function is shown in **Algorithm D.2**
 6. $F' = \text{Nearest Neighbor Selection}(F, \check{C}, N_{nn}, m')$ # where $F' = \{F'_1, \dots, F'_{m'}\}$
 7. $A' = m' \times n$ matrix # Initiate a $m' \times n$ matrix
 8. **for** $i = 1, \dots, m'$ **do**
 9. $A'(i, 1, \dots, n) = F'_i$ # assigns i^{th} element of array F' to i^{th} row of dictionary A'
 10. **end for**
 12. **Return** A'
-

Algorithm D.2 shows the details of the nearest neighbour selection algorithm in Step 6 in **Algorithm D.1**.

The input of the **Algorithm D.2** are

1. The original feature set F ,
2. A cluster centre set C ,
3. A cluster label set L ,
4. The larger number of nearest neighbours from the cluster centre be chosen in each cluster N_{nn} , and
5. The number of features to be chosen m' .

The output is a new feature set F' which is a subset of F .

In **Algorithm D.2**, Step 1 assigns the number of features in F to m . Step 2 assigns the remainder of m divided by n_c to k . Step 3 assigns $\frac{m-k}{n_c} + 1$ to N_{nn} . Step 4 initiates an empty array F' . Step 5 to 19 select either N_{nn} or $N_{nn} - 1$ features which are nearest neighbours of all cluster centres and insert them into the array F' . Step 6 to 10 decides whether to select N_{nn} or $N_{nn} - 1$ from the cluster. If i less than or equal to k , N_{nn} features will be selected from the cluster and the integer N_{nn} is assigned to N'_{nn} . Otherwise, the integer $N_{nn} - 1$ features will be selected and $N_{nn} - 1$ is assigned to N'_{nn} . Step 11 assigns all the indexes of label set elements which labels equal to i in L to L_i . Step 12 assigns the number of elements in L_i into N_{L_i} . Step 13 initiates a zeros vector whose size is $N_{L_i} \times 1$. Step 14 to 16 calculates all distances between the feature vectors which labels equals to i and their corresponding i^{th} cluster centre. Step 15 assigns the distance between j^{th} feature vector and the corresponding cluster centre to j^{th} element of D . Step 17 assigns the indexes of the N'_{nn} feature vectors corresponding to the N'_{nn} smallest distances in D to D_i . Step 18 inserts the N'_{nn} feature vectors which correspond to the N'_{nn} smallest distances in D to the feature array F' . Step 20 returns the feature array F' .

Algorithm D.2 Nearest Neighbor Selection

Inputs: a feature set, F ; cluster center set, C ; cluster label set, L ; the larger number of nearest neighbors from the cluster center be chosen in each cluster, N_{nn} ; number of features to be chosen, m'

Outputs: a new feature set, F'

1. $m = N(F)$ # function $N(a)$ returns the number of elements in the input array a (e.g. # array F)
2. $k = \text{mod}(m, n_c)$ # function $\text{mod}(a, b)$ returns the remainder when a is divided by b
3. $N_{nn} = \frac{m-k}{n_c} + 1$
4. $F' = \{\}$ # Create an empty array for storing chosen features
5. **for** $i = 1, \dots, N_c$ **do**
6. **if** $i \leq k$
7. $N'_{nn} = N_{nn}$
8. **else**
9. $N'_{nn} = N_{nn} - 1$
10. **end if**
- # function $\text{Index}(a = b)$ returns an array which contains all indexes of elements which are equal to b in array a
11. $L_i = \text{Index}(L = i)$ # indexes of elements which are equal to b in array a
12. $N_{L_i} = \text{Length}(L_i)$ # function $\text{Length}(a)$ returns the number of elements in a
- # function $\text{zeros}(a, b)$ returns an array which all values are zeroes and size is $a \times b$.
13. $D = \text{zeros}(N_{L_i}, 1)$
14. **for** $j = 1, \dots, N_{L_i}$ **do**
- # $\text{Distance}(a, b)$ returns the distance between a and b . The distance function is # arbitrarily set by programmer.
15. $D(j) = \text{Distance}(F(L_i), C_i)$
16. **end if**
- # function $\text{IndexMin}(a, b)$ returns the indexes of b elements which have the smallest values among all elements in array a where b is a number.
17. $D_i = \text{IndexMin}(D, N'_{nn})$
18. $F' = F' \cup F(L_i(D_i))$ # include the features of those N'_{nn} elements into feature # array F'
19. **end for**
20. **Return** F'

Appendix E (Appendix for Chapter 6)

This appendix shows the algorithms for finding the best parameters and threshold for the two proposed approaches sparse reconstruction approach (SRA) (Section E.1) and extended SRA (Section E.2) for defect detection and localization in Chapter 6.

E.1. Algorithm for Finding Best λ , η and $\bar{T}$ for SRA

Algorithm E.1 shows one of the procedures to derive best parameters λ in eqn. (6.2) and η in eqn. (6.4) and the threshold $\bar{T}$ for the classifier in eqn. (6.3) for achieving a competitive detection performance of SRA. Given a dictionary $\mathbf{A}$, the inputs to the algorithm consist of the set of labeled training images (all pixels are labeled either defective or normal pixels), $\mathbf{I}_t$, the stopping criterion $\mathbf{S}$ for the procedure, and the search granularity $\tilde{N}$ that affects the computational time of the algorithm.

Step 1 initializes λ and η and the variables required to compute the stopping criterion, i.e., the difference between the two best accuracies in one iteration. m is the total number of pixels for each image in $\mathbf{I}_t$. The accuracy computed in **Algorithm E.1** for the stopping criterion is the balanced accuracy to be described in eqn. (6.5). When the stopping criterion is not satisfied, the algorithm repeats steps 3 to 27, the search procedure to determine the best parameters.

For steps 3, $\maxAccp$, λ_p and η_p store the best accuracy and the λ and η obtained during the previous iteration of the outer while loop. Minimum and maximum of λ domain (λ_{min} and λ_{max}) are also initiated for the while loop from steps 4 to 12 (While Loop E.1). From steps 5 to 11, The possible λ domain is narrowed down and gives the best λ by looking for the best accuracy (based on $\mathbf{I}_t$) at a fixed η based on a search in the λ domain depending on the granularity $\tilde{N}$. Steps 5 to 10 (For Loop E.1) computes all balanced accuracies with different λ . Step 7 determines the likely regions of defect based on the sparse residuals $\mathbf{e}$ which is estimated using the sparse reconstruction in eqn. (6.2) for a given $\tilde{\lambda}[i]$. Step 8 determines threshold $\tilde{T}[i]$ using optimization problem in eqn. (6.4) and fixed η . Step 9

performs the classification on the pixels of the images in I_t based on the pixel-based threshold classifier described in eqn. (6.3) using $\tilde{T}[i]$. Then, the balanced accuracy $acc1[i]$ given follows eqn. (6.5) is computed (similarly for $acc2[i]$ in step 19).

For steps 11, the indices of λ in $\tilde{\lambda}$ vector with the highest and second-highest accuracies are found and stored. The best λ to be used for steps 15 to 21 and the smaller λ domain to be searched in the next iteration of the inner while loop (steps 4 to 12) when the stopping criterion in step 4 is not satisfied are stored too.

Steps 15 to 21 follow a similar procedure as steps 5 to 11 except that the best η is to be searched. Steps 15 to 20 (For Loop E.2) computes all balanced accuracies with different η . Step 23 stores the best accuracy $maxAcc$ achieved in the iteration of the outer while loop. It is clear that the highest accuracy is derived in the procedure from steps 14 to 22 (While Loop E.2) since this procedure also uses the best parameters obtained from steps 4 to 11 for anomaly detection. It also stores the threshold used to construct the best classifier in this iteration. In steps 24 to 27, if the current best accuracy $maxAcc$ obtained from this iteration is not superior to the previous iteration, revert λ and η to the values in the previous iteration and the optimization ends. Step 28 returns the optimized λ , η and $\bar{T}$ (λ_o , η_o and $\bar{T}_o$).

E.2. Algorithms for Finding Best λ and η for Extended SRA

In the following subsections, three following algorithms for finding the best parameters, λ in (6.7) and η in (6.10) of Extended SRA are presented:

- (1). the main algorithm (**Algorithm E.2** in Section E.2.1)
- (2). an algorithm (**Algorithm E.3** in Section E.2.2) for optimizing a parameter, and
- (3). an algorithm (**Algorithm E.4** in Section E.2.3) for determining the defect detection accuracy.

Algorithm E.3 is used in **Algorithm E.2** and **Algorithm E.4** is used in **Algorithm E.3**.

Algorithm E.1 Finding Best λ , η and $\bar{T}$ of SRA**Input:** set of labeled training images, I_t ; stopping criterion, S ; search granularity, $\tilde{N}$ **Output:** λ_o , η_o and $\bar{T}_o$

```

1:  $\lambda_o = 0$ ;  $\eta_o = \frac{m}{2}$ ;  $maxAccp = 1$ ;  $maxAcc = 0$ ;  $acc1a = 1$ ;  $acc1b = 0$ ;
    $acc2a = 1$ ;  $acc2b = 0$ 
   /*Initialization*/
2: while  $\|maxAcc - maxAccp\| > S$  do
3:    $maxAccp = maxAcc$ ;  $\lambda_p = \lambda_o$ ;  $\eta_p = \eta_o$ ;  $\lambda_{min} = 0.001$ ;  $\lambda_{max} = 1000$ 
4:   while  $\|acc1a - acc1b\| > S$  do
5:     for  $i = 0$  to  $\tilde{N}$  do
6:        $\tilde{\lambda}[i] = \lambda_{min} + \frac{i}{\tilde{N}}(\lambda_{max} - \lambda_{min})$ 
7:       estimate all residual images  $e$  using
       eqn. (6.2) with  $\tilde{\lambda}[i]$  on  $I_t$ 
8:       determine  $\tilde{T}[i]$  using eqn. (6.4)
       with  $\eta_o$ 
9:       Apply eqns (6.3) and (6.5) to obtain
        $acc1[i]$  using  $\tilde{T}[i]$  on all residual
       images  $e$ 
10:    end for
11:     $k1 = \text{argmax}_i(acc1)$ ;  $k2 = \text{argmax}_i(acc1 \setminus acc1[k1])$ ;
     $acc1a = acc1[k1]$ ;  $acc1b = acc1[k2]$ ;  $\lambda_o = \tilde{\lambda}[k1]$ ;
     $\lambda_{min} = \min(\tilde{\lambda}[k1], \tilde{\lambda}[k2])$ ;  $\lambda_{max} = \max(\tilde{\lambda}[k1], \tilde{\lambda}[k2])$ 
12:  end while
13:   $\eta_{min} = 1$ ;  $\eta_{max} = m$ 
14:  while  $\|acc2a - acc2b\| > S$  do
15:    for  $j = 0$  to  $\tilde{N}$  do
16:      estimate all residual images  $e$  using
      eqn. (6.2) with  $\lambda_o$  on  $I_t$ 
17:       $\tilde{\eta}[j] = \eta_{min} + \frac{j}{\tilde{N}}(\eta_{max} - \eta_{min})$ 
18:      determine  $\tilde{T}[j]$  using eqn. (6.4)
      with  $\tilde{\eta}_j$ 
19:      Apply eqns (6.3) and (6.5) to obtain
       $acc2[j]$  using  $\tilde{T}[j]$  on all residual
      images
20:    end for
21:     $k1 = \text{argmax}_i(acc1)$ ;  $k1 = \text{argmax}_i(acc1 \setminus acc1[k1])$ ;
     $acc1a = acc1[k1]$ ;  $acc1b = acc1[k2]$ ;  $\lambda_o = \tilde{\lambda}[k1]$ ;
     $\lambda_{min} = \min(\tilde{\lambda}[k1], \tilde{\lambda}[k2])$ ;  $\lambda_{max} = \max(\tilde{\lambda}[k1], \tilde{\lambda}[k2])$ 
22:  end while

```

Loop for compute all balanced accuracies in a λ range (For Loop E.1)

Optimize λ with a fixed η (While Loop E.1)

Loop for compute all balanced accuracies in a η range (For Loop E.2)

Optimize η with a fixed λ (While Loop E.2)

Algorithm E.1 Finding Best λ , η and $\bar{T}$ of SRA (Continue)

```

23:   $maxAcc = acc2[k3]; \bar{T}_o = \tilde{T}[k3]$ 
24:  if  $maxAccp > maxAcc$  then
25:     $maxAcc = maxAccp; \lambda_o = \lambda_p; \eta_o = \eta_p$ 
26:    break
27:  end if
28: end while
29: Return  $\lambda_o, \eta_o$  and  $\bar{T}_o$ 

```

The parameter learnings are performed in a two-step greedy search described by main algorithm **Algorithm E.2**. **Algorithms E.3** and **E.4** learn λ and η respectively. The two Algorithms (Step 4 and 5 (Parameter Optimizations E.1) in **Algorithm E.2**) are iterated until either the validation accuracy converges (depends on the stopping criterion of accuracy, S_1) or the maximum number of iterations S_2 has been reached.

Note that weighted accuracy defined in eqn. (6.11) is used in these algorithms instead of the balanced accuracy defined in eqn. (6.5) in **Algorithm E.1**.

E.2.1. Main Algorithm

Algorithm E.2 is the main algorithm of the two-step greedy search for the parameter (λ and η) optimizations.

Step 1 of the algorithm initializes all the parameters and variables required for the two-step greedy search. The parameter and variables are:

- λ_o and η_o are the optimized λ and η ;
- D_l and D_u , are the lower and upper bounds of the dimensions (both width and height) of the smallest bounding box size which binds an estimated defective connected region;
- N_m is the iteration number of the main loop;
- $maxAccp$ and $maxAcc$ are the validation accuracies using previous and current optimized parameters respectively;

- λ_{min} , λ_{max} , η_{min} and η_{max} are the lower and upper bounds of λ and η for the parameter searches.

Steps 2 to 10 (While Loop E.3) is the main optimization loop. The whole loop ends when either the difference between the previous and the latest optimized accuracies is no longer larger than S_1 or the iteration number is equal to S_2 . Step 3 computes the iteration number N_m of the main loop and update the most recently optimized parameters and their corresponding accuracies. Steps 4 and 5 (Parameter Optimizations E.1) optimize λ and η respectively with **Algorithm E.3** (in Section E.2.2) by fixing the other parameter. The details of the “Parameter optimization” function is shown in Section E.2.2. Step 6 to 9 check whether the current optimized accuracy is lower than the previous one. The loop is broken if it is lower and the optimized parameters are set to be equal to the previous one. Otherwise, the loop continues with the stopping criteria stated in Step 2.

Algorithm E.2 Finding Best λ and η of Extended SRA

Input: set of labeled training images, I_t ; lower and higher dimensional dictionaries, A' and A ; stopping criteria of accuracy and number of iterations, S_1 and S_2 ; search granularity, $\tilde{N}$; kernel size S_k ; trade-off coefficient, β .

Output: λ_o and η_o

```

1:   $\eta_o = 1000$ ;  $\lambda_o = 0$ ;  $D_l = 0$ ;  $D_u = 1000$ ;  $N_m = 0$ ;  $maxAccp = 1$ ;  $maxAcc = 0$ ;
    $\lambda_{min} = 0.0001$ ;  $\lambda_{max} = 5$ ;  $\eta_{min} = 0$ ;  $\eta_{max} = 1000$ 
   /*Initialization*/
2:  while  $\|maxAcc - maxAccp\| > S_1$  or  $N_m < S_2$  do
3:     $N_m = N_m + 1$ ;  $\lambda_p = \lambda_o$ ;  $\eta_p = \eta_o$ ;  $maxAccp = maxAcc$ 
4:     $\lambda_o, maxAcc =$ 
       $ParameterOptimization$ 
       $(I_t, A', A, S_1, S_2, \tilde{N}, S_k, D_l, D_u, \beta, \lambda_{min}, \lambda_{max}, \eta_o, 1)$ 
5:     $\eta_o, maxAcc =$ 
       $ParameterOptimization$ 
       $(I_t, A', A, S_1, S_2, \tilde{N}, S_k, D_l, D_u, \beta, \lambda_{min}, \lambda_{max}, \lambda_o, 2)$ 
6:    if  $maxAccp > maxAcc$  then /*Checks accuracy*/
7:       $maxAcc = maxAccp$ ;  $\lambda_o = \lambda_p$ ;  $\eta_o = \eta_p$ 
8:    break
9:  end if
10: end while

```

Parameter
optimizations
within λ and
 η ranges
(Parameter
Optimizations
E.1)

Parameter
optimizations
for λ and η
(While Loop
E.3)

E.2.2. Algorithm for Optimizing a Parameter

Algorithm E.3 describes the details of the “*ParameterOptimization*” function which is used at Steps 4 and 5 in **Algorithm E.2**. The algorithm is used for optimizing either parameter λ or η . When a parameter is optimized, another parameter is fixed. The input P_c of the algorithm states which parameter to be optimized. $P_c = 1$ and $P_c = 2$ are for optimizing the λ and η respectively. The outputs of the algorithm are the optimal value of the learnt parameter and its corresponding weighted defect detection accuracy.

Step 1 of the algorithm initializes the iteration number N_s of the optimization loop (Parameter Optimization Loop E.1) of Step 2 to 16 and the variables $acca$ and $accb$. $acca$ and $accb$ are the highest two accuracies determined from the parameter search list. The optimization loop stops when either one of the stopping criteria, S_1 or S_2 is met. Step 3 computes the iteration number N_s . Step 4 to 14 (For Loop E.3) compute all accuracies using different parameter values defined in Step 5. The values depend on the search granularity $\tilde{N}$ and the lower and upper bounds of the parameter search range, P_{min} and P_{max} . The weighted accuracy of each training image is computed at either Step 8 or 10 (depending on which parameter is being optimized with **Algorithm E.4** (in Section E.2.3)). Step 15 updates the lower and upper bounds of the search range for the next iteration.

E.2.3. Algorithm for Determining Detection Accuracy

Algorithm E.4 shows the details of the “*AccuracyComputation*” function which is used at Steps 8 and 10 in **Algorithm E.3**. The algorithm computes the weighted defect detection defined in the eqn. (6.11).

Algorithm E.3 Parameter Optimization

Input: set of labeled training images, I_t ; lower and higher dimensional dictionaries, A' and A ; stopping criteria of accuracy and number of iterations, S_1 and S_2 ; search granularity, $\tilde{N}$; kernel size S_k ; lower and upper bound of the dimensions (weight and height) of the smallest bounding box which binds the estimated defective connected region, D_l and D_u ; trade-off coefficient, β ; lower and upper bounds of parameter search range, P_{min} and P_{max} ; value of fixed parameter, P'_o ; learnt parameter choice (1 for λ , 2 for η), P_c

Output: P_o and Acc_o

```

1:  $N_s = 0$ ;  $acca = 1$ ;  $accb = 0$ 
   /*Initialization*/
2: while  $\|acca - accb\| > S_1$  or  $N_s < S_2$ ; do
3:    $N_s = N_s + 1$ 
4:   for  $i = 0$  to  $\tilde{N}$  do
5:      $\tilde{P}[i] = P_{min} + \frac{i}{\tilde{N}}(P_{max} - P_{min})$ 
6:     for  $j = 0$  to  $N_t$  do
7:       if  $P_c = 1$  then
8:          $acc_t[j] =$ 
           AccuracyComputation
            $(I_t, j, A', A, \tilde{P}[i], P'_o, S_k, D_l, D_u, \beta)$ 
9:       else if  $P_c = 2$  then
10:         $acc_t[j] =$ 
          AccuracyComputation
           $(I_t, j, A', A, P'_o, \tilde{P}[i], S_k, D_l, D_u, \beta)$ 
11:      end if
12:    end for
13:     $acc[i] = \sum_{n=1}^{N_t} \frac{acc_t[n]}{N_t}$ 
14:  end for
15:   $k1 = \text{argmax}_i(acc)$ ;  $k2 = \text{argmax}_i(acc \setminus acc[k1])$ ;
   $acca = acc[k1]$ ;  $accb = acc[k2]$ ;  $P_o = \tilde{P}[k1]$ ;
   $P_{min} = \min(\tilde{P}[k1], \tilde{P}[k2])$ ;  $P_{max} = \max(\tilde{P}[k1], \tilde{P}[k2])$ 
16: end while
17:  $Acc_o = acc[k1]$ 
18: Return  $P_o$  and  $Acc_o$ 

```

Algorithm E.4 Accuracy Computation

Input: input/observed image, I_{in} ; lower and higher dimensional dictionaries, A' and A ; parameters, λ and η ; kernel size S_k ; lower and upper bound of the dimensions (weight and height) of the smallest bounding box which binds the estimated defective connected region, D_l and D_u ; trade-off coefficient β

Output: *Accuracy*

- 1: Find estimated sparse coefficient vector $\hat{x}$ using eqn. (6.7) for I_{in} with A' and λ
- 2: Find estimated residuals $\hat{e}$ using eqn. (6.8) with A and $\hat{x}$
- 3: Round all values in the residuals e to integers between 0 and 255 (grayscale pixel value) and to have the estimated residual image $\hat{R}$
- 4: Find the classification threshold $\bar{T}$ using eqn. (6.10) with given S_k and η
- 5: Apply eqn. (6.9) to obtain the labels of all pixels with $\bar{T}$, D_l , D_u and S_k
- 6: Calculate *Accuracy* with β using (6.11)
- 7: **Return** *Accuracy*

Appendix F A Novel Ensemble Method

The choice of features and classifiers to be used in a classification task affects its final performance. The feature is a subset of overall information in data. Different features contain different information. The information on different features can be overlapped or non-overlapped. The discriminative power of the feature for a classification task depends on the label information of the data and the distribution of the data in the feature space. The distribution of the data depends on the characteristics of information of the data and the classification problem. A classifier is used to estimate the class of data based on the location of the data in the feature space. Different classifier performs differently with different distributions of data in a specified feature space. Hence, the performance of a classification approach depends on the problem and the choices of features and classifiers.

Ensemble classifier [493] combines results of different classifiers (base classifiers) with meta-learning, learning to learn technique [494]–[497], to provide final class estimation for the data. Meta-learning has been used in training ensemble classifier, algorithm recommendation [498]–[500] and selection [501]–[506], unsupervised clustering [507], [508], hyperparameter optimization [509], label ranking [215], online learning [510] and transfer learning [511], [512]. Algorithms apply meta-learning contains a meta learner which learns how to use the metadata/features and the results of other algorithms, to perform the tasks.

Stack generalization (stacking) [513] is a way of combining results of multiple algorithms with a meta learner. The learning of stack generalization contains two layers. The first layer trains all base learners and the second layer trains a meta learner with the results of all base learners as inputs. A state-of-the-art usage of stacking is in deep learning [514], [515] (i.e. stack sparse autoencoder, stack deep recurrent neural network, etc.). The other usages of stacking are in feature fusion [516], stacked sequential learning [517], etc.

In this appendix, a relevant sensitive stacked regression ensemble approach (RSSREA) which

- is built by diverse heterogeneous base classifiers with different combinations of feature and classifier, and
- is robust to noise with the final classification estimation based on the results of the set of diverse base classifiers.

is presented. Stack generalization is used for training the classifier with a meta learner which learns how to combine the results of all base classifiers (some base classifiers perform well for the task and some are not.) to provide robust final estimation for the task.

The contributions of the meta learner are in twofold:

- self learns the adequateness (relevance) of each base classifier (a combination of a feature and a classifier) for the classification problem, and
- learns how to combine all base classifier results with their relevance measures to provide better and robust estimation than individual base classifiers.

Meanwhile, the proposed approach also provides insights to human experts regarding which features and classifiers are relevant and discriminative for the classification problem they are facing. The proposed classifier is used in a metallic defect multi-class classification problem. The results are compared against the results of a boosting and an averaging algorithm. It shows that the classification results of the proposed classifier are robust and superior to the two algorithms.

The rest of the chapter is organized as follow. Section F.1 presents related works. Section F.2 introduces the notations used in the following contents. Sections F.3 and F.4 elaborate on the procedures of training multi-class classifiers and result generations from the trained classifiers using the following approaches:

- The proposed relevant sensitive stacked regression ensemble approach (RSSREA) (Section F.3),
- Boosting based ensemble approach (Section F.4.1), and
- Averaging ensemble approach (Section F.4.2).

The second and third approaches are used in experiments for model comparisons against the proposed approach. Section F.4.3 presents the procedure of the conversion of the proposed ensemble classifiers' results into probabilities of all classes. Furthermore, rankings of the adequateness (relevance) of both features and classifiers based on training

results are described in Section F.4.4. The error bound of the classifier which is generated using the proposed approach is analyzed in Section F.4.5. Section F.5 presents the experimental results with model comparisons and feature analysis for multi-class anomaly (defect) classification problem using a real-world dataset.

F.1. Related Works

Ensemble method [518] is a machine learning approach that generates a better classifier from the results of many base classifiers which were classifier trained using a specific feature and a specific classifier model. The learning algorithm, the representation and the training data used to train each base learner can be different.

F.1.1. Boosting

Boosting [519] and bagging [520] are two standard ensemble methods. Boosting such as Adaboost [493] is a sequential ensemble method which forms a strong classifier by combining multiple selected individual weak base classifiers which have been generated sequentially. Adaboost repeats:

1. The selection for an additional weak classifier which has the best classification results among all classifiers, and
2. 2. Generate new base classifiers by emphasizing the training samples which were misclassified by the selected weak classifier

till the stop criterion (error rate is larger than 0.5) is met. Adaboost finally forms a strong classifier which is the weighted combination of all selected weak classifiers. The estimated class is the sign of the strong classifier's result for a binary classification problem.

F.1.2. Bagging

Bagging (Bootstrap AGGREGatING) is a parallel ensemble method which generates base classifiers by bootstrapping and aggregation in parallel. The motivation of bagging is to have many less dependent base classifiers. Bagging applies bootstrap sampling [521] to randomly choose different subsets of training samples for training different base classifiers. The samples in different subsets can be overlapping. The final classifier's output is the class with the major votes (majority voting) by the base classifiers.

F.1.3. Combination Methods

Combination ensemble method [522] builds a strong classifier from the combination of results of all base classifiers. The method is based on the motivation of three issues: choices of representation, classifier and training technique and the availability of training data [522]. To design a good hand-crafted representation requires extensive experience and knowledge of the domain experts. A different classifier may suit different classification problems. Robustness of individual classifier to noise is potentially low. The results of classifier learning can be at local optimal. Training data is limited either due to the cost of labelling or sparsity of training data. Hence the combination ensemble methods are to reduce the risks of all these above-mentioned issues. There are two representative averaging approaches; simple averaging and weighted averaging [518]. The results of the two approaches are the average and weighted average of all base classifiers' results respectively. Voting is another type of combination ensemble method for classification [518]. Majority voting either estimates the class of input sample as the class which have more than half of the votes of base classifiers or do not make any prediction. Plurality voting output the class which has the largest number of votes as the estimated class. The difference of weighted voting from plurality voting is that all votes of base classifiers in weighted voting are weighted.

F.1.3.1. Stacking and Meta-learning

Stacking [513], [523], [524] is a combination ensemble method whereby the results are derived from a classifier which is trained with all results of base classifiers and combines them into final classification results. In this case, the base learner is called first-level learners and the learner which learns how to combine the results of first-level learners is called the second-level learner or meta learner. The learning of meta-learner is called meta-learning. Stacked regression is a linear combination of results from the base classifiers.

F.2. Notations

Table F.1 describes the notations used in the following contents.

Table F.1 Notation Descriptions

Symbol	Descriptions
I_{tr}	Training set
I_v	Validation set
I_{te}	Test set
I_{trnb}	Training set which contains noises from mislabelling and have labels with 1 (for targeted class) and -1 (for non-targeted class) using the one-versus-rest scheme
I_{vnb}	Validation set which contains noises of mislabelling and have labels with 1 (targeted class) and -1 (non-targeted class) using the one-versus-rest scheme
I_{teb}	Test set which has labels with 1 (targeted class) and -1 (nontargeted class) using the one-versus-rest scheme
C	Set of classes
$ci \in \mathbb{R}$	Targeted class index
$N_l \in \mathbb{R}$	Noise percentage due to mislabelling
EA_c	Choice of ensemble approach: 1: RSSREA (elaborated in Section F.3), 2: Boost, a boosting based ensemble approach (elaborated in Section F.4.1), 3: AVRGE, averaging ensemble approach (elaborated in Section F.4.2))
W_{tr}	Weights for training samples
F_s	Feature set
F_p	Feature parameter set (each element contains the parameters' values of the corresponding feature in F_s with the same index)
CM_s	Classifier model set
CM_p	Classifier model parameter set (each element contains the parameters' values of the corresponding classifier model in CM_s with the same index)
FCM	Set of F_s , F_p , CM_s and CM_p
SML_c	Optimization problem choice for RSSREA (1, 2: formula eqn. (F.3) with L_1 -norm and L_2 -norm of residual vector in the objective functions respectively, 3: formula eqn. (F.4) or 4: formula eqn. (F.5))
$S_{boost} \in \mathbb{R}$	Stopping criteria of maximum step for boosting ensemble approach
BC	Trained base classifier set
$BCN \in \mathbb{R}$	Total number of base classifiers
$y \in \mathbb{R}^m$	Ground truth validation label vector
$x \in \mathbb{R}^n$	Relevant metric (weight) vector which indicates how relevant the n base classifiers (in RSSREA)
$A \in \mathbb{R}^{m \times n}$	Base classifier result matrix for validation samples (m samples) (in RSSREA). Contain n column $\tilde{A}_i$ which are the result vector (estimated soft binary labels) generated by n base classifier

$\tilde{A}_i \in \mathbb{R}^m$	Estimated soft binary label of validation samples generated by i^{th} base classifier (in RSSREA)
$a_{ij} \in \mathbb{R}$	Estimated soft binary label of j^{th} validation sample generated by i^{th} base classifier (element of A at row i and column j) (in RSSREA)
$P_{ij} \in \mathbb{R}$	Probability of j^{th} validation sample belong to the targeted class in a binary classification generated by i^{th} base classifier
$e \in \mathbb{R}^m$	Residual vector which indicates mislabeling of validation samples (in RSSREA)
$\lambda \in \mathbb{R}$	Tradeoff parameter
$\check{A} \in \mathbb{R}^{m' \times n}$	Base classifier result matrix for test samples (m' samples) (in RSSREA). Contain n column of the result vector (soft estimated binary labels) generated by n base classifier
$\hat{z} \in \mathbb{R}^{m'}$	Final resultant estimated soft test labels (the combination of all base classifiers' results) in a binary classification
$\hat{z}_{ci} \in \mathbb{R}^{m'}$	Final resultant estimated soft test labels (the combination of all base classifiers' results) in a binary classification (one versus rest) when ci^{th} class is the targeted class
$\hat{z}_{te} \in \mathbb{R}^{m'}$	Final resultant estimated test labels (the combination of all base classifiers' results) in a multi-class classification
Acc	Classification accuracy
$E_{tr} \in \mathbb{R}^{m \times n}$	Training error rate of n base classifiers on m validation samples
$BestBC_i$	Chosen i^{th} best base classifier (in Boost)
$E_{v,best} \in \mathbb{R}^m$	Training error rate of the i^{th} best base classifier on m validation samples (in Boost)
$Ind_{best} \in \mathbb{R}$	Index of the best base classifier (in Boost)
$\beta_i \in \mathbb{R}$	Contributing weight of the i^{th} best base classifier in final estimated binary soft results (in Boost)
$Z_t \in \mathbb{R}$	Normalization factor (in Boost)
$W_{AVRG} \in \mathbb{R}^n$	Weights for n base classifiers (in AVRGE)
$N \in \mathbb{R}$	Number of classes
$P_{ci} \in \mathbb{R}^{m'}$	Probabilities of ci^{th} class of m' testing samples
$p_{ci} \in \mathbb{R}^{m'}$	Probabilities of ci^{th} class of m' testing samples estimated using $\hat{z}_{ci}$
$\tilde{f}$	Feature of a sample
$p(\tilde{f}) \in \mathbb{R}$	Distribution of feature $\tilde{f}$
$\hat{z}_{ci,i}(\tilde{f}) \in \mathbb{R}$	Binary classification result on a sample's feature $\tilde{f}$ generated by i^{th} base classifier when the ci^{th} class is the targeted class
$z_{ci}(\tilde{f}) \in \mathbb{R}$	Ground truth binary label of a sample's feature $\tilde{f}$ when the ci^{th} class is the targeted class
$\hat{e}_{ci,i}(\tilde{f}) \in \mathbb{R}$	Error of binary classification result on a sample's feature $\tilde{f}$ generated by i^{th} base classifier when the ci^{th} class is the targeted class
$x_{ci,i} \in \mathbb{R}$	Relevant metric (weight) which indicates how relevant the i^{th} base classifiers when ci^{th} class is the targeted class (in RSSREA)
$\hat{e}_{ave} \in \mathbb{R}$	Weighted average square errors made by all base binary classifiers

$\hat{e}_{RSSREA,ave} \in \mathbb{R}$	Average of square errors of binary classifications with RSSREA
$\hat{e}'_{ave} \in \mathbb{R}$	Weighted average of absolute errors made by all base binary classifiers
$\hat{e}'_{RSSREA,ave} \in \mathbb{R}$	Average of absolute errors of binary classifications with RSSREA
$N_{training} \in \mathbb{R}$	Number of instances in the training set
$N_{validation} \in \mathbb{R}$	Number of instances in the validation set
$N_{test} \in \mathbb{R}$	Number of instances in the test set

F.3. Methodology

In this section, the procedures of the multi-class ensemble classifier training and result generation using the proposed RSSREA are presented as follows:

1. The classifier training procedures consist of:
 - Binary base classifier (first-level learner as well as individual binary classifier) trainings, and
 - Meta-learning of the relevant metric (adequateness) of all binary base classifiers by meta-learner (second-level learner).
2. The classification results are generated as the combination of the relevant metrics and the binary base classifiers' results for a multi-class classification problem.

F.3.1. Base Classifier (first-level learner) Training

This subsection elaborates the first step of the classifier training procedure; namely the base classifier (first-level learner) training. Base classifiers are the fundamental blocks of the ensemble approach. Each base classifier is trained using a particular feature, train set and classifier. The purpose of the base classifier training in an ensemble method is to generate multiple diversified and mutually less correlated base classifiers. All base classifiers extract different information and have different strengths in classifying the samples under different types of classification problems.

There are two categories of base classifier:

- Homogeneous base classifiers, and
- Heterogeneous base classifiers.

Homogeneous base classifiers is a set of classifiers which are trained using an identical classifier (i.e. support vector machine (SVM) or k-nearest neighbour model (k-NN) with

specific parameters, etc.) while heterogeneous base classifier is trained using multiple classifier models (i.e. SVM and k-NN model and different parameters, etc.). The classification results of all base classifiers are combined using the ensemble approach to generate the final classification results.

In the proposed RSSREA approach, different combinations of a feature (i.e. texture feature or shape feature with specific parameters, etc.) and a classifier model (i.e. SVM or k-NN model with specific parameters, etc.) are used to train multiple binary base classifiers using a common training dataset. A one-versus-rest scheme is used. A set (two-class classification problem) or multiple sets (multi-class classification problem) of heterogeneous binary base classifiers are generated. The classification results of these heterogeneous binary base classifiers will be combined with their corresponding relevant metrics (the procedure of the relevant metrics learning is elaborated in Section F.3.2) to form the final classification results (the details of how to form the final classification results is described in Section F.3.3).

Algorithm F.1 shows the procedure of the binary base classifier training. The inputs are as follows (refers to Table F.1):

- I_{trnb} (a binary labelled training set containing mislabeling),
- W_{tr} (weights for training samples), and
- FCM (a set of features, classifier model and parameters).

The output is the generated base classifier set BC . Step 1 initializes the base classifier counter, please see **Algorithm F.1**. Step 2 to 11 (For Loop F.1) is the loop to train multiple base classifiers with different combinations of a feature (in F_s (a feature set) and (For Loop F.2)) and a classifier model (in CM_s (a classifier model set) and (For Loop F.3)) with their corresponding parameters (in F_p (a feature parameter set) and CM_p (a classifier model parameter set) and (For Loop F.4) and (For Loop F.5)). Step 6 computes the index of each base classifier. Step 7 runs the main function of the base classifier training. Step 12 returns the trained base classifier set BC .

The resultant sets of binary base classifiers of all classes are further used to determine the relevant metrics for them with stacked regression-based meta-learning and validation sets. The details of the procedure of relevant metric estimation are described in the following subsection. The combinations of the estimated relevant metrics and the binary classification results are ultimately used for multi-class label estimation for test data with soft voting (this is described in Section F.3.3.2).

F.3.2. Estimation of Relevant Metric with Stacked Regression-based Meta-learning

Meta-learning is a procedure which trains a learner in combining base classifiers' (first-learners) results (meta-data) [518]. Stacked regression is a linear regression of a variable on meta-data (factors). The learning process is called stacked regression-based meta-learning.

In the proposed RSSREA approach, the classification results of a binary base classifier set on a validation dataset are the meta-data. The validation samples' ground truth labels are set to

Algorithm F.1 Base Classifier Training

Inputs: a binary labelled training set containing mislabeling, I_{trnb} ; weights for training samples, W_{tr} ; a set of F_s (a feature set), F_p (a feature parameter set), CM_s (a classifier model set) and CM_p (a set of classifier model parameter set), FCM

Note: refers Table F.1 for full terminology descriptions

Outputs: a base classifier set, BC

```

1.  $BC_n = 0$  /*Initialization*/
2. for  $i = 1$  to Size of  $F_s$  do
3.   for  $j = 1$  to Size of  $F_p\{i\}$  do
4.     for  $k = 1$  to Size of  $CM_s$  do
5.       for  $l = 1$  to Size of  $CM_p\{l\}$  do
6.          $BC_n = BC_n + 1$ 
7.          $BC\{BC_n\} =$ 
            $TrainBaseLearner$ 
            $(I_{trnb}, W_{tr}, F_s\{i\}, F_p\{i, j\},$ 
            $CM_s\{k\}, CM_p\{k, l\})$ 
8.       endfor
9.     endfor
10.  end for
11. end for
12. Return  $BC$ 

```

Loop for
training
base
classifiers
with
different
classifier
parameters
(For Loop
F.4)

Loop for
training
base
classifiers
with
different
classifier
models
(For Loop
F.3)

Loop for
training
base
classifiers
with
different
feature
parameters
(For Loop
F.2)

Loop for
training
base
classifiers
with
different
feature
and
classifier
models
(For Loop
F.1)

be the outcome variable (regressand) of the base classifiers. The weights learnt from the stacked regression-based meta-learning are the relevant metrics of all binary base classifiers (for the binary (one-versus-rest) classification problem). Note that a validation set is used in the meta-learning instead of the train set to prevent overfitting.

In a binary classification problem with labels 1 and -1, a m -dimensional ground truth validation label vector $y \in \mathbb{R}^m$ can be stacked regressed based on the base classifier's classification results X as described in eqn. (F.1)

$$y = Ax + e \quad (F.1)$$

where $A = [\tilde{A}_1, \dots, \tilde{A}_n] \in \mathbb{R}^{m \times n}$ is a $m \times n$ -dimensional matrix, $x \in \mathbb{R}^n$ is an n -dimensional weight vector and $e \in \mathbb{R}^m$ is a m -dimensional residual vector. A is the matrix which contains n columns of base classifier results where $\tilde{A}_i = (a_{i1}, a_{i2}, \dots, a_{im})^T \in \mathbb{R}^m$ is the vector containing the estimated soft labels $-1 \leq a_{ij} \leq 1$ of the validation samples

1 to m using i^{th} base classifier. If probabilistic classifiers are used, the estimated soft labels of j^{th} validation sample which is estimated by i^{th} base classifier is defined by eqn. (F.2)

$$a_{ij} = 2 \times (P_{ij} - 0.5) \quad (F.2)$$

where P_{ij} is the estimated probability of the label of j^{th} validation sample being 1 generated by i^{th} base classifier. x is the relevant metric (weight) vector which indicates how relevant each base classifier is in the classification problem. x is also a metric which shows the contribution of each base classifier's results in the estimation of soft labels Ax of all validation samples. e is the residual vector from the stacked regression.

Four optimization problems are proposed for the meta-learning of relevant metrics x . They are the following optimization problems described by the eqns. (F.3) (first and second optimization problems), (F.4) (third optimization problem) and (F.5) (forth optimization problem). The objective functions of the four optimizations are as follows:

- L_1 -norm of residual vector e ,
- L_2 -norm of residual vector e ,
- L_1 -norm of weight vector $x + \lambda L_1$ -norm of residual vector e , and
- L_1 -norm of residual vector e (contains an additional constraint of $\|x\|_1 = 1$ from others).

where λ in the third optimization problem is the trade-off parameter of L_1 -norm of weight vector x and L_1 -norm of residual vector e . All optimization problems contain the constraints of $y = Ax$ and all elements of x are positive (the estimated labels from each base classifier should not be reverted). The forth optimization problem contains an additional constraint of $\|x\|_1 = 1$ from the first optimization problem. See eqns. (F.3), (F.4) and (F.5).

$$\begin{aligned} & \min_x \|e\|_{1 \text{ or } 2} \\ & \text{s. t. (such that) } y = Ax + e \end{aligned} \quad (F.3)$$

$$\begin{aligned} & x_i \geq 0, i = 1, 2, \dots, n \\ & \min_x \|x\|_1 + \lambda \|e\|_1 \\ & \text{s. t. } y = Ax + e \\ & x_i \geq 0, i = 1, 2, \dots, n \end{aligned} \quad (F.4)$$

$$\begin{aligned}
& \min_x \|e\|_1 \\
& s. t. \quad y = Ax + e \\
& \quad \|x\|_1 = 1 \\
& \quad x_i \geq 0, i = 1, 2, \dots, n
\end{aligned} \tag{F.5}$$

The four optimization problems are suitable for different classification problems. In Section F.5, the four optimization problems are used for model comparisons based on a multi-class anomaly (defect) classification problem using a real-world industrial dataset. Note that each element in vector x is the relevant metric of the corresponding base classifier for the binary classification problem.

F.3.3. Test Set Labels Estimation

In the following, the procedures of test set label estimation are described.

F.3.3.1. Binary Classification

The soft (non-integer) estimated binary labels of a test set are estimated as $\check{A}x$ where $\check{A}$ is a $m' \times n$ -dimensional matrix which contains n columns of base classifiers' estimated test results (for the m' test sample). The ultimate test set labels can be estimated from the signs of soft labels.

Algorithm F.2 shows the procedure of estimating the soft binary test set labels using the proposed RSSREA for binary classification. It includes binary base classifier training, meta-learning of relevant metrics and soft test set labels estimation. The inputs of the algorithm are as follows (refers to Table F.1):

- I_{trnb} (a binary labelled training set containing mislabelling);
- I_{vnb} (a binary labelled validation set containing mislabelling),
- I_{teb} (a binary labelled test set),
- FCM (a set of features, classifier model and parameters), and
- SML_c (optimization problem choice).

The output is the estimated soft binary test set labels. Step 1 initializes the uniform weights W_{tr} of training samples for all base classifier trainings. Each sample has a weight of one divided by sample size of I_{trnb} . Step 2 trains all base classifiers BC using different combinations of a feature (in F_s (a feature set)), a classifier model (in CM_s (a classifier model set)) and corresponding parameters (F_p (a feature parameter set) and CM_p (a classifier model parameter set)) with the train set I_{trnb} . The base classifier training is described in details in **Algorithm F.1** (in Section F.3.1). Step 3 generates the estimated soft binary labels A for all validation samples. Step 4 optimizes (learns) the relevant metric (weight) x of each base classifier using the ground truth labels of the validation set I_{vnb} , the estimated soft binary validation labels and the chosen optimization problem assigned by SML_c using RSSREA. Step 5 generates the soft binary test set labels $\hat{z}_{BC}$ which are estimated by all base classifiers in BC . Step 6 computes the resultant estimated soft binary test set labels $\hat{z}$ as the relevant metric (x) weighted combination of $\hat{z}_{BC}$. Step 7 returns the labels $\hat{z}$.

Algorithm F.2 Binary Classification using RSSREA

Inputs: a binary labelled training set containing mislabeling, I_{trnb} ; a binary labelled validation set containing mislabeling, I_{vnb} ; a binary labelled test set, I_{teb} ; a set of F_s (a feature set), F_p (a feature parameter set), CM_s (a classifier model set) and CM_p (a set of classifier model parameter set), FCM ; optimization problem choice, SML_c

Note: refers Table F.1 for full terminology descriptions

Outputs: estimated soft binary test set labels, $\hat{z}$

1. W_{tr} = vector with values equal to one divided by sample size of I_{trnb} and size equal to sample size of I_{trnb} /*Initialization*/
 2. $BC = BaseClassifierTraining(I_{trnb}, W_{tr}, FCM)$
 3. $A = ValidationResults\_BC(I_{vnb}, BC, FCM)$
 4. $x = StackingAndMetaLearning(A, I_{vnb}, SML_c)$
 5. $\hat{z}_{BC} = TestResults\_BC(I_{teb}, x, BC, FCM)$
 6. $\hat{z} = ScoresOfTargetedDefectClass(x, \hat{z}_{BC})$
 7. **Return** $\hat{z}$
-

F.3.3.2. Multi-Class Classification

The estimated multi-class test set labels are based on the estimated soft binary test set labels (generated by base classifiers trained with a one-versus-rest scheme) which are described

in Section F.3.3.1. The estimated soft binary test set labels are the scores of the corresponding targeted classes. An estimated multi-class test set label is the targeted class which has the highest score. This scoring method is called soft voting [518]. **Algorithm F.3** shows the procedure of the multi-class classification using the proposed RSSREA. The inputs are as follows (refers to Table F.1):

- I_{tr} (a training set),
- I_v (a validation set),
- I_{te} (a test set),
- C (a set of classes),
- N_l (noise percentage due to mislabelling),
- EA_c (ensemble approach choice),
- FCM (a set of features, classifier model and parameters),
- SML_c (optimization problem choice for RSSREA), and
- S_{boost} (stopping criteria of maximum step for boosting ensemble approach).

When RSSREA is used, EA_c is set to 1 and S_{boost} does not need to be defined. SML_c is only set when RSSREA is chosen. The output is the estimated multi-class test set labels $\hat{z}_{te}$ and the test accuracy Acc . Step 1 initializes new training and validation sets, I_{trn} and I_{vn} which contain mislabelled samples which have been randomly mislabelled N_l of samples in the original sets (I_{tr} and I_v). Step 2 to 11 (For Loop F.5) estimates the soft binary labels (generated from a binary classification using one-versus-rest scheme) $\hat{z}_{ci}$ which are the scores of each defect class for all test samples. Step 3 generates binary labels for new training, validation and test set, I_{trnb} , I_{vnbn} and I_{tenbn} , using the relabelling of the samples of ci^{th} class (ci is the targeted class index) to be 1 and for the others to be 0. Step 4 to 10 estimates the soft binary labels $\hat{z}_{ci}$ for all test samples with the chosen ensemble approach (stated by EA_c) with all combinations of a feature (in F_s (a feature set)), a classifier model (in CM_s (a classifier model set)) and corresponding parameters (F_p (a feature parameter set) and CM_p (a classifier model parameter set)). Step 5, 7, 9 utilizes RSSREA using the chosen optimization problem (stated by SML_c), the boosting based ensemble approach (elaborated in Section F.4.1) and the averaging ensemble approach (elaborated in Section F.4.2) to generate the soft binary labels $\hat{z}_{ci}$. Step 12 determines the

final estimated labels $\hat{z}_{te}$ for all test samples using soft voting based on all estimated soft binary test set labels $\hat{z}_{ci}$ of different targeted classes. Step 13 computes the test accuracy Acc of the classification. Step 14 returns the final estimated labels $\hat{z}_{te}$ and the test accuracy Acc .

Algorithm F.3 Multi-Class Classification

Inputs: a training set, I_{tr} ; a validation set, I_v ; a test set, I_{te} ; a set of classes, C ; noise percentage due to mislabeling, N_l ; ensemble approach choice, EA_c ; a set of F_s (a feature set), F_p (a feature parameter set), CM_s (a classifier model set) and CM_p (a set of classifier model parameter set), FCM ; optimization problem choice for RSSREA, SML_c ; stopping criteria of maximum step for boosting ensemble approach, S_{boost} ;
 Note: refers Table F.1 for full terminology descriptions

Outputs: estimated multi-class test set labels, $\hat{L}_{te}$; test accuracy, Acc

1. I_{trn}, I_{vn} = new training and validation sets generated by changing labels of N_l of all samples in I_{tr} and I_v to random wrong labels respectively
2. **for** $ci = 1$ to Size of C **do**
3. $I_{trnb}, I_{vnb}, I_{teb}$ = new training, validation and test sets generated by changing labels of all samples of Defect class $C\{ci\}$ in I_{trnb}, I_{vnb} and I_{teb} to 1 and labels of the other samples to -1 respectively
4. **if** $EA_c = 1$ **then**
5. $\hat{z}_{ci} = BinaryClassificationUsingRSSREA(I_{trnb}, I_{vnb}, I_{teb}, FCM, SML_c)$
6. **else if** $EA_c = 2$ **then**
7. $\hat{z}_{ci} = BinaryClassificationUsingBoosting(I_{trnb}, I_{vnb}, I_{teb}, FCM, S_{boost})$
8. **else if** $EA_c = 3$ **then**
9. $\hat{z}_{ci} = BinaryClassificationUsingAveraging(I_{trnb}, I_{teb}, FCM)$
10. **end if**
11. **end for**
12. $\hat{z}_{te} = SoftVoting(\hat{z}_1, \hat{z}_2, \dots, \hat{z}_N)$ # where N is the total class number
13. $Acc = AccuracyComputation(\hat{z}_{te}, I_{te})$
14. **Return** $\hat{z}_{te}, Acc$

Loop for generating estimated binary labels for all test samples (For Loop F.5)

F.4. Boosting-based and Averaging Ensemble Approaches

In this section, two following ensemble approaches which are used for model comparisons in the experiments (in Section F.5) are introduced:

- Boosting-based ensemble approach (Section F.4.1), and
- Averaging ensemble approach (Section F.4.2).

F.4.1. Boosting-based Ensemble Approach

The boosting-based ensemble approach which is used in the experiments is an extension of Adaboost [525]. It is an approach which sequentially chooses a binary base classifier which is trained with the procedure described in Section 5.1.1. All chosen base classifiers are combined to form a resultant classifier for a binary classification problem.

F.4.1.1. Binary Classification

The binary classification training algorithm iteratively chooses the best base classifier based on the validation results and retrains all base classifiers with weight adjustments (based on the training results) of the training samples. The weights are initialized to be uniformly distributed. In the weight adjustments of training samples, the correct classified samples' weights are magnified and incorrectly classified samples' weights are minimized. The resultant binary classifier is a weighted combination of the chosen base classifiers. The details of the derivations of training sample weight adjustment function and weight function of each chosen binary classifier can be found in [518]. The stop criterion is based on either the maximum of the number of steps that has been reached or the validation result of the latest chosen best base classifier is lower than or equal to 0.5. Similarly, the validation set is used in selecting the best base classifiers to avoid overfitting.

Algorithm F.4 shows the procedure for generating binary test set labels using the boosting based ensemble approach. The inputs of the algorithm are as follows (refers to Table F.1):

- I_{trnb} (a binary labelled training set containing mislabelling);
- I_{vnb} (a binary labelled validation set containing mislabelling),
- I_{teb} (a binary labelled test set),
- FCM (a set of features, classifier model and parameters), and
- S_{boost} (stopping criteria of maximum step).

The outputs are the estimated soft binary test set labels $\hat{z}$. Step 1 initializes the uniform weights W_{tr} of training samples for all base classifier trainings. Step 2 to 13 (For Loop F.6) iteratively choose the best base classifier $BestBC_j$ from all base classifiers BC which are trained using different combinations of a feature (in F_s (a feature set)), a classifier model (in CM_s (a classifier model set)) and corresponding parameters (F_p (a feature parameter set))

and CM_p (a classifier model parameter set)) with the train set I_{trnb} and the weights W_{tr} of training samples. The weights W_{tr} are redefined in each iteration based on the training results of all base classifiers. The selection is done sequentially until the maximum number S_{boost} of selections, is reached. Step 3 trains all base learners BC with the procedure described in Section F.3.1. Step 4 computes the training error rates E_{tr} of all base classifiers on all validation samples. Step 5 computes the validation error rate of each base classifier and returns the best base classifier $BestBC_j$ (the one which has the least validation error rate), the best base classifier's validation error rate $E_{v,best}$ and its index Ind_{best} (base classifier index). Steps 6 and 7 exit the loop if the number j of selections is larger than 1 and the average validation error rate $E_{v,best}$ of the best base classifier is larger than 0.5. Step 9 computes the contributing weight β_i of the selected best base classifier in the resultant binary classifier which estimates the soft binary test set labels. Steps 10 and 11 redefine the training weights W_{tr} which will be used for the base classifier training in the following best base classifier selection according to the error in training labels estimated by the best base classifier $BestBC_j$. Step 12 normalizes W_{tr} with the sum of all elements of W_{tr} . Steps 14 to 16 (For Loop F.7) estimate the resultant soft binary test set labels $\hat{z}$ as the weighted combination of the individual soft binary test set labels estimated by all selected base classifiers $BestBC_j$. Step 17 returns the resultant estimated soft binary test set labels $\hat{z}$.

F.4.1.2. Multi-class Classification

The multi-class test set labels are estimated using soft voting (the same method as described in Section F.3.3.2) used in RSSREA. It is the targeted class with the highest score. The score of each class is the estimated soft binary test label (in $\hat{z}$ generated by **Algorithm F.4**) which has been determined using the procedure of binary classification (as described in Section F.4.1.1). The algorithm of the procedure of the multi-class classification with the boosting based ensemble approach is the **Algorithm F.3** (described in Section F.3.3.2) with the inputs of EA_c equal to 2 and S_{boost} equal to the maximum number of best base classifier selection.

Algorithm F.4 Binary Classification using Boosting

Inputs: a binary labelled training set containing mislabeling, I_{trnb} ; a binary labelled validation set containing mislabeling, I_{vnb} ; a binary labelled test set, I_{teb} ; a set of F_s (a feature set), F_p (a feature parameter set), CM_s (a classifier model set) and CM_p (a set of classifier model parameter set), FCM ; stopping criteria of maximum step, S_{boost}
 Note: refers Table F.1 for full terminology descriptions

Outputs: estimated soft binary test set labels, $\hat{z}$

1. W_{tr} = vector with values equal to one divided by sample size of I_{trnb} and size equal to sample size of I_{trnb} /*Initialization*/
2. **for** $i = 1$ to S_1 **do**
3. $BC = BaseClassifierTraining(I_{trnb}, W_{tr}, FCM)$
4. $E_{tr} = Errors\_BC(I_{trnb}, BC, FCM)$
5. $BestBC_i, E_{v,best}, Ind_{best} = Best\_BC(I_{vnb}, BC, FCM)$
6. **if** $i \neq 1$ and $E_{v,best} > 0.5$ **then**
7. **break**
8. **end if**
9. $\beta_i = \frac{1}{2} \ln \left(\frac{1-E_{v,best}}{E_{v,best}} \right)$
10. $W_{tr}[E_{tr}[Ind_{best}] = 0] = W_{tr}[E_{tr}[Ind_{best}] = 0] \times \exp(-\beta_i)$
11. $W_{tr}[E_{tr}[Ind_{best}] \neq 0] = W_{tr}[E_{tr}[Ind_{best}] \neq 0] \times \exp(\beta_i)$
12. $W_{tr} = \frac{W_{tr}}{Z_t}$ # Z_t is a normalization factor which enables W_{tr} to # be a distribution.
13. **end for**
14. **for** $j = 1$ to Size of I_{teb} **do**
15. $\hat{z}[k] = \sum_{i=1}^i \beta_i BestBC_j(I_{teb}[k])$
16. **end for**
17. **Return** $\hat{z}$

Loop for
choosing the
best binary
classifiers
using boosting
based ensemble
approach (For
Loop F.6)

Loop for generating estimated binary labels for all
testing samples (For Loop F.7)

F.4.2. Averaging Ensemble Approach

Averaging ensemble approach is one of the simplest ensemble approaches. It combines all base classifier results using uniform weights.

F.4.2.1. Binary Classification

The approach generates the estimated soft binary test set labels as the equal-weighted average of all the soft labels which that have been estimated by all base classifiers. The base classifier training is the same procedure which is described in Section F.3.1.

Algorithm F.5 shows the procedure of estimating the soft binary test set label estimation using the averaging ensemble approach. The inputs of the algorithm are as follows (refers to Table F.1):

- I_{trnb} (a binary labelled training set containing mislabelling);
- I_{teb} (a binary labelled test set), and
- FCM (a set of features, classifier model and parameters).

The outputs are the estimated soft binary labels $\hat{z}$ of all test samples. Step 1 initializes the uniform weights W_{tr} of training samples for all base classifier trainings. Step 2 initializes the uniform weights W_{tr} of base classifiers. Step 3 trains all base classifiers BC using a different combination of a feature (in F_s (a feature set)), a classifier model (in CM_s (a classifier model set)) and corresponding parameters (F_p (a feature parameter set) and CM_p (a classifier model parameter set)) with the train set I_{trnb} . Step 4 estimates all individual estimated soft binary labels $\check{A}$ of all test samples using all base classifiers BC which are trained in Step 3. Step 5 computes the resultant estimated soft binary test set labels as the equal-weighted combination of results $\check{A}$ of all individual base classifiers BC . Step 6 returns the resultant estimated soft binary test set labels $\hat{z}$.

Algorithm F.5 Binary Classification using Averaging

Inputs: a binary labelled training set containing mislabeling, I_{trnb} ; a binary labelled test set, I_{teb} ; a set of F_s (a feature set), F_p (a feature parameter set), CM_s (a classifier model set) and CM_p (a set of classifier model parameter set), FCM ;

Note: refers Table F.1 for full terminology descriptions

Outputs: estimated soft binary test set labels, $\hat{z}$

1. W_{tr} = vector with values equal to one divided by sample size of I_{trnb} and size equal to sample size of I_{trnb} /*Initialization*/
 2. W_{AVRG} = vector with values equal to one divided by n (number of base classifier) and size equal to n /*Initialization*/
 3. $BC = BaseClassifierTraining(I_{trnb}, W_{tr}, FCM)$
 4. $\check{A} = TestResults\_BC(I_{teb}, BC, FCM)$
 5. $\hat{z} = ScoresOfTargetedDefectClass(x, \check{A})$
 6. **Return** $\hat{z}$
-

F.4.2.2. Multi-class Classification

Similar to boosting based ensemble approach and RSSREA, the multi-class test set labels are estimated using soft voting (as described in Section F.3.3.2). The targeted class with the highest score is the resultant estimated test label of that testing sample. Similarly, the score of each class is the estimated soft binary test set label (in $\hat{z}$ generated by **Algorithm F.5**) which is determined using the procedure of binary classification (as described in Section F.4.2.1). The algorithm of the procedure for the multi-class classification with the average ensemble approach is shown in **Algorithm F.3** (as described in Section F.3.3.2) with the input of EA_c equal to 3.

F.4.3. Probability of Each Class

In this subsection, the procedure of conversion of the score (estimated soft binary test set label) of a targeted class to a probability of that targeted class for a testing sample is presented. The conversion is shown in the following formula: see eqn. (F.6)

$$\begin{aligned} P_{ci}[k] &= \frac{p_{ci}[k]}{\sum_{ci=1}^N p_{ci}[k]} \\ p_{ci}[k] &= \frac{\hat{z}_{ci}[k] + 1}{2} \end{aligned} \tag{F.6}$$

where $P_{ci}[k]$ is the probability of ci^{th} class of k^{th} testing sample, $\hat{z}_{ci}[k]$ is the score (estimated soft binary test set label) of ci^{th} class of the k^{th} testing sample and N is the total number of classes. Note that, the probabilities can be used to rank all classes for the testing sample.

F.4.4. Feature and Classifier Rankings based on Adequateness

The RSSREA and boosting based approach can be used to indicate the adequateness (relevance) of features and classifiers for a particular classification problem (when conducting enough independent experiments). An exemplar ranking procedure is as follows. The adequateness of each feature is ranked by the statistics of the number of occurrences of the feature where the base classifiers are highly weighted (with set a threshold such as 95% percentile of all weights) in the resultant binary classifiers. Similarly, the adequateness of each classifier can be also ranked with similar statistics based on the number of occurrences of the classifier.

F.4.5. Error Bound Analysis

Assuming the distribution of a sample's feature $\tilde{f}$ is $p(\tilde{f})$ and the binary classification result of i^{th} base classifier on the sample's feature is $\hat{z}_{ci,i}(\tilde{f})$ when ci^{th} class is the targeted class, the following equation shows the relationship between the ground truth binary label $z_{ci}(\tilde{f})$ and the error $\hat{e}_{ci,i}(\tilde{f})$ of the base classifier, see eqn. (F.7):

$$\hat{z}_{ci,i}(\tilde{f}) = z_{ci}(\tilde{f}) + \hat{e}_{ci,i}(\tilde{f}) \quad (F.7)$$

The weighted average square errors made by all base classifiers can be expressed as shown in eqn. (F.8):

$$\hat{e}_{ave} = \sum_{i=1}^{BCN} x_{ci,i} \int \hat{e}_{ci,i}^2(\tilde{f}) p(\tilde{f}) d\tilde{f} \quad (F.8)$$

where BCN is the total number of base classifiers and $x_{ci,i}$ is the relevant metric of i^{th} base classifier.

The average of square errors of binary classifications with RSSREA is as shown in eqn. (F.9):

$$\hat{e}_{RSSREA,ave} = \int \left(\sum_{i=1}^{BCN} x_{ci,i} \hat{e}_{ci,i}(\tilde{f}) \right)^2 p(\tilde{f}) d\tilde{f} \quad (F.9)$$

Hence, inequality in eqn. (F.10) holds.

$$\hat{e}_{RSSREA,ave} \leq \hat{e}_{ave} \quad (F.10)$$

The weighted average of absolute errors made by all base classifiers and the average of absolute errors of binary classifications with RSSREA are as shown in eqns. (F.11) and (F.12):

$$\hat{e}'_{ave} = \sum_{i=1}^{BCN} x_{ci,i} \int |\hat{e}_{ci,i}(\tilde{f})| p(\tilde{f}) d\tilde{f} \quad (F.11)$$

$$\hat{e}'_{RSSREA,ave} = \int \left| \sum_{i=1}^{BCN} x_{ci,i} \hat{e}_{ci,i}(\tilde{f}) \right| p(\tilde{f}) d\tilde{f} \quad (F.12)$$

Similarly,

$$\hat{e}'_{RSSREA,ave} \leq \hat{e}'_{ave} \quad (F.13)$$

Hence, average errors (both weighted average of square error and absolute error) of RSSREA is bounded by the weighted average errors of all base classifiers.

F.5. Experimental Results

F.5.1. Metallic Surface Defect Dataset

A real-world dataset which is a subset of a large number of images collected from an AVI process for quality control in a metallic object manufacturing process¹¹ is used. The anomalies that are being identified are defects on metallic surfaces. The dataset consists of 300 images and 30 instances with various resolutions; with widths and heights varying in the range of 34 and 658 pixels. Each instance belongs to one of the 4 different classes; namely: Melt (3 instances), Plus Metal (4), Scuff (18) and Shadowing (5). Each instance has 10 images acquired under the same viewpoint and different illumination conditions. Figure F.1 shows the exemplar images of each defect class.

¹¹Due to confidentiality, the description is limited.

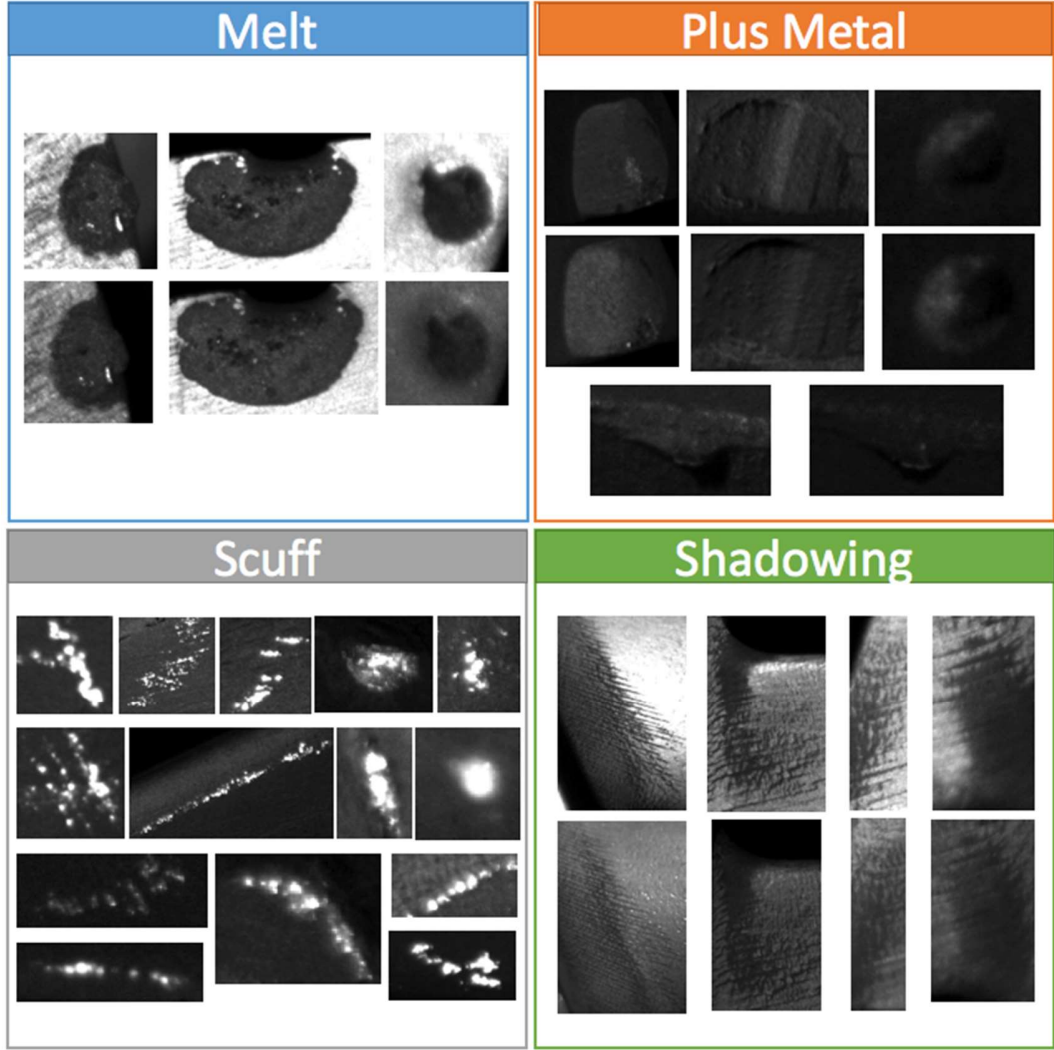


Figure F.1 Exemplar images of 4 classes (Melt, Plus Metal, Scuff and Shadowing) in the experimental metallic surface defect dataset

F.5.2. Experiment Design

In each experiment, the whole image dataset is split into three sets, training, validation and test sets. The three sets are formed using three rules:

1. Each set has as much similar number of instances as possible with other sets,
2. Each instance is only used in a set, and
3. $N_{training} \geq N_{validation} \geq N_{test}$ where $N_{training}$, $N_{validation}$ and N_{test} are the number of instances in training, validation and test sets.

100 independently randomized combination sets (an independent set contains a train set, a validation set and a test set) are used. Each combination of the three sets is used in an experiment. The overall results are based on the results of the 100 experiments.

For performance evaluation, the average of the testing accuracies of the 100 experiments is used.

F.5.3. Model Comparisons With and Without Mislabelling

The five following algorithms are used in model comparisons:

- SML1: RSSREA (Section F.3) with optimization problem eqn. (F.3) which contains the objective function of L_2 -norm of residual vector e ,
- SML2: RSSREA with optimization problem eqn. (F.3) which contains the objective function of L_1 -norm of residual vector e ,
- SML3: RSSREA with optimization problem eqn. (F.4) with $\lambda = 1$,
- SML4: RSSREA with optimization problem eqn. (F.5),
- Boost: Boosting based ensemble approach (Section F.4.1), and
- AVR: Averaging ensemble approach (Section F.4.2).

are tested in 100 independent experiments. The specific percentage of samples' labels in training and validation sets are randomly changed to the labels of the other classes as mislabelling. The purpose of mislabelling is to investigate the robustness of all ensemble approaches.

Different combinations of the following features and classifiers are used for training base classifiers:

1. The feature types used in the experiments are the following:
 - Local Binary Pattern (LBP) [526]
 - Multi-fractal Spectrum (MFS) [527],
 - "GIST of Scene" (GIST) [528], [529], and
 - Spectral Feature (SPEC) [530].

The features used in the experiments are the 4 features with a different combination of parameters.

2. The classifiers used in the experiment are support vector machines (SVM) [531] with different parameter values ($C = 0.1; 1; 10; 100; 1000$).

There are 345 combinations (including the different combinations of parameters) of a feature and a classifier. Figure F.2 shows the average accuracies of each algorithm versus the mislabelling percentages.

The results in Figure F.2 show that all algorithms with RSSREA and boosting based ensemble approach have similar performances in classification with 0% mislabelling. When the mislabelling percentage increases, SML3 and Boost performances decrease dramatically. It shows that these two approaches are much more sensitive to mislabelling than others. The poor performance of Boost may be due to the poor trainings of classifiers caused by the increases in weights of wrongly classified mislabelled samples (noises) in the latter training stage. The poor performance of SML3 may be due to the inappropriate sparse enforcement (L_1 -norm) of weights or/and the inadequate value of λ which are not suitable for the classification with noises. SML2 always outperforms SML4 in term of average accuracies when the mislabelling percentage is low. It implies that restriction of the absolute sum of weights to be equal to 1 does not improve the robustness of base classifier trainings. The performances of SML1 and SML2 do not have significant difference and have better performance than the other approaches in experiments which mislabelling percentage are less than 50%.

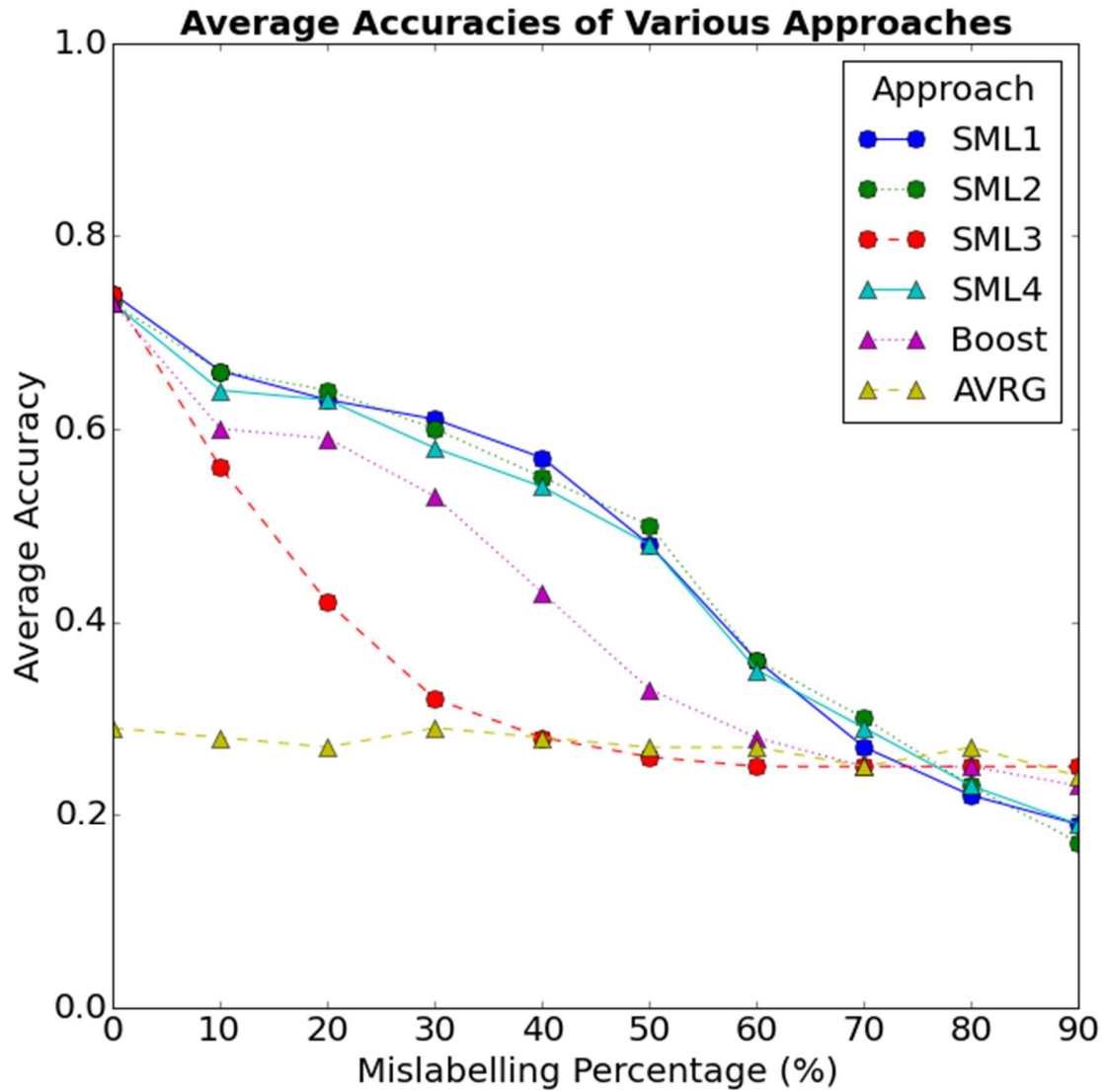


Figure F.2 Average accuracies of different ensemble approaches with different mislabelling percentages

F.5.4. Feature Analysis

In feature analysis, the two best-proposed approaches (SML1 and SML2) are compared with Boost.

The numbers behind the feature names are corresponding parameter values used in extracting those features.

Figures Figure F.3-Figure F.5 show the feature histograms of SML1, SML2 and Boost in 100 binary classification experiments which contain mislabelling percentage of 0%, 20% and 40% respectively. Each chart contains a feature histogram of a binary classification problem using a particular ensemble approach (SML1, SML2 or Boost). In each histogram, the sum of histogram values of all selected features is equal to 1. The numbers behind the feature names are corresponding parameter values used in extracting those features. The features shown in the feature histograms are the top 5 input features which binary classifiers have the highest frequencies in the binary classifier sets which have highest weights and weighted sum more than 95% of total weightage in the ensemble classifiers which are trained from 100 independent binary classification experiments. The value in the y-axis is the frequency ratios.

The construction procedure of feature histograms of SML1 and SML2 is as follows:

1. In each binary classification experiment, the features which have the highest weight and weightage sum that is more than 95% of total weightage are chosen.
2. Compute the chosen feature frequencies for each type of feature.
3. Select the top 5 features which have the highest frequencies to construct the frequency histogram. The sum of histogram values of all selected features is equal to 1.

The construction procedure of feature histogram of boosting is as follows:

1. In each binary classification experiment, all features used in the boosting based ensemble classifier are chosen.
2. Compute the chosen feature frequencies for each type of feature.
3. Select the top 5 features which have the highest frequencies to construct the frequency histogram. The sum of histogram values of all selected features is equal to 1.

Overall, Figures Figure F.3-Figure F.5 show that SPEC, MFS and LBP are the features which are frequently chosen in most histograms. In the histograms of Boost, only the histogram of Melt constructed from experiments which contain noise percentage of 0%

contains 5 feature types and features of GIST. The other histograms of Boost contain only 4 features and do not contain any feature of GIST. Hence, the more robust results of SML1 and SML2 in experiments which have higher noise levels may due to the use of many features or/and the use of GIST features. Hence, it may imply that the increase in the diversity of feature in constructing ensemble classifiers is the key to increase the robustness of the classification results.

F.5.5. Nemenyi Post-hoc Test

In this subsection, Nemenyi post-hoc test [532] is used for testing whether the results of either SML1 or SML2 and the results of Boost are significantly different. In the testing, the value of $\alpha = 0.1$ is used. The testing results show that SML1 and Boost results are significantly different when noise percentage in the 100 experiments are more than or equal to 0.2 while SML2 and Boost results are significantly different when noise percentage in the 100 experiments are more than or equal to 0.1.

F.6. Summary

In this chapter, a proposed approach; namely: relevant sensitive stacked regression ensemble approach (RSSREA), which involves learning the adequateness (relevance) of various combinations of feature and classifier is proposed for multi-class anomaly classification. The novelties of RSSREA are that it

- self-learns the adequateness (relevance) of a different combination of a feature as well as a classifier for a specific multi-class classification problem,
- estimates the class of input data by combining individual classification results of the different combination of feature and classifier with the learnt adequateness, and
- provides human experts with insights on the relevance of different features and classifiers for a specific problem.

Empirical results show that the proposed approach outperforms a boosting based ensemble approach and average ensemble approach in terms of robustness to noises and accuracy in an industrial multi-class anomaly (defect) classification problem with a real-world dataset.

Binary Classification Problem with Mislabeling of 0%

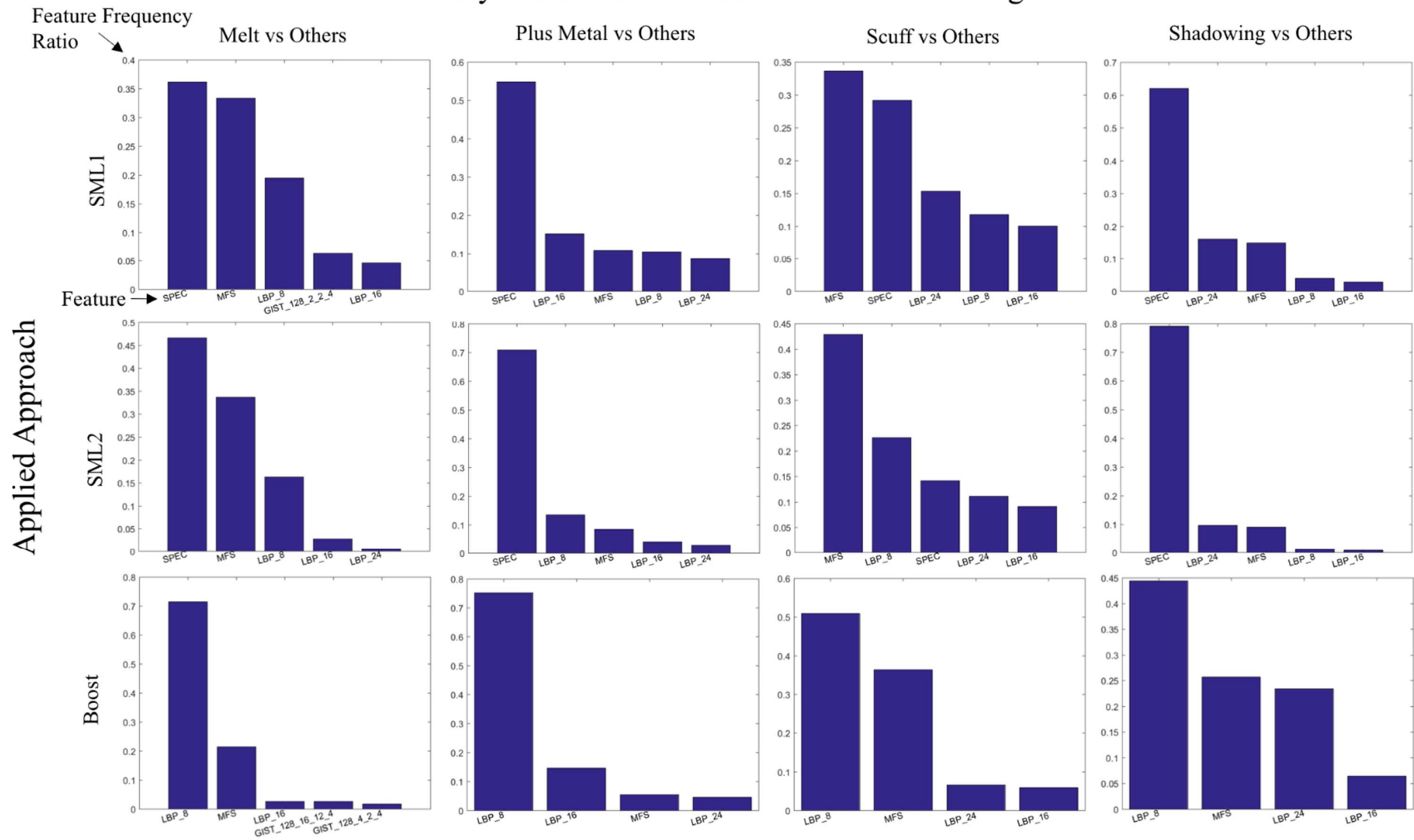


Figure F.3 Charts of feature histograms of a binary classification problem mislabelling percentage of 0%

Binary Classification Problem with Mislabeling of 20%

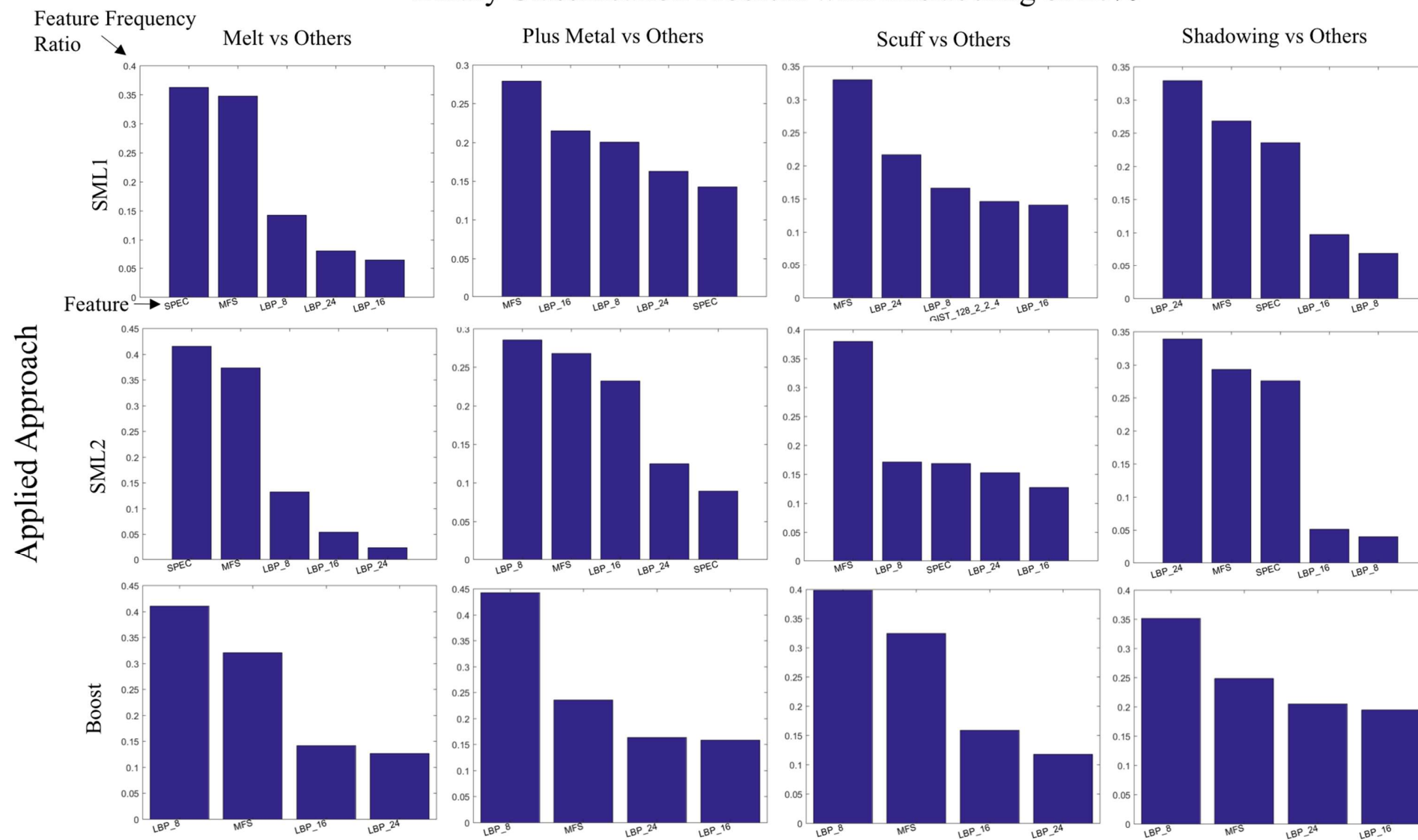


Figure F.4 Charts of feature histograms of a binary classification problem mislabelling percentage of 20%

Binary Classification Problem with Mislabeling of 40%

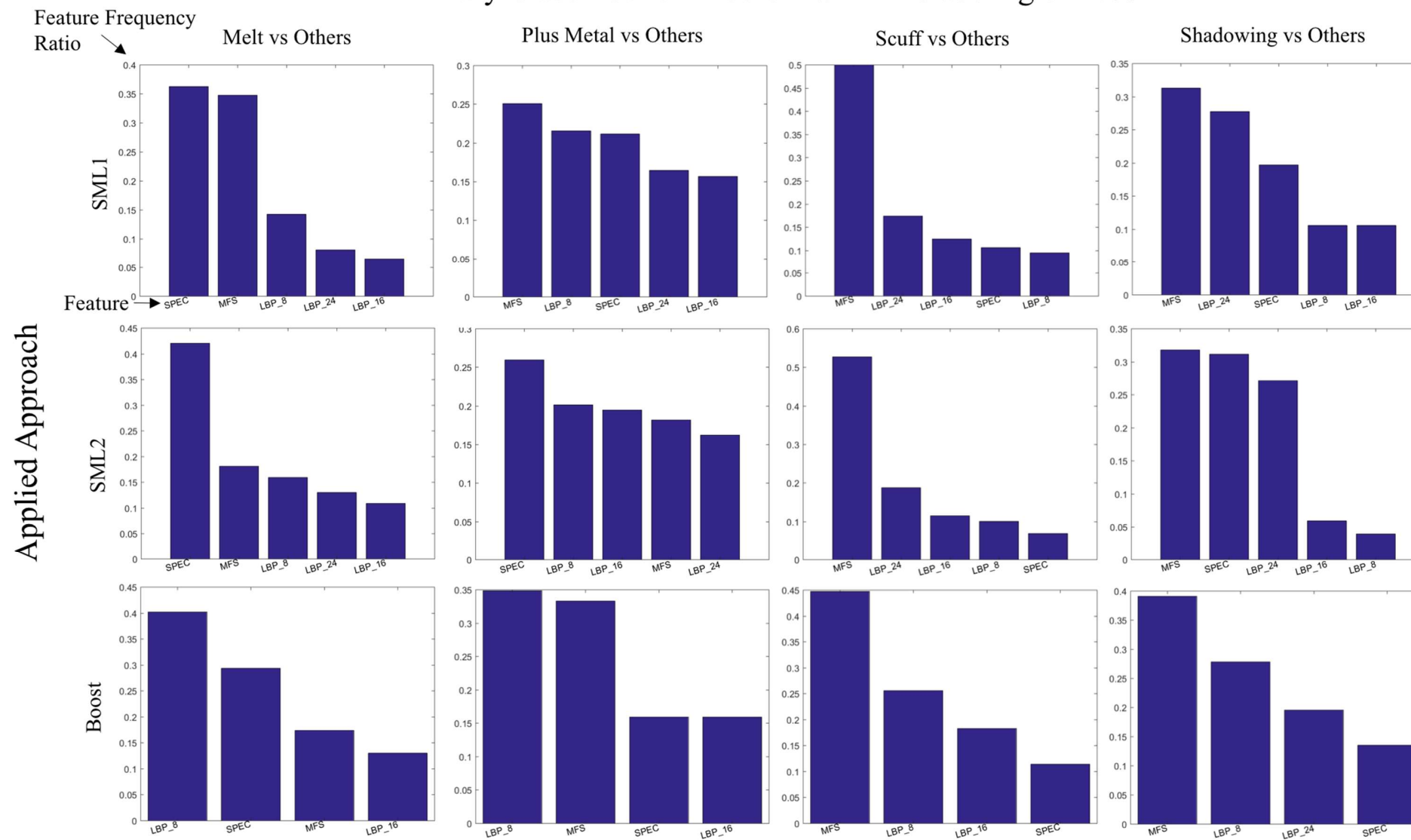


Figure F.5 Charts of feature histograms of a binary classification problem mislabelling percentage of 40%

References

- [1] L. Shen, W. Liu, J. Huang, Y.-G. Jiang, and S. Ma, “Adaptive Proximal Average Approximation for Composite Convex Minimization,” in *AAAI*, 2017, pp. 2513–2519.
- [2] Z. Zhang, Y. Xu, J. Yang, X. Li, and D. Zhang, “A Survey of Sparse Representation: Algorithms and Applications,” *IEEE Access*, vol. 3, pp. 490–530, 2015, doi: 10.1109/ACCESS.2015.2430359.
- [3] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Yi Ma, “Robust Face Recognition via Sparse Representation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009, doi: 10.1109/TPAMI.2008.79.
- [4] J. Wright and Y. Ma, “Dense error correction via l_1 -minimization,” in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, Apr. 2009, pp. 3033–3036, doi: 10.1109/ICASSP.2009.4960263.
- [5] W. H. Chai, S.-S. Ho, and C.-K. Goh, “Exploiting sparsity for image-based object surface anomaly detection,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2016, pp. 1986–1990, doi: 10.1109/ICASSP.2016.7472024.
- [6] W. H. Chai, S.-S. Ho, C.-K. Goh, L.-T. Chia, and H. C. Quek, “A fast sparse reconstruction approach for high resolution image-based object surface anomaly detection,” in *2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA)*, May 2017, pp. 13–16, doi: 10.23919/MVA.2017.7986761.
- [7] L. Shen, W. Liu, G. Yuan, and S. Ma, “ $\{\text{GSOS}\}$: $\{\text{G}\}$ auss- $\{\text{S}\}$ eidel Operator Splitting Algorithm for Multi-Term Nonsmooth Convex Composite Optimization,” in *Proceedings of the 34th International Conference on Machine Learning*, 2017, vol. 70, pp. 3125–3134, [Online]. Available: <http://proceedings.mlr.press/v70/shen17b.html>.
- [8] Y. Nesterov, “Smooth minimization of non-smooth functions,” *Math. Program.*, vol. 103, no. 1, pp. 127–152, May 2005, doi: 10.1007/s10107-004-0552-5.
- [9] V. Pappas, Y. Romano, and M. Elad, “Convolutional neural networks analyzed via convolutional sparse coding,” *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 2887–2938, 2017.
- [10] M. Elad, “Sparse Modeling,” 2018.
- [11] B. K. Natarajan, “Sparse approximate solutions to linear systems,” *SIAM J. Comput.*, vol.

- 24, no. 2, pp. 227–234, 1995.
- [12] M. Huang *et al.*, “Brain extraction based on locally linear representation-based classification,” *Neuroimage*, vol. 92, pp. 322–339, 2014.
 - [13] X. Tang, S. Wu, R. Dong, and G. Xia, “Application of compressed sensing theory in the sampling and reconstruction of speech signals,” in *2nd International Forum on Management, Education and Information Technology Application (IFMEITA 2017)*, 2018.
 - [14] Z. Qin, J. Fan, Y. Liu, Y. Gao, and G. Y. Li, “Sparse representation for wireless communications: A compressive sensing approach,” *IEEE Signal Process. Mag.*, vol. 35, no. 3, pp. 40–58, 2018.
 - [15] T. M. Quan, T. Nguyen-Duc, and W.-K. Jeong, “Compressed sensing MRI reconstruction using a generative adversarial network with a cyclic loss,” *IEEE Trans. Med. Imaging*, vol. 37, no. 6, pp. 1488–1497, 2018.
 - [16] Z. Gao, L. Dai, S. Han, I. Chih-Lin, Z. Wang, and L. Hanzo, “Compressive sensing techniques for next-generation wireless communications,” *IEEE Wirel. Commun.*, vol. 25, no. 3, pp. 144–153, 2018.
 - [17] L. Wan, G. Han, L. Shu, and N. Feng, “The critical patients localization algorithm using sparse representation for mixed signals in emergency healthcare system,” *IEEE Syst. J.*, vol. 12, no. 1, pp. 52–63, 2018.
 - [18] S. Raj and K. C. Ray, “Sparse representation of ECG signals for automated recognition of cardiac arrhythmias,” *Expert Syst. Appl.*, vol. 105, pp. 49–64, 2018.
 - [19] W. Lu, F. Cai, P. Sheridan, and C. Du, “Sparse coding with Memristor networks.” 2019.
 - [20] R. Fuentes, C. Mineo, S. G. Pierce, K. Worden, and E. J. Cross, “A probabilistic compressive sensing framework with applications to ultrasound signal processing,” *Mech. Syst. Signal Process.*, vol. 117, pp. 383–402, 2019.
 - [21] A. S. Lalos, E. Vlachos, G. Arvanitis, K. Moustakas, and K. Berberidis, “Signal Processing on Static and Dynamic 3D Meshes: Sparse Representations and Applications,” *IEEE Access*, 2019.
 - [22] Y. Xu, Z. Wu, J. Chanussot, and Z. Wei, “Nonlocal Patch Tensor Sparse Representation for Hyperspectral Image Super-Resolution,” *IEEE Trans. Image Process.*, 2019.
 - [23] D. Zou, X. Chen, G. Cao, and X. Wang, “Unsupervised Video Matting via Sparse and Low-Rank Representation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019.

-
- [24] R. Abiantun, F. Juefei-Xu, U. Prabhu, and M. Savvides, “SSR2: Sparse Signal Recovery for Single-Image Super-Resolution on Faces with Extreme Low Resolutions,” *Pattern Recognit.*, 2019.
 - [25] H. Zhang, Y. Zhang, H. Wang, Y.-S. Ho, and S. Feng, “WLDISR: Weighted Local Sparse Representation-Based Depth Image Super-Resolution for 3D Video System,” *IEEE Trans. Image Process.*, vol. 28, no. 2, pp. 561–576, 2019.
 - [26] S. Liu, G. Zhang, and W. Liu, “Group Sparse Representation based Dictionary Learning for SAR Image Despeckling,” *IEEE Access*, 2019.
 - [27] S. Shazeeda and B. A. Rosdi, “Nearest Centroid Neighbor Based Sparse Representation Classification for Finger Vein Recognition,” *IEEE Access*, vol. 7, pp. 5874–5885, 2019.
 - [28] E. N. Zois, D. Tsourounis, I. Theodorakopoulos, A. L. Kesidis, and G. Economou, “A Comprehensive Study of Sparse Representation Techniques for Offline Signature Verification,” *IEEE Trans. Biometrics, Behav. Identity Sci.*, 2019.
 - [29] C.-Y. Chiou, W.-C. Wang, S.-C. Lu, C.-R. Huang, P.-C. Chung, and Y.-Y. Lai, “Driver Monitoring Using Sparse Representation With Part-Based Temporal Face Descriptors,” *IEEE Trans. Intell. Transp. Syst.*, 2019.
 - [30] L. Bai, J. Han, and J. Yue, “Night-Vision Data Classification Based on Sparse Representation and Random Subspace,” in *Night Vision Processing and Understanding*, Springer, 2019, pp. 127–173.
 - [31] J. Xu, W. An, L. Zhang, and D. Zhang, “Sparse, collaborative, or nonnegative representation: Which helps pattern classification?,” *Pattern Recognit.*, vol. 88, pp. 679–688, 2019.
 - [32] C. Peng, X. Gao, N. Wang, and J. Li, “Sparse graphical representation based discriminant analysis for heterogeneous face recognition,” *Signal Processing*, vol. 156, pp. 46–61, 2019.
 - [33] J. Lu, V. E. Liong, and J. Zhou, “Simultaneous local binary feature learning and encoding for homogeneous and heterogeneous face recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 8, pp. 1979–1993, 2018.
 - [34] R. Yu, L. Qiao, M. Chen, S.-W. Lee, X. Fei, and D. Shen, “Weighted graph regularized sparse brain network construction for MCI identification,” *Pattern Recognit.*, vol. 90, pp. 220–231, 2019.

-
- [35] G.-J. Peng, "Adaptive ADMM for Dictionary Learning in Convolutional Sparse Representation," *IEEE Trans. Image Process.*, 2019.
- [36] S. Kumar, V. Dhiman, P. A. Koch, and J. J. Corso, "Learning Compositional Sparse Bimodal Models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1032–1044, 2018.
- [37] Y. Ding and B. D. Rao, "Dictionary learning-based sparse channel representation and estimation for FDD massive MIMO systems," *IEEE Trans. Wirel. Commun.*, vol. 17, no. 8, pp. 5437–5451, 2018.
- [38] B. Yang, A. J. Ma, and P. C. Yuen, "Learning domain-shared group-sparse representation for unsupervised domain adaptation," *Pattern Recognit.*, vol. 81, pp. 615–632, 2018.
- [39] S. Yi, Z. He, Y.-M. Cheung, and W.-S. Chen, "Unified sparse subspace learning via self-contained regression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 10, pp. 2537–2550, 2018.
- [40] C. Lu, J. Shi, and J. Jia, "Scale adaptive dictionary learning," *Image Process. IEEE Trans.*, vol. 23, no. 2, pp. 837–847, 2014.
- [41] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. theory*, vol. 52, no. 4, pp. 1289–1306, 2006.
- [42] R. G. Baraniuk, "Compressive sensing [lecture notes]," *IEEE Signal Process. Mag.*, vol. 24, no. 4, pp. 118–121, 2007.
- [43] E. J. Candes, J. Romberg, and T. Tao, "Robust Uncertainty Principles : Exact Signal Frequency Information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, 2006, doi: 10.1109/TIT.2005.862083.
- [44] E. C. Marques, N. Maciel, L. Naviner, H. Cai, and J. Yang, "A Review of Sparse Recovery Algorithms," *IEEE Access*, vol. 7, pp. 1300–1322, 2019.
- [45] J. Cao, S. Liu, H. Liu, X. Tan, and X. Zhou, "Sparse representation of classified patches for CS-MRI reconstruction," *Neurocomputing*, 2019.
- [46] Z. Liu, D. Jiang, Y. Li, Y. Cao, M. Wang, and Y. Xu, "Automatic Face Recognition Based on Sparse Representation and Extended Transfer Learning," *IEEE Access*, vol. 7, pp. 2387–2395, 2019.
- [47] Y. Li, F. Dai, X. Cheng, L. Xu, and G. Gui, "Nonconvex Adaptive Weighted Lp Regularization for Compressive Sensing Image Reconstruction Using Multiple-

- prespecified-dictionary Sparse Representation,” *J. Franklin Inst.*, 2019.
- [48] Y. Zhou and H. Guo, “Collaborative block compressed sensing reconstruction with dual-domain sparse representation,” *Inf. Sci. (Ny)*, vol. 472, pp. 77–93, 2019.
- [49] J. Xue, Y. Zhao, W. Liao, and J. C.-W. Chan, “Nonlocal Tensor Sparse Representation and Low-Rank Regularization for Hyperspectral Image Compressive Sensing Reconstruction,” *Remote Sens.*, vol. 11, no. 2, p. 193, 2019.
- [50] R. Dian, S. Li, L. Fang, and Q. Wei, “Multispectral and Hyperspectral Image Fusion with Spatial-Spectral Sparse Representation,” *Inf. Fusion*, 2019.
- [51] E. Amaldi and V. Kann, “On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems,” *Theor. Comput. Sci.*, vol. 209, no. 1, pp. 237–260, 1998.
- [52] Y.-B. Zhao and M. Kočvara, “A new computational method for the sparsest solutions to systems of linear equations,” *SIAM J. Optim.*, vol. 25, no. 2, pp. 1110–1134, 2015.
- [53] Y.-B. Zhao and Z.-Q. Luo, “Constructing New Weighted ℓ_1 -Algorithms for the Sparsest Points of Polyhedral Sets,” *Math. Oper. Res.*, vol. 42, no. 1, pp. 57–76, 2016.
- [54] Q. Li, Y. Bai, C. Yu, and Y. Yuan, “A new piecewise quadratic approximation approach for L_0 norm minimization problem,” *Sci. China Math.*, vol. 62, no. 1, pp. 185–204, 2019.
- [55] S. S. Chen, D. L. Donoho, and M. A. Saunders, “Atomic decomposition by basis pursuit,” *SIAM Rev.*, vol. 43, no. 1, pp. 129–159, 2001.
- [56] E. J. Candes, M. B. Wakin, and S. P. Boyd, “Enhancing sparsity by reweighted ℓ_1 minimization,” *J. Fourier Anal. Appl.*, vol. 14, no. 5–6, pp. 877–905, 2008.
- [57] E. T. Hale, W. Yin, and Y. Zhang, “Fixed-Point Continuation for ℓ_1 -Minimization: Methodology and Convergence,” *SIAM J. Optim.*, vol. 19, no. 3, pp. 1107–1130, Jan. 2008, doi: 10.1137/070698920.
- [58] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, “Bregman Iterative Algorithms for ℓ_1 -Minimization with Applications to Compressed Sensing,” *SIAM J. Imaging Sci.*, vol. 1, no. 1, pp. 143–168, Jan. 2008, doi: 10.1137/070703983.
- [59] A. M. Bruckstein, D. L. Donoho, and M. Elad, “From sparse solutions of systems of equations to sparse modeling of signals and images,” *SIAM Rev.*, vol. 51, no. 1, pp. 34–81, 2009.
- [60] J. A. Tropp and S. J. Wright, “Computational Methods for Sparse Solution of Linear

- Inverse Problems,” *Proc. IEEE*, vol. 98, no. 6, pp. 948–958, Jun. 2010, doi: 10.1109/JPROC.2010.2044010.
- [61] Y. Zhao and D. Li, “Reweighted ℓ_1 -Minimization for Sparse Solutions to Underdetermined Linear Systems,” *SIAM J. Optim.*, vol. 22, no. 3, pp. 1065–1088, 2012, doi: 10.1137/110847445.
- [62] Z. Fan, M. Ni, Q. Zhu, C. Sun, and L. Kang, “L0-norm sparse representation based on modified genetic algorithm for face recognition,” *J. Vis. Commun. Image Represent.*, vol. 28, pp. 15–20, 2015.
- [63] M. Ç. Pinar, “Necessary and Sufficient Conditions for Noiseless Sparse Recovery via Convex Quadratic Splines,” *SIAM J. Matrix Anal. Appl.*, vol. 40, no. 1, pp. 194–209, 2019.
- [64] Y.-B. Zhao, *Sparse Optimization Theory and Methods*. CRC Press, 2018.
- [65] C. Weaver and N. Saito, “The Use of Mutual Coherence to Prove ℓ_1/ℓ_0 -Equivalence in Classification Problems,” *arXiv Prepr. arXiv1901.02783*, 2019.
- [66] R. Wang, M. Shen, T. Wang, and W. Cao, “L1-Norm Minimization for Multi-dimensional Signals Based on Geometric Algebra,” *Adv. Appl. Clifford Algebr.*, vol. 29, no. 2, p. 33, 2019.
- [67] O. L. Mangasarian, “Machine Learning via Polyhedral Concave Minimization,” in *Applied Mathematics and Parallel Computing: Festschrift for Klaus Ritter*, H. Fischer, B. Riedmüller, and S. Schäffler, Eds. Heidelberg: Physica-Verlag HD, 1996, pp. 175–188.
- [68] D. Donoho and J. Tanner, “Counting faces of randomly projected polytopes when the projection radically lowers dimension,” *J. Am. Math. Soc.*, vol. 22, no. 1, pp. 1–53, 2009.
- [69] S. Foucart and M.-J. Lai, “Sparsest solutions of underdetermined linear systems via ℓ_q -minimization for $0 < q \leq 1$,” *Appl. Comput. Harmon. Anal.*, vol. 26, no. 3, pp. 395–407, 2009.
- [70] T. T. Cai, L. Wang, and G. Xu, “New bounds for restricted isometry constants,” *IEEE Trans. Inf. Theory*, vol. 56, no. 9, pp. 4388–4394, 2010.
- [71] Q. Mo and S. Li, “New bounds on the restricted isometry constant δ_{2k} ,” *Appl. Comput. Harmon. Anal.*, vol. 31, no. 3, pp. 460–468, 2011.
- [72] T. T. Cai and A. Zhang, “Sharp RIP bound for sparse signal and low-rank matrix recovery,” *Appl. Comput. Harmon. Anal.*, vol. 35, no. 1, pp. 74–93, 2013.

-
- [73] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing*. Springer Science & Business Media, 2013.
 - [74] S. Zhou, L. Kong, and N. Xiu, “New bounds for RIC in compressed sensing,” *J. Oper. Res. Soc. China*, vol. 1, no. 2, pp. 227–237, 2013.
 - [75] J. Andersson and J. Strömberg, “On the Theorem of Uniform Recovery of Random Sampling Matrices,” *IEEE Trans. Inf. Theory*, vol. 60, no. 3, pp. 1700–1710, Mar. 2014, doi: 10.1109/TIT.2014.2300092.
 - [76] T. T. Cai and A. Zhang, “Sparse representation of a polytope and recovery of sparse signals and low-rank matrices,” *IEEE Trans. Inf. Theory*, vol. 60, no. 1, pp. 122–132, 2014.
 - [77] A. Cohen, W. Dahmen, and R. DeVore, “Compressed sensing and best k-term approximation,” *J. Am. Math. Soc.*, vol. 22, no. 1, pp. 211–231, 2009.
 - [78] Y. Zhang, “A simple proof for recoverability of ℓ_1 -minimization (ii): the nonnegativity case,” 2005.
 - [79] H. Cheng, *Sparse representation, modeling and learning in visual recognition*. Springer, 2016.
 - [80] Y. Zhang, “On Theory of Compressive Sensing via ℓ_1 -Minimization: Simple Derivations and Extensions,” 2008. Accessed: Mar. 30, 2019. [Online]. Available: <https://scholarship.rice.edu/handle/1911/102091>.
 - [81] Y. Zhang, “Theory of compressive sensing via ℓ_1 -minimization: a Non-RIP analysis and extensions,” *J. Oper. Res. Soc. China*, vol. 1, no. 1, pp. 79–105, 2013.
 - [82] J. A. Tropp, “Greed is good: Algorithmic results for sparse approximation,” *IEEE Trans. Inf. theory*, vol. 50, no. 10, pp. 2231–2242, 2004.
 - [83] J.-J. Fuchs, “On sparse representations in arbitrary redundant bases,” *IEEE Trans. Inf. theory*, vol. 50, no. 6, pp. 1341–1344, 2004.
 - [84] A. Juditsky and A. Nemirovski, “On verifiable sufficient conditions for sparse signal recovery via ℓ_1 minimization,” *Math. Program.*, vol. 127, no. 1, pp. 57–88, 2011.
 - [85] Y.-B. Zhao, “RSP-Based Analysis for Sparsest and Least ℓ_1 -Norm Solutions to Underdetermined Linear Systems,” *IEEE Trans. Signal Process.*, vol. 61, no. 22, pp. 5777–5788, 2013.
 - [86] D. L. Donoho and J. Tanner, “Sparse nonnegative solution of underdetermined linear

- equations by linear programming,” *Proc. Natl. Acad. Sci.*, vol. 102, no. 27, pp. 9446–9451, 2005.
- [87] M. A. Khajehnejad, A. G. Dimakis, W. Xu, and B. Hassibi, “Sparse Recovery of Nonnegative Signals With Minimal Expansion,” *IEEE Trans. Signal Process.*, vol. 59, no. 1, pp. 196–208, Jan. 2011, doi: 10.1109/TSP.2010.2082536.
- [88] A. Juditsky, F. K. Karzan, and A. Nemirovski, “Verifiable conditions of ℓ_1 -recovery for sparse signals with sign restrictions,” *Math. Program.*, vol. 127, no. 1, pp. 89–122, 2011.
- [89] T. Candes, EJ and Tao, “Decoding by linear programming,” *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, 2005.
- [90] D. L. Donoho and X. Huo, “Uncertainty principles and ideal atomic decomposition,” *IEEE Trans. Inf. Theory*, vol. 47, no. 7, pp. 2845–2862, Nov. 2001, doi: 10.1109/18.959265.
- [91] M. Elad and A. M. Bruckstein, “A generalized uncertainty principle and sparse representation in pairs of bases,” *IEEE Trans. Inf. Theory*, vol. 48, no. 9, pp. 2558–2567, 2002, doi: 10.1109/TIT.2002.801410.
- [92] D. L. Donoho and M. Elad, “Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization,” *Proc. Natl. Acad. Sci.*, vol. 100, no. 5, pp. 2197–2202, 2003.
- [93] A. Feuer and A. Nemirovski, “On sparse representation in pairs of bases,” *IEEE Trans. Inf. Theory*, vol. 49, no. 6, pp. 1579–1581, Jun. 2003, doi: 10.1109/TIT.2003.811926.
- [94] R. Gribonval and M. Nielsen, “Sparse decompositions in unions of bases,” *IEEE Trans. Inform. Theory*, vol. 49, no. 12, pp. 3320–3325, 2003.
- [95] E. J. Candes, “The restricted isometry property and its implications for compressed sensing,” *Comptes rendus Math.*, vol. 346, no. 9–10, pp. 589–592, 2008.
- [96] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. New York, NY : Springer Science+Business Media, LLC, 2010., 2010.
- [97] Y. Sun, S. Schaefer, and W. Wang, “Image Structure Retrieval via ℓ_0 Minimization,” *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 7, pp. 2129–2139, 2018.
- [98] C. Ye, G. Gui, and L. Xu, “Compressive Sensing Signal Reconstruction Using ℓ_0 -Norm Normalized Least Mean Fourth Algorithms,” *Circuits, Syst. Signal Process.*, vol. 37, no.

- 4, pp. 1724–1752, 2018.
- [99] N. Liu, Y. Du, and Y. Xu, “QR codes blind deconvolution algorithm based on binary characteristic and L0 norm minimization,” *Pattern Recognit. Lett.*, vol. 111, pp. 117–123, 2018.
- [100] F. Yang, Y. Shen, and Z. S. Liu, “The proximal alternating iterative hard thresholding method for l0 minimization, with complexity $O(1/k)$,” *J. Comput. Appl. Math.*, vol. 311, pp. 115–129, 2017.
- [101] G. B. Esmer, “Accurate diffraction field calculation method based on L 1-norm minimization from three-dimensional objects,” *Appl. Opt.*, vol. 58, no. 5, pp. A267--A272, 2019.
- [102] D. Ghilli and K. Kunisch, “On monotone and primal-dual active set schemes for ℓ^p ℓ^q -type problems, $p \in (0, 1]$ $q \in (0, 1]$,” *Comput. Optim. Appl.*, vol. 72, no. 1, pp. 45–85, 2019.
- [103] M. Thapa, A. Paudel, S. B. Mulani, and R. W. Walters, “Efficient Adaptive Sparse Polynomial Chaos Expansion with l 1-minimization and Sequential Sampling,” in *AIAA Scitech 2019 Forum*, 2019, p. 1226.
- [104] V. Bonifaci, “A Laplacian Approach to ℓ_1 -Norm Minimization,” *arXiv Prepr. arXiv1901.08836*, 2019.
- [105] Y. Wang, S. Wu, and B. Yu, “Unique Sharp Local Minimum in ℓ_1 -minimization Complete Dictionary Learning,” *arXiv Prepr. arXiv1902.08380*, 2019.
- [106] F. Santosa and W. W. Symes, “Linear inversion of band-limited reflection seismograms,” *SIAM J. Sci. Stat. Comput.*, vol. 7, no. 4, pp. 1307–1330, 1986.
- [107] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *J. R. Stat. Soc. Ser. B*, pp. 267–288, 1996.
- [108] R. Tibshirani, “Regression shrinkage and selection via the lasso: a retrospective,” *J. R. Stat. Soc. Ser. B (Statistical Methodol.)*, vol. 73, no. 3, pp. 273–282, 2011.
- [109] S. Ren, S. Huang, J. Ye, and X. Qian, “Safe feature screening for generalized lasso,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2992–3006, 2018.
- [110] S. Lee, N. Gornitz, E. P. Xing, D. Heckerman, and C. Lippert, “Ensembles of lasso screening rules,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2841–2852, 2018.

-
- [111] Y. Wang, Y. Shen, S. Mao, X. Chen, and H. Zou, "LASSO & LSTM integrated temporal model for short-term solar intensity forecasting," *IEEE Internet Things J.*, 2018.
 - [112] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical learning with Sparsity: the lasso and generalizations*. Chapman and Hall/CRC, 2015.
 - [113] H. C. So and W.-J. Zeng, "Outlier-Robust Matrix Completion via ℓ_p -Minimization," 2018.
 - [114] Y. Su and Z. Li, "Multi-band Weighted ℓ_p Norm Minimization for Color and Multispectral Image Denoising," *arXiv Prepr. arXiv1901.04206*, 2019.
 - [115] Y. Zhang, L. Comerford, I. A. Kougiumtzoglou, and M. Beer, "Lp-norm minimization for stochastic process power spectrum estimation subject to incomplete data," *Mech. Syst. Signal Process.*, vol. 101, pp. 361–376, 2018.
 - [116] L. Coroianu and L. Stefanini, "Properties of fuzzy transform obtained from Lp minimization and a connection with Zadeh's extension principle," *Inf. Sci. (Ny)*, vol. 478, pp. 331–354, 2019.
 - [117] A. Cui, J. Peng, H. Li, M. Wen, and J. Jia, "Iterative thresholding algorithm based on non-convex method for modified lp-norm regularization minimization," *J. Comput. Appl. Math.*, vol. 347, pp. 173–180, 2019.
 - [118] Z. Zhang, L. Wang, Q. Zhu, Z. Liu, and Y. Chen, "Noise modeling and representation based classification methods for face recognition," *Neurocomputing*, vol. 148, pp. 420–429, 2015.
 - [119] Y. Yang, H. T. Shen, Z. Ma, Z. Huang, and X. Zhou, " ℓ_2 , ℓ_1 -norm regularized discriminative feature selection for unsupervised learning," in *IJCAI proceedings-international joint conference on artificial intelligence*, 2011, vol. 22, no. 1, p. 1589.
 - [120] W. Deng, W. Yin, and Y. Zhang, "Group sparse optimization by alternating direction method," in *SPIE Optical Engineering+ Applications*, 2013, pp. 88580R--88580R.
 - [121] Z. Zhou and J. Yu, "Recovery analysis for weighted mixed ℓ_2/ℓ_p minimization with $0 < p \leq 1$," *J. Comput. Appl. Math.*, vol. 352, pp. 210–222, 2019.
 - [122] Y. Gao, J. Peng, and S. Yue, "Stability and robustness of the ℓ_2/ℓ_q -minimization for block sparse recovery," *Signal Processing*, vol. 137, pp. 287–297, 2017.
 - [123] Y. Xie, M. B. Wakin, and G. Tang, "Simultaneous Sparse Recovery and Blind Demodulation," *arXiv Prepr. arXiv1902.05023*, 2019.
 - [124] J. Wen, Z. Zhou, Z. Liu, M.-J. Lai, and X. Tang, "Sharp sufficient conditions for stable

- recovery of block sparse signals by block orthogonal matching pursuit,” *Appl. Comput. Harmon. Anal.*, 2018.
- [125] Q. Liu, F. Davoine, J. Yang, Y. Cui, Z. Jin, and F. Han, “A Fast and Accurate Matrix Completion Method Based on QR Decomposition and $L_{2,1}$ -Norm Minimization,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 30, no. 3, pp. 803–817, Mar. 2019, doi: 10.1109/TNNLS.2018.2851957.
- [126] H. Zhao, Q. Ye, M. A. Naiem, and L. Fu, “Robust $L_{2,1}$ -Norm Distance Enhanced Multi-Weight Vector Projection Support Vector Machine,” *IEEE Access*, vol. 7, pp. 3275–3286, 2019, doi: 10.1109/ACCESS.2018.2879052.
- [127] R. Li, X. Wang, L. Lei, and Y. Song, “ L_{21} -Norm Based Loss Function and Regularization Extreme Learning Machine,” *IEEE Access*, vol. 7, pp. 6575–6586, 2019, doi: 10.1109/ACCESS.2018.2887260.
- [128] X. Du, F. Nie, W. Wang, Y. Yang, and X. Zhou, “Exploiting Combination Effect for Unsupervised Feature Selection by $\ell_{2,0}$ Norm,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 30, no. 1, pp. 201–214, Jan. 2019, doi: 10.1109/TNNLS.2018.2837100.
- [129] F. Nie, H. Huang, X. Cai, and C. H. Ding, “Efficient and Robust Feature Selection via Joint $\ell_{2,1}$ -Norms Minimization,” in *Advances in Neural Information Processing Systems 23*, J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, Eds. Curran Associates, Inc., 2010, pp. 1813–1821.
- [130] S.-B. Chen, Y. Zhang, C. H. Q. Ding, Z.-L. Zhou, and B. Luo, “A discriminative multi-class feature selection method via weighted $\ell_{2,1}$ -norm and Extended Elastic Net,” *Neurocomputing*, vol. 275, pp. 1140–1149, Jan. 2018, doi: 10.1016/J.NEUCOM.2017.09.055.
- [131] R. Shang, W. Wang, R. Stolkin, and L. Jiao, “Non-Negative Spectral Learning and Sparse Regression-Based Dual-Graph Regularized Feature Selection,” *IEEE Trans. Cybern.*, vol. 48, no. 2, pp. 793–806, Feb. 2018, doi: 10.1109/TCYB.2017.2657007.
- [132] L. Zhang, Q. Zhang, B. Du, X. Huang, Y. Y. Tang, and D. Tao, “Simultaneous Spectral-Spatial Feature Selection and Extraction for Hyperspectral Images,” *IEEE Trans. Cybern.*, vol. 48, no. 1, pp. 16–28, Jan. 2018, doi: 10.1109/TCYB.2016.2605044.
- [133] X. Shi, Y. Yang, Z. Guo, and Z. Lai, “Face recognition by sparse discriminant analysis via joint $L_{2,1}$ -norm minimization,” *Pattern Recognit.*, vol. 47, no. 7, pp. 2447–2453, Jul.

- 2014, doi: 10.1016/J.PATCOG.2014.01.007.
- [134] Z. Lai, N. Liu, L. Shen, and H. Kong, “Robust Locally Discriminant Analysis via Capped Norm,” *IEEE Access*, vol. 7, pp. 4641–4652, 2019, doi: 10.1109/ACCESS.2018.2885131.
 - [135] S. Liao, Q. Gao, Z. Yang, F. Chen, F. Nie, and J. Han, “Discriminant Analysis via Joint Euler Transform and $\ell_{2,1}$ -Norm,” *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5668–5682, Nov. 2018, doi: 10.1109/TIP.2018.2859589.
 - [136] F. R. Boroujeni *et al.*, “Trace Ratio Optimization With Feature Correlation Mining for Multiclass Discriminant Analysis,” in *Thirty-Second AAAI Conference on Artificial Intelligence*, Apr. 2018, Accessed: Mar. 30, 2019. [Online]. Available: <https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/viewPaper/17259>.
 - [137] C. Ding, D. Zhou, X. He, and H. Zha, “R1-PCA: Rotational Invariant L1-norm Principal Component Analysis for Robust Subspace Factorization,” in *Proceedings of the 23rd International Conference on Machine Learning*, 2006, pp. 281–288, doi: 10.1145/1143844.1143880.
 - [138] S. Yi, Y. Liang, W. Liu, and F. Meng, “Adaptive Weighted Sparse Principal Component Analysis,” in *2018 IEEE International Conference on Multimedia and Expo (ICME)*, Jul. 2018, pp. 1–6, doi: 10.1109/ICME.2018.8486553.
 - [139] Y. Sun, N. Sheng, T. Li, Y. Li, E. Li, and P. Wei, “Fast nonlinear compressive sensing lithographic source and mask optimization method using Newton-IHTs algorithm,” *Opt. Express*, vol. 27, no. 3, pp. 2754–2770, 2019.
 - [140] F. Zhu, A. Liu, and V. Lau, “Signal Recovery for Massive Carrier Aggregation via Non-Linear Compressive Sensing,” in *2018 IEEE International Conference on Communications (ICC)*, 2018, pp. 1–7.
 - [141] X. Ma, Z. Wang, Y. Li, G. R. Arce, L. Dong, and J. Garcia-Frias, “Fast optical proximity correction method based on nonlinear compressive sensing,” *Opt. Express*, vol. 26, no. 11, pp. 14479–14498, 2018.
 - [142] K. Zhang, Z. Yang, and Z. Wang, “Nonlinear Structured Signal Estimation in High Dimensions via Iterative Hard Thresholding,” in *International Conference on Artificial Intelligence and Statistics*, 2018, pp. 258–268.
 - [143] T. Blumensath, “Compressed sensing with nonlinear observations and related nonlinear optimization problems,” *IEEE Trans. Inf. Theory*, vol. 59, no. 6, pp. 3466–3474, 2013.

-
- [144] Y. T. Lee, “Faster algorithms for convex and combinatorial optimization,” 2016, Accessed: Mar. 30, 2019. [Online]. Available: <https://dspace.mit.edu/handle/1721.1/104467>.
 - [145] M. B. Cohen, Y. T. Lee, and Z. Song, “Solving Linear Programs in the Current Matrix Multiplication Time,” *CoRR*, vol. abs/1810.0, 2018, [Online]. Available: <http://arxiv.org/abs/1810.07896>.
 - [146] H. Zhang, W. Yin, and L. Cheng, “Necessary and sufficient conditions of solution uniqueness in 1-norm minimization,” *J. Optim. Theory Appl.*, vol. 164, no. 1, pp. 109–122, 2015.
 - [147] J. Wright and Y. Ma, “Dense Error Correction Via ℓ^1 -Minimization,” *IEEE Trans. Inf. Theory*, vol. 56, no. 7, pp. 3540–3560, Jul. 2010, doi: 10.1109/TIT.2010.2048473.
 - [148] G. Zhang, S. Cai, and N. Xiong, “The application of social characteristic and L1 optimization in the error correction for network coding in wireless sensor networks,” *Sensors*, vol. 18, no. 2, p. 450, 2018.
 - [149] G. Zhang *et al.*, “The combination of sparse learning and list decoding of subspace codes for error correction in random network coding,” *EURASIP J. Wirel. Commun. Netw.*, vol. 2018, no. 1, p. 70, 2018.
 - [150] J. Yang and Y. Zhang, “Alternating direction algorithms for ℓ_1 -problems in compressive sensing,” *SIAM J. Sci. Comput.*, vol. 33, no. 1, pp. 250–278, 2011.
 - [151] D. Guo, Z. Tu, J. Wang, M. Xiao, X. Du, and X. Qu, “Salt and Pepper Noise Removal with Multi-Class Dictionary Learning and L0 Norm Regularizations,” *Algorithms*, vol. 12, no. 1, p. 7, 2019.
 - [152] J. Jiang, Z. Wang, C. Chen, and T. Lu, “L1-l1 norms for face super-resolution with mixed Gaussian-impulse noise,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 2089–2093.
 - [153] C. Zhao, X. Wang, and W.-K. Cham, “Background subtraction via robust dictionary learning,” *EURASIP J. Image Video Process.*, vol. 2011, no. 1, pp. 1–12, 2011.
 - [154] Z. Fan, D. Zhang, X. Wang, Q. Zhu, and Y. Wang, “Virtual dictionary based kernel sparse representation for face recognition,” *Pattern Recognit.*, vol. 76, pp. 1–13, 2018.
 - [155] S. Gao, I. W.-H. Tsang, and L.-T. Chia, “Sparse representation with kernels,” *IEEE Trans. Image Process.*, vol. 22, no. 2, pp. 423–434, 2013.

-
- [156] C.-Y. Lu, H. Min, J. Gui, L. Zhu, and Y.-K. Lei, “Face recognition via weighted sparse representation,” *J. Vis. Commun. Image Represent.*, vol. 24, no. 2, pp. 111–116, 2013.
 - [157] Y. Xu, Z. Zhang, G. Lu, and J. Yang, “Approximately symmetrical face images for image preprocessing in face recognition and sparse representation based classification,” *Pattern Recognit.*, vol. 54, pp. 68–82, 2016.
 - [158] L. Zhuang, T.-H. Chan, A. Y. Yang, S. S. Sastry, and Y. Ma, “Sparse illumination learning and transfer for single-sample face recognition with image corruption and misalignment,” *Int. J. Comput. Vis.*, vol. 114, no. 2–3, pp. 272–287, 2015.
 - [159] J. Liu, P. C. Cosman, and B. D. Rao, “Robust Linear Regression via ℓ_0 Regularization,” *IEEE Trans. Signal Process.*, vol. 66, no. 3, pp. 698–713, 2018.
 - [160] A. M. Tillmann and M. E. Pfetsch, “The computational complexity of the restricted isometry property, the nullspace property, and related concepts in compressed sensing,” *IEEE Trans. Inf. Theory*, vol. 60, no. 2, pp. 1248–1259, 2014.
 - [161] Y.-B. Zhao, “Equivalence and Strong Equivalence Between the Sparsest and Least ℓ_1 -Norm Nonnegative Solutions of Linear Systems and Their Applications,” *J. Oper. Res. Soc. China*, vol. 2, no. 2, pp. 171–193, 2014.
 - [162] M. D. Plumbley, “On Polar Polytopes and the Recovery of Sparse Representations,” *IEEE Trans. Inf. Theory*, vol. 53, no. 9, pp. 3188–3195, 2007, doi: 10.1109/TIT.2007.903129.
 - [163] S. Foucart and M.-J. Lai, “Sparsest solutions of underdetermined linear systems via ℓ_q -minimization for $0 < q < 1$,” *Appl. Comput. Harmon. Anal.*, vol. 26, no. 3, pp. 395–407, May 2009, doi: 10.1016/J.ACHA.2008.09.001.
 - [164] M.-J. Lai and J. Wang, “An Unconstrained ℓ_q Minimization with $0 < q \leq 1$ for Sparse Solution of Underdetermined Linear Systems,” *SIAM J. Optim.*, vol. 21, no. 1, pp. 82–101, Jan. 2011, doi: 10.1137/090775397.
 - [165] D. Wipf and S. Nagarajan, “Iterative Reweighted ℓ_1 and ℓ_2 Methods for Finding Sparse Solutions,” *IEEE J. Sel. Top. Signal Process.*, vol. 4, no. 2, pp. 317–329, Apr. 2010, doi: 10.1109/JSTSP.2010.2042413.
 - [166] E. J. Candès, M. B. Wakin, and S. P. Boyd, “Enhancing Sparsity by Reweighted ℓ_1 Minimization,” *J. Fourier Anal. Appl.*, vol. 14, no. 5, pp. 877–905, Dec. 2008, doi: 10.1007/s00041-008-9045-x.
 - [167] I. F. Gorodnitsky and B. D. Rao, “Sparse signal reconstruction from limited data using

-
- FOCUSS: a re-weighted minimum norm algorithm,” *IEEE Trans. Signal Process.*, vol. 45, no. 3, pp. 600–616, Mar. 1997, doi: 10.1109/78.558475.
- [168] Y. Sun, P. Babu, and D. P. Palomar, “Majorization-Minimization Algorithms in Signal Processing, Communications, and Machine Learning,” *IEEE Trans. Signal Process.*, vol. 65, no. 3, pp. 794–816, Feb. 2017, doi: 10.1109/TSP.2016.2601299.
- [169] E. J. Candes, J. Romberg, and T. Tao, “Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information,” *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006, doi: 10.1109/TIT.2005.862083.
- [170] E. J. Candes, J. K. Romberg, and T. Tao, “Stable signal recovery from incomplete and inaccurate measurements,” *Commun. Pure Appl. Math. A J. Issued by Courant Inst. Math. Sci.*, vol. 59, no. 8, pp. 1207–1223, 2006.
- [171] D. Needell, “Noisy signal recovery via iterative reweighted L1-minimization,” in *2009 Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers*, Nov. 2009, pp. 113–117, doi: 10.1109/ACSSC.2009.5470154.
- [172] M. S. Asif and J. Romberg, “Sparse Recovery of Streaming Signals Using ℓ_1 -Homotopy,” *IEEE Trans. Signal Process.*, vol. 62, no. 16, pp. 4209–4223, Aug. 2014, doi: 10.1109/TSP.2014.2328981.
- [173] Z. Xie and J. Hu, “Rewighted L1-minimization for sparse solutions to underdetermined linear systems,” in *2013 6th International Congress on Image and Signal Processing (CISP)*, Dec. 2013, vol. 03, pp. 1660–1664, doi: 10.1109/CISP.2013.6743943.
- [174] M. A. Khajehnejad, W. Xu, A. S. Avestimehr, and B. Hassibi, “Weighted ℓ_1 minimization for sparse recovery with prior information,” in *2009 IEEE International Symposium on Information Theory*, Jun. 2009, pp. 483–487, doi: 10.1109/ISIT.2009.5205716.
- [175] D. Needell, R. Saab, and T. Woolf, “Weighted ℓ_1 -minimization for sparse recovery under arbitrary prior information,” *Inf. Inference A J. IMA*, vol. 6, no. 3, pp. 284–309, 2017, doi: 10.1093/imaiai/iaw023.
- [176] S. Zhou, N. Xiu, Y. Wang, and L. Kong, “Exact Recovery for Sparse Signal via Weighted ℓ_1 Minimization,” *CoRR*, vol. abs/1312.2, 2013, [Online]. Available: <http://arxiv.org/abs/1312.2358>.
- [177] R. Chartrand, “Iteratively reweighted algorithms for compressive sensing,” in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, Mar. 2008, pp.

- 3869–3872, doi: 10.1109/ICASSP.2008.4518498.
- [178] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk, “Iteratively reweighted least squares minimization for sparse recovery,” *Commun. Pure Appl. Math.*, vol. 63, no. 1, pp. 1–38, 2010, doi: 10.1002/cpa.20303.
- [179] R. Ahmad and P. Schniter, “Iteratively Reweighted ℓ_1 Approaches to Sparse Composite Regularization,” *IEEE Trans. Comput. Imaging*, vol. 1, no. 4, pp. 220–235, Dec. 2015, doi: 10.1109/TCI.2015.2485078.
- [180] U. Ayaz and H. Rauhut, “Nonuniform sparse recovery with subgaussian matrices,” *Electron. Trans. Numer. Anal.*, vol. 41, pp. 167–178, 2014.
- [181] Y. Zhao and C. Xu, “1-Bit compressive sensing: Reformulation and RRSP-based sign recovery theory,” *Sci. China Math.*, vol. 59, no. 10, pp. 2049–2074, 2016.
- [182] K. Knudson, R. Saab, and R. Ward, “One-Bit Compressive Sensing With Norm Estimation,” *IEEE Trans. Inf. Theory*, vol. 62, no. 5, pp. 2748–2758, May 2016, doi: 10.1109/TIT.2016.2527637.
- [183] X. Huang *et al.*, “Mixed one-bit compressive sensing with applications to overexposure correction for $\{CT\}$ reconstruction,” *CoRR*, vol. abs/1701.0, 2017, [Online]. Available: <http://arxiv.org/abs/1701.00694>.
- [184] L. Shen and B. W. Suter, “Blind One-Bit Compressive Sampling,” *CoRR*, vol. abs/1302.1, 2013, [Online]. Available: <http://arxiv.org/abs/1302.1419>.
- [185] C. Xu, “Sparsity optimization and RRSP-based theory for 1-bit compressive sensing,” University of Birmingham, 2016.
- [186] L. Jacques, J. N. Laska, P. T. Boufounos, and R. G. Baraniuk, “Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors,” *IEEE Trans. Inf. Theory*, vol. 59, no. 4, pp. 2082–2102, 2013.
- [187] J. Acharya, A. Bhattacharyya, and P. Kamath, “Improved bounds for universal one-bit compressive sensing,” in *2017 IEEE International Symposium on Information Theory (ISIT)*, 2017, pp. 2353–2357.
- [188] M. Yan, Y. Yang, and S. Osher, “Robust 1-bit Compressive Sensing Using Adaptive Outlier Pursuit,” *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3868–3875, Jul. 2012, doi: 10.1109/TSP.2012.2193397.
- [189] A. Movahed, A. Panahi, and G. Durisi, “A robust RFPI-based 1-bit compressive sensing

- reconstruction algorithm,” in *2012 IEEE Information Theory Workshop*, 2012, pp. 567–571, doi: 10.1109/ITW.2012.6404739.
- [190] F. Li, J. Fang, H. Li, and L. Huang, “Robust One-Bit Bayesian Compressed Sensing with Sign-Flip Errors,” *IEEE Signal Process. Lett.*, vol. 22, no. 7, pp. 857–861, Jul. 2015, doi: 10.1109/LSP.2014.2373380.
- [191] D.-Q. Dai, L. Shen, Y. Xu, and N. Zhang, “Noisy 1-bit compressive sensing: models and algorithms,” *Appl. Comput. Harmon. Anal.*, vol. 40, no. 1, pp. 1–32, Jan. 2016, doi: 10.1016/J.ACHA.2014.12.001.
- [192] R. G. Baraniuk, S. Foucart, D. Needell, Y. Plan, and M. Wootters, “Exponential Decay of Reconstruction Error From Binary Measurements of Sparse Signals,” *IEEE Trans. Inf. Theory*, vol. 63, no. 6, pp. 3368–3385, Jun. 2017, doi: 10.1109/TIT.2017.2688381.
- [193] T. T. Cai, G. Xu, and J. Zhang, “On Recovery of Sparse Signals Via ℓ_1 Minimization,” *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3388–3397, Jul. 2009, doi: 10.1109/TIT.2009.2021377.
- [194] D. L. Donoho, M. Elad, and V. N. Temlyakov, “Stable recovery of sparse overcomplete representations in the presence of noise,” *IEEE Trans. Inf. Theory*, vol. 52, no. 1, pp. 6–18, Jan. 2006, doi: 10.1109/TIT.2005.860430.
- [195] Y. C. Eldar and G. Kutyniok, *Compressed sensing: theory and applications*. Cambridge University Press, 2012.
- [196] D. Donoho and P. Stark, “Uncertainty Principles and Signal Recovery,” *SIAM J. Appl. Math.*, vol. 49, no. 3, pp. 906–931, 1989, doi: 10.1137/0149053.
- [197] Y. Sharon, J. Wright, and Y. Ma, “Computation and relaxation of conditions for equivalence between ℓ_1 and ℓ_0 minimization,” *Submitt. to IEEE Trans. Inf. Theory*, vol. 5, 2007.
- [198] Y.-B. Zhao, “New and improved conditions for uniqueness of sparsest solutions of underdetermined linear systems,” *Appl. Math. Comput.*, vol. 224, pp. 58–73, 2013.
- [199] E. Candes, T. Tao, and others, “The Dantzig selector: Statistical estimation when p is much larger than n ,” *Ann. Stat.*, vol. 35, no. 6, pp. 2313–2351, 2007.
- [200] J. N. Laska, Z. Wen, W. Yin, and R. G. Baraniuk, “Trust, But Verify: Fast and Accurate Signal Recovery From 1-Bit Compressive Measurements,” *IEEE Trans. Signal Process.*, vol. 59, no. 11, pp. 5289–5301, Nov. 2011, doi: 10.1109/TSP.2011.2162324.

-
- [201] Y. Kong, Z. Zheng, and J. Lv, “The Constrained Dantzig Selector with Enhanced Consistency,” *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 4205–4226, Jan. 2016, [Online]. Available: <http://dl.acm.org/citation.cfm?id=2946645.3007076>.
 - [202] R. Mazumder and P. Radchenko, “The Discrete Dantzig Selector: Estimating Sparse Linear Models via Mixed Integer Linear Optimization,” *IEEE Trans. Inf. Theory*, vol. 63, no. 5, pp. 3053–3075, May 2017, doi: 10.1109/TIT.2017.2658023.
 - [203] P. Bühlmann and S. van de Geer, *Statistics for high-dimensional data: methods, theory and applications*. Berlin, Heidelberg: Springer Science & Business Media, 2011.
 - [204] Y.-B. Zhao and D. Li, “A Theoretical Analysis of Sparse Recovery Stability of Dantzig Selector and LASSO,” *arXiv Prepr. arXiv1711.03783*, 2017.
 - [205] E. Candès and T. Tao, “Rejoinder: The Dantzig selector: Statistical estimation when p is much larger than n ,” *Ann. Stat.*, vol. 35, 2008, doi: 10.1214/009053607000000532.
 - [206] A. Iouditski and A. Nemirovski, “On Verifiable Sufficient Conditions for Sparse Signal Recovery via ℓ_1 Minimization,” *arXiv Prepr. arXiv0809.2650*, 2008.
 - [207] J. A. Tropp, “Just relax: Convex programming methods for identifying sparse signals in noise,” *IEEE Trans. Inf. theory*, vol. 52, no. 3, pp. 1030–1051, 2006.
 - [208] M. Grasmair, O. Scherzer, and M. Haltmeier, “Necessary and sufficient conditions for linear convergence of ℓ_1 -regularization,” *Commun. Pure Appl. Math.*, vol. 64, no. 2, pp. 161–182, 2011, doi: 10.1002/cpa.20350.
 - [209] J. C. Gilbert, “On the solution uniqueness characterization in the L_1 norm and polyhedral gauge recovery,” *J. Optim. Theory Appl.*, vol. 172, no. 1, pp. 70–101, 2017.
 - [210] M. E. Davies and R. Gribonval, “Restricted Isometry Constants Where ℓ_p Sparse Recovery Can Fail for $0 < p \leq 1$,” *IEEE Trans. Inf. Theory*, vol. 55, no. 5, pp. 2203–2214, May 2009, doi: 10.1109/TIT.2009.2016030.
 - [211] S. Foucart, “A note on guaranteed sparse recovery via ℓ_1 -minimization,” *Appl. Comput. Harmon. Anal.*, vol. 29, no. 1, pp. 97–103, Jul. 2010, doi: 10.1016/J.ACHA.2009.10.004.
 - [212] S. Foucart, “Stability and robustness of ℓ_1 -minimizations with Weibull matrices and redundant dictionaries,” *Linear Algebra Appl.*, vol. 441, pp. 4–21, Jan. 2014, doi: 10.1016/J.LAA.2012.10.003.
 - [213] Y.-B. Zhao, H. Jiang, and Z.-Q. Luo, “Weak Stability of ℓ_1 -Minimization Methods in

- Sparse Data Reconstruction,” *Math. Oper. Res.*, 2018.
- [214] H. Zhang, M. Yan, and W. Yin, “One condition for solution uniqueness and robustness of both ℓ_1 -synthesis and ℓ_1 -analysis minimizations,” *Adv. Comput. Math.*, vol. 42, no. 6, pp. 1381–1399, 2016.
- [215] D. Schäfer and E. Hüllermeier, “Dyad ranking using a bilinear Plackett-Luce model,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2015, pp. 227–242.
- [216] U. Ayaz, S. Dirksen, and H. Rauhut, “Uniform recovery of fusion frame structured sparse signals,” *Appl. Comput. Harmon. Anal.*, vol. 41, no. 2, pp. 341–361, 2016.
- [217] R. Saab, R. Wang, and Ö. Yılmaz, “Quantization of compressive samples with stable and robust recovery,” *Appl. Comput. Harmon. Anal.*, vol. 44, no. 1, pp. 123–143, 2018.
- [218] Y. Traonmilin and R. Gribonval, “Stable recovery of low-dimensional cones in Hilbert spaces: One RIP to rule them all,” *Appl. Comput. Harmon. Anal.*, vol. 45, no. 1, pp. 170–205, 2018.
- [219] Q. Sun, “Recovery of sparsest signals via ℓ_q -minimization,” *Appl. Comput. Harmon. Anal.*, vol. 32, no. 3, pp. 329–341, May 2012, doi: 10.1016/J.ACHA.2011.07.001.
- [220] J. Cahill, X. Chen, and R. Wang, “The gap between the null space property and the restricted isometry property,” *Linear Algebra Appl.*, vol. 501, pp. 363–375, 2016.
- [221] N. Meinshausen and P. Bühlmann, “Stability selection,” *J. R. Stat. Soc. Ser. B (Statistical Methodol.)*, vol. 72, no. 4, pp. 417–473, 2010, doi: 10.1111/j.1467-9868.2010.00740.x.
- [222] P. Zhao and B. Yu, “On Model Selection Consistency of Lasso,” *J. Mach. Learn. Res.*, vol. 7, no. Nov, pp. 2541–2563, 2006, Accessed: Mar. 31, 2019. [Online]. Available: <http://www.jmlr.org/papers/v7/zhao06a.html>.
- [223] P. J. Bickel, Y. Ritov, and A. B. Tsybakov, “Simultaneous analysis of Lasso and Dantzig selector,” *Ann. Stat.*, vol. 37, no. 4, pp. 1705–1732, Aug. 2009, doi: 10.1214/08-AOS620.
- [224] M. Rudelson and S. Zhou, “Reconstruction From Anisotropic Random Measurements,” *IEEE Trans. Inf. Theor.*, vol. 59, no. 6, pp. 3434–3447, Jun. 2013, doi: 10.1109/TIT.2013.2243201.
- [225] A. Juditsky, F. K. Karzan, A. Nemirovski, and B. Polyak, “On the Accuracy of ℓ_1 -filtering of Signals with Block-sparse Structure,” in *Proceedings of the 24th International Conference on Neural Information Processing Systems*, 2011, pp. 1260–1268, [Online].

- Available: <http://dl.acm.org/citation.cfm?id=2986459.2986600>.
- [226] A. Juditsky and A. Nemirovski, “Accuracy Guarantees for ℓ_1 -Recovery,” *IEEE Trans. Inf. Theory*, vol. 57, no. 12, pp. 7818–7839, Dec. 2011, doi: 10.1109/TIT.2011.2162569.
 - [227] S. M. Robinson, “Bounds for error in the solution set of a perturbed linear program,” *Linear Algebra Appl.*, vol. 6, pp. 69–81, Jan. 1973, doi: 10.1016/0024-3795(73)90007-4.
 - [228] D. L. Donoho and X. Huo, “Uncertainty Principles and Ideal Atomic Decomposition,” 1999.
 - [229] M. Elad and A. M. Bruckstein, “On sparse signal representations,” in *Proceedings 2001 International Conference on Image Processing (Cat. No.01CH37205)*, Oct. 2001, vol. 1, pp. 3–6 vol.1, doi: 10.1109/ICIP.2001.958936.
 - [230] C. Herzet, A. Drémeau, and C. Soussen, “Relaxed recovery conditions for OMP/OLS by exploiting both coherence and decay,” *IEEE Trans. Inf. Theory*, vol. 62, no. 1, pp. 459–470, 2016.
 - [231] M. Yang and F. de Hoog, “Orthogonal matching pursuit with thresholding and its application in compressive sensing,” *IEEE Trans. Signal Process.*, vol. 63, no. 20, pp. 5479–5486, 2015.
 - [232] E. J. Candes and T. Tao, “Near-Optimal Signal Recovery From Random Projections: Universal Encoding Strategies?,” *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.
 - [233] X.-L. Hu, J. Wen, W. K. Wong, L. Tong, and J. Cui, “On uniqueness of sparse signal recovery,” *Signal Processing*, vol. 150, pp. 66–74, 2018.
 - [234] E. Candes and T. Tao, “Near optimal signal recovery from random projections: Universal encoding strategies?,” *arXiv Prepr. math/0410542*, 2004.
 - [235] J. Weed, “Approximately certifying the restricted isometry property is hard,” *IEEE Trans. Inf. Theory*, vol. 64, no. 8, pp. 5488–5497, 2018.
 - [236] A. S. Bandeira, E. Dobriban, D. G. Mixon, and W. F. Sawin, “Certifying the restricted isometry property is hard,” *IEEE Trans. Inf. theory*, vol. 59, no. 6, pp. 3448–3450, 2013.
 - [237] E. J. Candès and others, “Compressive sampling,” in *Proceedings of the international congress of mathematicians*, 2006, vol. 3, pp. 1433–1452.
 - [238] Z. Sun, W. Cui, and Y. Liu, “Recovery of Structured Signals From Corrupted Non-Linear

- Measurements,” *arXiv Prepr. arXiv1901.08349*, 2019.
- [239] Y. Tsaig and D. L. Donoho, “Extensions of compressed sensing,” *Signal Processing*, vol. 86, no. 3, pp. 549–571, 2006.
 - [240] S. A. Van De Geer, P. Bühlmann, and others, “On the conditions used to prove oracle results for the Lasso,” *Electron. J. Stat.*, vol. 3, pp. 1360–1392, 2009.
 - [241] S. Gopi, P. Netrapalli, P. Jain, and A. Nori, “One-bit compressed sensing: Provable support and vector recovery,” in *International Conference on Machine Learning*, 2013, pp. 154–162.
 - [242] R. Gribonval and M. Nielsen, “Sparse representations in unions of bases,” INRIA, 2002.
 - [243] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, “Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition,” in *Proceedings of 27th Asilomar conference on signals, systems and computers*, 1993, pp. 40–44.
 - [244] Y. Tsaig and D. L. Donoho, “Extensions of Compressed Sensing,” 2004.
 - [245] A. Juditsky, F. K. Karzan, and A. S. Nemirovski, “Verifiable Conditions of L1-recovery of Sparse Signals with Sign Restriction,” in *INFORMS Annual Meeting*, 2009.
 - [246] C. Blair, “Problem Complexity and Method Efficiency in Optimization (A. S. Nemirovsky and D. B. Yudin),” *SIAM Rev.*, vol. 27, no. 2, pp. 264–265, Jun. 1985, doi: 10.1137/1027074.
 - [247] J. Mairal, R. Jenatton, F. R. Bach, and G. R. Obozinski, “Network Flow Algorithms for Structured Sparsity,” in *Advances in Neural Information Processing Systems 23*, 2010, pp. 1558–1566, [Online]. Available: <http://papers.nips.cc/paper/3965-network-flow-algorithms-for-structured-sparsity.pdf>.
 - [248] L. Jacob, G. Obozinski, and J.-P. Vert, “Group lasso with overlap and graph lasso,” in *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*, 2009, pp. 1–8, doi: 10.1145/1553374.1553431.
 - [249] X. Chen, Q. Lin, S. Kim, J. G. Carbonell, and E. P. Xing, “Smoothing proximal gradient method for general structured sparse regression,” *Ann. Appl. Stat.*, vol. 6, no. 2, pp. 719–752, Jun. 2012, doi: 10.1214/11-AOAS514.
 - [250] S. Kim and E. P. Xing, “Statistical Estimation of Correlated Genome Associations to a Quantitative Trait Network,” *PLoS Genet.*, vol. 5, no. 8, p. e1000587, Aug. 2009, doi:

- 10.1371/journal.pgen.1000587.
- [251] H. Ouyang and A. Gray, “Stochastic Alternating Direction Method of Multipliers,” in *Proceedings of the 30th International Conference on Machine Learning*, 2012, vol. 28, no. 1, pp. 80–88, [Online]. Available: <http://proceedings.mlr.press/v28/ouyang13.html>.
 - [252] X. Chen, Q. Lin, S. Kim, J. G. Carbonell, and E. P. Xing, “An efficient proximal gradient method for general structured sparse learning,” *Stat*, vol. 1050, p. 26, 2011.
 - [253] W. Zhong and J. Kwok, “Accelerated stochastic gradient method for composite regularization,” in *Artificial Intelligence and Statistics*, 2014, pp. 1086–1094.
 - [254] C. H. Teo, A. Smola, S. V. N. Vishwanathan, and Q. V. Le, “A scalable modular convex solver for regularized risk minimization,” in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '07*, 2007, p. 727, doi: 10.1145/1281192.1281270.
 - [255] C. H. Teo, S. V. N. Vishwanathan, A. J. Smola, and Q. V. Le, “Bundle Methods for Regularized Risk Minimization,” *J. Mach. Learn. Res.*, vol. 11, no. Jan, pp. 311–365, 2010, Accessed: Dec. 16, 2017. [Online]. Available: <http://www.jmlr.org/papers/v11/teo10a.html>.
 - [256] E. Richard, P.-A. Savalle, and N. Vayatis, “Estimation of Simultaneously Sparse and Low Rank Matrices,” Jun. 2012, Accessed: Dec. 17, 2017. [Online]. Available: <http://arxiv.org/abs/1206.6474>.
 - [257] K. Zhou, H. Zha, and L. Song, “Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes,” in *Artificial Intelligence and Statistics*, 2013, pp. 641–649.
 - [258] F.-X. Dupe, J. M. Fadili, and J.-L. Starck, “A Proximal Iteration for Deconvolving Poisson Noisy Images Using Sparse Representations,” *IEEE Trans. Image Process.*, vol. 18, no. 2, pp. 310–321, Feb. 2009, doi: 10.1109/TIP.2008.2008223.
 - [259] N. Pustelnik, C. Chaux, and J. Pesquet, “Parallel Proximal Algorithm for Image Restoration Using Hybrid Regularization,” *IEEE Trans. Image Process.*, vol. 20, no. 9, pp. 2450–2462, Sep. 2011, doi: 10.1109/TIP.2011.2128335.
 - [260] A. Y. Yang, S. S. Sastry, A. Ganesh, and Y. Ma, “Fast ℓ_1 -minimization algorithms and an application in robust face recognition: A review,” in *2010 IEEE International Conference on Image Processing*, Sep. 2010, pp. 1849–1852, doi: 10.1109/ICIP.2010.5651522.

-
- [261] J. Huang, T. Zhang, and D. Metaxas, “Learning with Structured Sparsity,” *J. Mach. Learn. Res.*, vol. 12, no. Nov, pp. 3371–3412, 2011, Accessed: Dec. 17, 2017. [Online]. Available: <http://www.jmlr.org/papers/v12/huang11b.html>.
 - [262] F. R. Bach, “Structured sparsity-inducing norms through submodular functions,” in *Advances in Neural Information Processing Systems*, 2010, pp. 118–126.
 - [263] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski, “Structured Sparsity through Convex Optimization,” *Stat. Sci.*, vol. 27, no. 4, pp. 450–468, Nov. 2012, doi: 10.1214/12-STS394.
 - [264] D. Chen and R. J. Plemmons, “Nonnegativity constraints in numerical analysis,” in *The Birth of Numerical Analysis*, WORLD SCIENTIFIC, 2009, pp. 109–139.
 - [265] Y. Xu and W. Yin, “A Block Coordinate Descent Method for Regularized Multiconvex Optimization with Applications to Nonnegative Tensor Factorization and Completion,” *SIAM J. Imaging Sci.*, vol. 6, no. 3, pp. 1758–1789, Jan. 2013, doi: 10.1137/120887795.
 - [266] D. Davis and W. Yin, “A Three-Operator Splitting Scheme and its Optimization Applications,” *Set-Valued Var. Anal.*, pp. 1–30, Jun. 2017, doi: 10.1007/s11228-017-0421-z.
 - [267] Y. Nesterov, “A method of solving a convex programming problem with convergence rate $O(1/k^2)$,” in *Soviet Mathematics Doklady*, 1983, vol. 27, no. 2, pp. 372–376.
 - [268] Y. Nesterov, “Gradient methods for minimizing composite functions,” *Math. Program.*, vol. 140, no. 1, pp. 125–161, Aug. 2013, doi: 10.1007/s10107-012-0629-5.
 - [269] Y. Nesterov, *Introductory lectures on convex optimization : a basic course*. Springer, 2014.
 - [270] A. Beck and M. Teboulle, “A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems,” *SIAM J. Imaging Sci.*, vol. 2, no. 1, pp. 183–202, Jan. 2009, doi: 10.1137/080716542.
 - [271] A. Beck and M. Teboulle, “Fast Gradient-Based Algorithms for Constrained Total Variation Image Denoising and Deblurring Problems,” *IEEE Trans. Image Process.*, vol. 18, no. 11, pp. 2419–2434, Nov. 2009, doi: 10.1109/TIP.2009.2028250.
 - [272] A. Nemirovskii, D. B. Yudin, and E. R. Dawson, “Problem complexity and method efficiency in optimization,” 1983.
 - [273] B. T. Polyak, “Subgradient methods: A survey of Soviet research,” in *Nonsmooth*

- optimization: Proceedings of the IIASA workshop March*, 1978, pp. 5–30.
- [274] Y. Nesterov, “Excessive Gap Technique in Nonsmooth Convex Minimization,” *SIAM J. Optim.*, vol. 16, no. 1, pp. 235–249, Jan. 2005, doi: 10.1137/S1052623403422285.
- [275] R. I. Boş and C. Hendrich, “A variable smoothing algorithm for solving convex optimization problems,” *TOP*, vol. 23, no. 1, pp. 124–150, Apr. 2015, doi: 10.1007/s11750-014-0326-z.
- [276] Q. Tran-Dinh, “Adaptive smoothing algorithms for nonsmooth composite convex minimization,” *Comput. Optim. Appl.*, vol. 66, no. 3, pp. 425–451, Apr. 2017, doi: 10.1007/s10589-016-9873-6.
- [277] Y.-L. Yu, “Better Approximation and Faster Algorithm Using the Proximal Average,” in *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 458–466.
- [278] L. M. Briceño-Arias, P. L. Combettes, J.-C. Pesquet, and N. Pustelnik, “Proximal Algorithms for Multicomponent Image Recovery Problems,” *J. Math. Imaging Vis.*, vol. 41, no. 1–2, pp. 3–22, Sep. 2011, doi: 10.1007/s10851-010-0243-1.
- [279] P. L. Combettes and J.-C. Pesquet, “A Douglas–Rachford Splitting Approach to Nonsmooth Convex Variational Signal Recovery,” *IEEE J. Sel. Top. Signal Process.*, vol. 1, no. 4, pp. 564–574, Dec. 2007, doi: 10.1109/JSTSP.2007.910264.
- [280] P. L. Combettes and J.-C. Pesquet, “A proximal decomposition method for solving convex variational inverse problems,” *Inverse Probl.*, vol. 24, no. 6, p. 065014, Dec. 2008, doi: 10.1088/0266-5611/24/6/065014.
- [281] L. Condat, “A Primal–Dual Splitting Method for Convex Optimization Involving Lipschitzian, Proximable and Linear Composite Terms,” *J. Optim. Theory Appl.*, vol. 158, no. 2, pp. 460–479, Aug. 2013, doi: 10.1007/s10957-012-0245-9.
- [282] R. D. C. Monteiro and B. F. Svaiter, “Iteration-Complexity of Block-Decomposition Algorithms and the Alternating Direction Method of Multipliers,” *SIAM J. Optim.*, vol. 23, no. 1, pp. 475–507, Jan. 2013, doi: 10.1137/110849468.
- [283] P. L. Combettes and J.-C. Pesquet, “Primal-Dual Splitting Algorithm for Solving Inclusions with Mixtures of Composite, Lipschitzian, and Parallel-Sum Type Monotone Operators,” *Set-Valued Var. Anal.*, vol. 20, no. 2, pp. 307–330, Jun. 2012, doi:

- 10.1007/s11228-011-0191-y.
- [284] A. Chambolle and T. Pock, “A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging,” *J. Math. Imaging Vis.*, vol. 40, no. 1, pp. 120–145, May 2011, doi: 10.1007/s10851-010-0251-1.
- [285] Y. Cui, X. Li, D. Sun, and K.-C. Toh, “On the Convergence Properties of a Majorized Alternating Direction Method of Multipliers for Linearly Constrained Convex Optimization Problems with Coupled Objective Functions,” *J. Optim. Theory Appl.*, vol. 169, no. 3, pp. 1013–1041, Jun. 2016, doi: 10.1007/s10957-016-0877-2.
- [286] Z. Lin, R. Liu, and Z. Su, “Linearized Alternating Direction Method with Adaptive Penalty for Low-Rank Representation,” in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2011, pp. 612–620.
- [287] G. H.-G. Chen and R. T. Rockafellar, “Convergence Rates in Forward--Backward Splitting,” *SIAM J. Optim.*, vol. 7, no. 2, pp. 421–444, May 1997, doi: 10.1137/S1052623495290179.
- [288] J. Eckstein and D. P. Bertsekas, “On the Douglas—Rachford splitting method and the proximal point algorithm for maximal monotone operators,” *Math. Program.*, vol. 55, no. 1–3, pp. 293–318, Apr. 1992, doi: 10.1007/BF01581204.
- [289] H. Raguet, J. Fadili, and G. Peyré, “A Generalized Forward-Backward Splitting,” *SIAM J. Imaging Sci.*, vol. 6, no. 3, pp. 1199–1226, Jan. 2013, doi: 10.1137/120872802.
- [290] Q. Li and N. Zhang, “Fast proximity-gradient algorithms for structured convex optimization problems,” *Appl. Comput. Harmon. Anal.*, vol. 41, no. 2, pp. 491–517, Sep. 2016, doi: 10.1016/J.ACHA.2015.11.004.
- [291] J. Li, N. Ou, G. Lin, and W. Wei, “Compressive Sensing Based Stochastic Economic Dispatch With High Penetration Renewables,” *IEEE Trans. Power Syst.*, vol. 34, no. 2, pp. 1438–1449, 2019.
- [292] L. Yao *et al.*, “Compressive representation for device-free activity recognition with passive RFID signal strength,” *IEEE Trans. Mob. Comput.*, vol. 17, no. 2, pp. 293–306, 2018.
- [293] T. Qiao *et al.*, “Joint bilateral filtering and spectral similarity-based sparse representation: a generic framework for effective feature extraction and data classification in

- hyperspectral imaging,” *Pattern Recognit.*, vol. 77, pp. 316–328, 2018.
- [294] M. M. Abo-Zahhad, A. I. Hussein, and A. M. Mohamed, “Compressive sensing algorithms for signal processing applications: A survey,” *Int. J. Commun. Netw. Syst. Sci.*, vol. 8, no. 06, p. 197, 2015.
- [295] Z. Chen *et al.*, “Vehicle detection in high-resolution aerial images via sparse representation and superpixels,” *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 1, pp. 103–116, 2016.
- [296] Z. Chen *et al.*, “Vehicle detection in high-resolution aerial images based on fast sparse representation classification and multiorder feature,” *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 8, pp. 2296–2309, 2016.
- [297] A. Farcomeni and L. Greco, *Robust methods for data reduction*. Chapman and Hall/CRC, 2016.
- [298] M. Fan *et al.*, “Android malware familial classification and representative sample selection via frequent subgraph analysis,” *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 8, pp. 1890–1905, 2018.
- [299] X. Zhang, C. Mei, D. Chen, and Y. Yang, “A fuzzy rough set-based feature selection method using representative instances,” *Knowledge-Based Syst.*, vol. 151, pp. 216–229, 2018.
- [300] X. Sun, L. Liu, C. Geng, and S. Yang, “Fast Data Reduction with Granulation based Instances Importance Labeling,” *IEEE Access*, 2018.
- [301] G. Dong and H. Liu, *Feature engineering for machine learning and data analytics*. .
- [302] H. Oh, J. H. Jung, B. C. Jeon, and B. D. Youn, “Scalable and unsupervised feature engineering using vibration-imaging and deep learning for rotor system diagnosis,” *IEEE Trans. Ind. Electron.*, vol. 65, no. 4, pp. 3539–3549, 2018.
- [303] J. Zhang, F. Fogelman-Soulié, and C. Largeron, “Towards Automatic Complex Feature Engineering,” in *International Conference on Web Information Systems Engineering*, 2018, pp. 312–322.
- [304] G. Katz, E. C. R. Shin, and D. Song, “Explorekit: Automatic feature generation and selection,” in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, 2016, pp. 979–984.
- [305] F. Nargesian, H. Samulowitz, U. Khurana, E. B. Khalil, and D. S. Turaga, “Learning

- Feature Engineering for Classification,,” in *IJCAI*, 2017, pp. 2529–2535.
- [306] C. Fan, Y. Sun, Y. Zhao, M. Song, and J. Wang, “Deep learning-based feature engineering methods for improved building energy prediction,” *Appl. Energy*, vol. 240, pp. 35–45, 2019.
- [307] R. Razavi, A. Gharipour, M. Fleury, and I. J. Akpan, “A practical feature-engineering framework for electricity theft detection in smart grids,” *Appl. Energy*, vol. 238, pp. 481–494, 2019.
- [308] H. T. Lam, J.-M. Thiebaud, M. Sinn, B. Chen, T. Mai, and O. Alkan, “One button machine for automating feature engineering in relational databases,” *arXiv Prepr. arXiv1706.00327*, 2017.
- [309] U. Khurana, H. Samulowitz, and D. Turaga, “Feature engineering for predictive modeling using reinforcement learning,” in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [310] J. Duan, Z. Zeng, A. Oprea, and S. Vasudevan, “Automated Generation and Selection of Interpretable Features for Enterprise Security,” in *2018 IEEE International Conference on Big Data (Big Data)*, 2018, pp. 1258–1265.
- [311] A. Kaul, S. Maheshwary, and V. Pudi, “Autolearn—Automated feature generation and selection,” in *2017 IEEE International Conference on Data Mining (ICDM)*, 2017, pp. 217–226.
- [312] F. Nargesian, U. Khurana, T. Pedapati, H. Samulowitz, and D. Turaga, “Dataset Evolver: An Interactive Feature Engineering Notebook,” in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [313] U. Khurana, H. Samulowitz, and D. Turaga, “Ensembles with Automated Feature Engineering,” in *ICML Workshop on AutoML*, 2018.
- [314] L. Mou, P. Ghamisi, and X. X. Zhu, “Unsupervised spectral--spatial feature learning via deep residual conv--deconv network for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 1, pp. 391–406, 2018.
- [315] Y. Duan, J. Lu, J. Feng, and J. Zhou, “Context-aware local binary feature learning for face recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1139–1153, 2018.
- [316] J. Shi, X. Zheng, Y. Li, Q. Zhang, and S. Ying, “Multimodal neuroimaging feature

- learning with multimodal stacked deep polynomial networks for diagnosis of Alzheimer's disease," *IEEE J. Biomed. Heal. informatics*, vol. 22, no. 1, pp. 173–183, 2018.
- [317] P. Li, Z. Chen, L. T. Yang, Q. Zhang, and M. J. Deen, "Deep convolutional computation model for feature learning on big data in Internet of Things," *IEEE Trans. Ind. Informatics*, vol. 14, no. 2, pp. 790–798, 2018.
- [318] J. Pan, Y. Zi, J. Chen, Z. Zhou, and B. Wang, "LiftingNet: A novel deep learning network with layerwise feature learning from noisy mechanical data for fault classification," *IEEE Trans. Ind. Electron.*, vol. 65, no. 6, pp. 4973–4982, 2018.
- [319] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: A new perspective," *Neurocomputing*, vol. 300, pp. 70–79, 2018.
- [320] M. Mafarja *et al.*, "Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems," *Knowledge-Based Syst.*, vol. 145, pp. 25–45, 2018.
- [321] M. Luo, F. Nie, X. Chang, Y. Yang, A. G. Hauptmann, and Q. Zheng, "Adaptive unsupervised feature selection with structure regularization," *IEEE Trans. neural networks Learn. Syst.*, vol. 29, no. 4, pp. 944–956, 2018.
- [322] S. Aljawarneh, M. Aldwairi, and M. B. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," *J. Comput. Sci.*, vol. 25, pp. 152–160, 2018.
- [323] M. Mafarja *et al.*, "Binary dragonfly optimization for feature selection using time-varying transfer functions," *Knowledge-Based Syst.*, vol. 161, pp. 185–204, 2018.
- [324] H. Faris *et al.*, "An efficient binary salp swarm algorithm with crossover scheme for feature selection problems," *Knowledge-Based Syst.*, vol. 154, pp. 43–67, 2018.
- [325] C. Wang, Q. Hu, X. Wang, D. Chen, Y. Qian, and Z. Dong, "Feature selection based on neighborhood discrimination index," *IEEE Trans. neural networks Learn. Syst.*, vol. 29, no. 7, pp. 2986–2999, 2018.
- [326] M. Mafarja and S. Mirjalili, "Whale optimization approaches for wrapper feature selection," *Appl. Soft Comput.*, vol. 62, pp. 441–453, 2018.
- [327] M. A. Schmitz *et al.*, "Wasserstein dictionary learning: Optimal transport-based unsupervised nonlinear dictionary learning," *SIAM J. Imaging Sci.*, vol. 11, no. 1, pp. 643–678, 2018.

-
- [328] J. Tang, Z. Li, H. Lai, L. Zhang, S. Yan, and others, “Personalized age progression with bi-level aging dictionary learning,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 905–917, 2018.
- [329] P. Zhou, C. Fang, Z. Lin, C. Zhang, and E. Y. Chang, “Dictionary learning with structured noise,” *Neurocomputing*, vol. 273, pp. 414–423, 2018.
- [330] E. Zheng, D. Ji, E. Dunn, and J.-M. Frahm, “Self-expressive dictionary learning for dynamic 3d reconstruction,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 9, pp. 2223–2237, 2018.
- [331] D. Schmitter and M. Unser, “Landmark-Based Shape Encoding and Sparse-Dictionary Learning in the Continuous Domain,” *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 365–378, 2018.
- [332] J. Sulam, V. Pappas, Y. Romano, and M. Elad, “Multilayer convolutional sparse modeling: Pursuit and dictionary learning,” *IEEE Trans. Signal Process.*, vol. 66, no. 15, pp. 4090–4104, 2018.
- [333] B. Yang, A. J. Ma, and P. C. Yuen, “Domain-Shared Group-Sparse Dictionary Learning for Unsupervised Domain Adaptation,” in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [334] W. Zhang *et al.*, “Experimental comparisons of sparse dictionary learning and independent component analysis for brain network inference from fMRI data,” *IEEE Trans. Biomed. Eng.*, vol. 66, no. 1, pp. 289–299, 2019.
- [335] T. Zhang, W. Jia, C. Gong, J. Sun, and X. Song, “Semi-supervised dictionary learning via local sparse constraints for violence detection,” *Pattern Recognit. Lett.*, vol. 107, pp. 98–104, 2018.
- [336] C. G. Drury and J. Watson, “Good practices in visual inspection,” *Hum. factors Aviat. maintenance-phase nine, Prog. report, FAA/Human Factors Aviat. Maintenance.* @ URL <http://hfskyway.faa.gov>, 2002.
- [337] O. Dictionaries, “Oxford dictionaries,” *Oxford Univ. Press.* <http://oxforddictionaries.com/definition/english/VAR>. Accessed Oct, vol. 15, p. 2012, 2010.
- [338] T. S. Newman and A. K. Jain, “A Survey of Automated Visual Inspection,” *Comput. Vis. Image Underst.*, vol. 61, no. 2, pp. 231–262, 1995, doi: <http://dx.doi.org/10.1006/cviu.1995.1017>.

-
- [339] J. W. Schoonahd, J. D. Gould, and L. A. Miller, "Studies of visual inspection," *Ergonomics*, vol. 16, no. 4, pp. 365–379, 1973.
- [340] J. E. See, "Visual inspection of aircraft (AC-43-204)," 1997.
- [341] A. J. McEvily, "Failures in inspection procedures: case studies," *Eng. Fail. Anal.*, vol. 11, no. 2, pp. 167–176, 2004.
- [342] N. Yamamoto, K. Tanaka, and A. Hisano, "Method and device for supporting the repair of defective substrates." Google Patents, 1998.
- [343] K. G. Harding, J. W. Devitt, N. R. Corby Jr, and K. H. V. Hedengren, "Inspection and sorting system and method for part repair." Google Patents, 2004.
- [344] J. E. See, "Visual Inspection: A Review of the Literature," 2012.
- [345] M. Kumru, "Assessing the visual quality of sanitary ware by fuzzy logic," *Appl. Soft Comput.*, vol. 13, no. 8, pp. 3646–3656, 2013.
- [346] R. Gong, M. Chu, A. Wang, and Y. Yang, "A Fast Detection Method for Region of Defect on Strip Steel Surface," *ISIJ Int.*, vol. 55, no. 1, pp. 207–212, 2015.
- [347] A. Rebhi, S. Abid, and F. Fnaiech, "Texture defect detection using local homogeneity and discrete cosine transform," *World Appl. Sci. J.*, vol. 31, no. 9, pp. 1677–1683, 2014.
- [348] C.-C. Wang, B. C. Jiang, J.-Y. Lin, and C.-C. Chu, "Machine vision-based defect detection in IC images using the partial information correlation coefficient," *Semicond. Manuf. IEEE Trans.*, vol. 26, no. 3, pp. 378–384, 2013.
- [349] J. L. Raheja, B. Ajay, and A. Chaudhary, "Real time fabric defect detection system on an embedded DSP platform," *Opt. J. Light Electron Opt.*, vol. 124, no. 21, pp. 5280–5284, 2013.
- [350] M. Chu and R. Gong, "Invariant Feature Extraction Method Based on Smoothed Local Binary Pattern for Strip Steel Surface Defect," *ISIJ Int.*, vol. 55, no. 9, pp. 1956–1962, 2015.
- [351] K. Ogawa, J. Kodama, K. Machida, K. Nakashima, and Y. Watakabe, "Phase Defect Inspection by Differential Interference," in *Photomask and Next Generation Lithography Mask Technology VIII*, 2001, pp. 543–554.
- [352] H. Zheng, L. X. Kong, and S. Nahavandi, "Automatic inspection of metallic surface defects using genetic algorithms," *J. Mater. Process. Technol.*, vol. 125, pp. 427–433, 2002.

-
- [353] I. Kunttu, L. Lepistö, J. Rauhamaa, and A. Visa, “Multiscale Fourier descriptors for defect image retrieval,” *Pattern Recognit. Lett.*, vol. 27, no. 2, pp. 123–132, 2006.
- [354] X. Xie, M. Mirmehdi, and B. Thomas, “Colour tonality inspection using eigenspace features,” *Mach. Vis. Appl.*, vol. 16, no. 6, pp. 364–373, 2006.
- [355] H. Y. T. Ngan, G. K. H. Pang, and N. H. C. Yung, “Motif-based defect detection for patterned fabric,” *Pattern Recognit.*, vol. 41, no. 6, pp. 1878–1894, 2008.
- [356] H. Y. T. Ngan, G. K. H. Pang, and N. H. C. Yung, “Performance evaluation for motif-based patterned texture defect detection,” *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 1, pp. 58–72, 2010.
- [357] M. Sharifzadeh, S. Alirezaee, R. Amirfattahi, and S. Sadri, “Detection of steel defect using the image processing algorithms,” in *Multitopic Conference, 2008. INMIC 2008. IEEE International*, 2008, pp. 125–127.
- [358] S. Jahanbin, A. C. Bovik, E. Pérez, and D. Nair, “Automatic inspection of textured surfaces by support vector machines,” in *SPIE Optical Engineering+ Applications*, 2009, pp. 74320A--74320A.
- [359] D. Aiger and H. Talbot, “The phase only transform for unsupervised surface defect detection,” in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 2010, pp. 295–302.
- [360] S. Ravikumar, K. I. Ramachandran, and V. Sugumaran, “Machine learning approach for automated visual inspection of machine components,” *Expert Syst. Appl.*, vol. 38, no. 4, pp. 3260–3266, 2011.
- [361] V. Asha, N. U. Bhajantri, and P. Nagabhushan, “Automatic detection of texture defects using texture-periodicity and Gabor wavelets,” in *Computer Networks and Intelligent Computing*, Springer, 2011, pp. 548–553.
- [362] Z. Xue-Wu, D. Yan-Qiong, L. Yan-Yun, S. Ai-Ye, and L. Rui-Yu, “A vision inspection system for the surface defects of strongly reflected metal based on multi-class SVM,” *Expert Syst. Appl.*, vol. 38, no. 5, pp. 5930–5939, 2011.
- [363] J. Masci, U. Meier, D. Ciresan, J. Schmidhuber, and G. Fricout, “Steel defect classification with max-pooling convolutional neural networks,” in *Neural Networks (IJCNN), The 2012 International Joint Conference on*, 2012, pp. 1–6.
- [364] S. Singh and M. Bharti, “Image processing based automatic visual inspection system for

- PCBs,” *ISOR J. Eng.*, vol. 2, pp. 1451–1455, 2012.
- [365] V. Asha, N. U. Bhajantri, and P. Nagabhushan, “Automatic Detection of Texture-defects using Texture-periodicity and Jensen-Shannon Divergence,” *J. Inf. Process. Syst.*, vol. 8, no. 2, pp. 359–374, 2012.
- [366] C. Benedek, O. Krammer, M. Janóczki, and L. Jakab, “Solder paste scooping detection by multilevel visual inspection of printed circuit boards,” *Ind. Electron. IEEE Trans.*, vol. 60, no. 6, pp. 2318–2331, 2013.
- [367] S. Ghorai, A. Mukherjee, M. Gangadaran, and P. K. Dutta, “Automatic Defect Detection on Hot-Rolled Flat Steel Products,” *IEEE Trans. Instrum. Meas.*, vol. 62, no. 3, pp. 612–621, Mar. 2013, doi: 10.1109/TIM.2012.2218677.
- [368] M. S. SANKARI, P. R, R. K, and T. A, “A novel method for identification of defects in fabric,” *Int. J. Electron. Commun. Eng.*, vol. 3, no. 4, pp. 37–44, 2014.
- [369] S. Guan and Z. Gao, “Fabric defect image segmentation based on visual attention mechanism of wavelet domain,” *Text. Res. J.*, p. 0040517513517964, 2014.
- [370] X. Li, H. Jiang, and G. Yin, “Detection of surface crack defects on ferrite magnetic tile,” *NDT E Int.*, vol. 62, pp. 6–13, 2014.
- [371] Y.-J. Jeon, D. Choi, S. J. Lee, J. P. Yun, and S. W. Kim, “Defect detection for corner cracks in steel billets using a wavelet reconstruction method,” *JOSA A*, vol. 31, no. 2, pp. 227–237, 2014.
- [372] H. W. Kim and S. I. Yoo, “Defect detection using feature point matching for non-repetitive patterned images,” *Pattern Anal. Appl.*, vol. 17, no. 2, pp. 415–429, 2014.
- [373] A. Chondronasios, “Investigation of surface defects for extruded aluminium profiles using pattern recognition techniques,” University of Portsmouth, 2015.
- [374] A. S. Tolba and H. M. Raafat, “Multiscale image quality measures for defect detection in thin films,” *Int. J. Adv. Manuf. Technol.*, vol. 79, no. 1–4, pp. 113–122, 2015.
- [375] M. Liu, Y. Liu, H. Hu, and L. Nie, “Genetic algorithm and mathematical morphology based binarization method for strip steel defect image with non-uniform illumination,” *J. Vis. Commun. Image Represent.*, 2015.
- [376] S. Guan, “Strip Steel Defect Detection Based on Saliency Map Construction Using Gaussian Pyramid Decomposition,” *ISIJ Int.*, vol. 55, no. 9, pp. 1950–1955, 2015.
- [377] Y. Park and I. S. Kweon, “Ambiguous Surface Defect Image Classification of AMOLED

- Displays in Smartphones,” *IEEE Trans. Ind. Informatics*, vol. 12, no. 2, pp. 597–607, Apr. 2016, doi: 10.1109/TII.2016.2522191.
- [378] K.-W. Liao and Y.-T. Lee, “Detection of rust defects on steel bridge coatings via digital image recognition,” *Autom. Constr.*, vol. 71, pp. 294–306, 2016, doi: <https://doi.org/10.1016/j.autcon.2016.08.008>.
- [379] H. Zakeri, F. M. Nejad, and A. Fahimifar, “Rahbin: A quadcopter unmanned aerial vehicle based on a systematic image processing approach toward an automated asphalt pavement inspection,” *Autom. Constr.*, vol. 72, pp. 211–235, 2016, doi: <https://doi.org/10.1016/j.autcon.2016.09.002>.
- [380] Y.-J. Jeon, D. Choi, S. J. Lee, J. P. Yun, and S. W. Kim, “Steel-surface defect detection using a switching-lighting scheme,” *Appl. Opt.*, vol. 55, no. 1, pp. 47–57, 2016.
- [381] W. Jin, W. Lin, X. Yang, and H. Gao, “Reference-Free Path-Walking Method for Ball Grid Array Inspection in Surface Mounting Machines,” *IEEE Trans. Ind. Electron.*, vol. 64, no. 8, pp. 6310–6318, Aug. 2017, doi: 10.1109/TIE.2017.2682008.
- [382] H. Kong, J. Yang, and Z. Chen, “Accurate and Efficient Inspection of Speckle and Scratch Defects on Surfaces of Planar Products,” *IEEE Trans. Ind. Informatics*, vol. 13, no. 4, pp. 1855–1865, Aug. 2017, doi: 10.1109/TII.2017.2668438.
- [383] D. Carrera, F. Manganini, G. Boracchi, and E. Lanzarone, “Defect Detection in SEM Images of Nanofibrous Materials,” *IEEE Trans. Ind. Informatics*, vol. 13, no. 2, pp. 551–561, Apr. 2017, doi: 10.1109/TII.2016.2641472.
- [384] A. R. Rizvi, P. R. Khan, and S. Ahmad, “Crack detection in railway track using image processing,” *Int. J. Adv. Res. Ideas Innov. Technol.*, vol. 3, no. 4, pp. 489–496, 2017.
- [385] J.-H. Chen, M.-C. Su, R. Cao, S.-C. Hsu, and J.-C. Lu, “A self organizing map optimization based image recognition and processing model for bridge crack inspection,” *Autom. Constr.*, vol. 73, pp. 58–66, 2017, doi: <https://doi.org/10.1016/j.autcon.2016.08.033>.
- [386] T. Dawood, Z. Zhu, and T. Zayed, “Machine vision-based model for spalling detection and quantification in subway networks,” *Autom. Constr.*, vol. 81, pp. 149–160, 2017, doi: <https://doi.org/10.1016/j.autcon.2017.06.008>.
- [387] Y. O. Ouma and M. Hahn, “Pothole detection on asphalt pavements from 2D-colour pothole images using fuzzy c-means clustering and morphological reconstruction,” *Autom.*

-
- Constr.*, vol. 83, pp. 196–211, 2017, doi: <https://doi.org/10.1016/j.autcon.2017.08.017>.
- [388] R. Ren, T. Hung, and K. C. Tan, “A Generic Deep-Learning-Based Approach for Automated Surface Inspection,” *IEEE Trans. Cybern.*, vol. 48, no. 3, pp. 929–940, Mar. 2018, doi: 10.1109/TCYB.2017.2668395.
- [389] L. Bai, X. Yang, and H. Gao, “A Novel Coarse–Fine Method for Ball Grid Array Component Positioning and Defect Inspection,” *IEEE Trans. Ind. Electron.*, vol. 65, no. 6, pp. 5023–5031, Jun. 2018, doi: 10.1109/TIE.2017.2748053.
- [390] F. Chen and M. R. Jahanshahi, “NB-CNN: Deep Learning-Based Crack Detection Using Convolutional Neural Network and Naïve Bayes Data Fusion,” *IEEE Trans. Ind. Electron.*, vol. 65, no. 5, pp. 4392–4400, May 2018, doi: 10.1109/TIE.2017.2764844.
- [391] R. Li, Y. Yuan, W. Zhang, and Y. Yuan, “Unified Vision-Based Methodology for Simultaneous Concrete Defect Detection and Geolocalization,” *Comput. Civ. Infrastruct. Eng.*, vol. 33, no. 7, pp. 527–544, 2018, doi: 10.1111/mice.12351.
- [392] H. y. Wang, J. Zhang, Y. Tian, H. y. Chen, H. x. Sun, and K. Liu, “A simple guidance template-based defect detection method for strip steel surfaces,” *IEEE Trans. Ind. Informatics*, p. 1, 2018, doi: 10.1109/TII.2018.2887145.
- [393] S. Martelli *et al.*, “Deep Endoscope: Intelligent Duct Inspection for the Avionic Industry,” *IEEE Trans. Ind. Informatics*, vol. 14, no. 4, pp. 1701–1711, Apr. 2018, doi: 10.1109/TII.2018.2807797.
- [394] J. Gan, J. Wang, H. Yu, Q. Li, and Z. Shi, “Online Rail Surface Inspection Utilizing Spatial Consistency and Continuity,” *IEEE Trans. Syst. Man, Cybern. Syst.*, pp. 1–11, 2018, doi: 10.1109/TSMC.2018.2827937.
- [395] Q. Li *et al.*, “A cyber-enabled visual inspection system for rail corrugation,” *Futur. Gener. Comput. Syst.*, vol. 79, pp. 374–382, 2018, doi: <https://doi.org/10.1016/j.future.2017.04.032>.
- [396] \cSaban Öztürk and B. Akdemir, “Fuzzy logic-based segmentation of manufacturing defects on reflective surfaces,” *Neural Comput. Appl.*, vol. 29, no. 8, pp. 107–116, Apr. 2018, doi: 10.1007/s00521-017-2862-6.
- [397] A. Hawari, M. Alamin, F. Alkadour, M. Elmasry, and T. Zayed, “Automated defect detection tool for closed circuit television (cctv) inspected sewer pipelines,” *Autom. Constr.*, vol. 89, pp. 99–109, 2018, doi: <https://doi.org/10.1016/j.autcon.2018.01.004>.

-
- [398] S. S. Kumar, D. M. Abraham, M. R. Jahanshahi, T. Iseley, and J. Starr, “Automated defect classification in sewer closed circuit television inspections using deep convolutional neural networks,” *Autom. Constr.*, vol. 91, pp. 273–283, 2018, doi: <https://doi.org/10.1016/j.autcon.2018.03.028>.
- [399] J. Myrans, R. Everson, and Z. Kapelan, “Automated detection of faults in sewers using CCTV image sequences,” *Autom. Constr.*, vol. 95, pp. 64–71, 2018, doi: <https://doi.org/10.1016/j.autcon.2018.08.005>.
- [400] J. Sun, C. Li, X. Wu, V. Palade, and W. Fang, “An Effective Method of Weld Defect Detection and Classification Based on Machine Vision,” *IEEE Trans. Ind. Informatics*, p. 1, 2019, doi: 10.1109/TII.2019.2896357.
- [401] E. Essa, M. S. Hossain, A. S. Tolba, H. M. Raafat, S. Elmogy, and G. Muahmmad, “Toward cognitive support for automated defect detection,” *Neural Comput. Appl.*, Jan. 2019, doi: 10.1007/s00521-018-03969-x.
- [402] C. V. Dung and others, “Autonomous concrete crack detection using deep fully convolutional neural network,” *Autom. Constr.*, vol. 99, pp. 52–58, 2019.
- [403] N. Wang, X. Zhao, P. Zhao, Y. Zhang, Z. Zou, and J. Ou, “Automatic damage detection of historic masonry buildings based on mobile deep learning,” *Autom. Constr.*, vol. 103, pp. 53–66, 2019, doi: <https://doi.org/10.1016/j.autcon.2019.03.003>.
- [404] C. V. Dung, H. Sekiya, S. Hirano, T. Okatani, and C. Miki, “A vision-based method for crack detection in gusset plate welded joints of steel bridges using deep convolutional neural networks,” *Autom. Constr.*, vol. 102, pp. 217–229, 2019, doi: <https://doi.org/10.1016/j.autcon.2019.02.013>.
- [405] C. G. Broyden, “A new double-rank minimization algorithm,” *Not. Am. Math. Soc.*, vol. 16, p. 670, 1969.
- [406] R. Fletcher, “A new approach to variable metric algorithms,” *Comput. J.*, vol. 13, no. 3, pp. 317–322, 1970.
- [407] D. Goldfarb, “A family of variable-metric methods derived by variational means,” *Math. Comput.*, vol. 24, no. 109, pp. 23–23, Jan. 1970, doi: 10.1090/S0025-5718-1970-0258249-6.
- [408] D. F. Shanno, “Conditioning of quasi-Newton methods for function minimization,” *Math. Comput.*, vol. 24, no. 111, pp. 647–647, Sep. 1970, doi: 10.1090/S0025-5718-1970-

- 0274029-X.
- [409] N. Parikh, S. Boyd, and others, “Proximal algorithms,” *Found. Trends®in Optim.*, vol. 1, no. 3, pp. 127–239, 2014.
 - [410] H. H. Bauschke, R. Goebel, Y. Lucet, and X. Wang, “The Proximal Average: Basic Theory,” *SIAM J. Optim.*, vol. 19, no. 2, pp. 766–785, Jan. 2008, doi: 10.1137/070687542.
 - [411] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, 2007.
 - [412] J. Sherman, “Adjustment of an inverse matrix corresponding to changes in the elements of a given column or a given row of the original matrix,” *Ann. Math. Stat.*, vol. 20, no. 4, p. 621, 1949.
 - [413] J. Sherman and W. J. Morrison, “Adjustment of an inverse matrix corresponding to a change in one element of a given matrix,” *Ann. Math. Stat.*, vol. 21, no. 1, pp. 124–127, 1950.
 - [414] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, “Section 2.7. 1 Sherman--Morrison Formula,” *Numer. Recipes Art Sci. Comput. (3rd ed.)*. Cambridge Univ. Press. New York, vol. 1, 2007.
 - [415] J. M. Steele, *The Cauchy-Schwarz master class: an introduction to the art of mathematical inequalities*. Cambridge University Press, 2004.
 - [416] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer, 2006.
 - [417] J.-F. Bonnans, J. C. Gilbert, C. Lemaréchal, and C. A. Sagastizábal, *Numerical optimization: theoretical and practical aspects*. Springer Science & Business Media, 2006.
 - [418] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
 - [419] S. Bubeck, “Convex Optimization: Algorithms and Complexity,” *Found. Trends® Mach. Learn.*, vol. 8, no. 3–4, pp. 231–357, 2015, doi: 10.1561/22000000050.
 - [420] L. Zhao *et al.*, “Improved imaging performance in super-resolution localization microscopy by YALL1 method,” *IEEE Access*, vol. 6, pp. 5438–5446, 2018.
 - [421] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth, “On the LambertW function,” *Adv. Comput. Math.*, vol. 5, no. 1, pp. 329–359, 1996.
 - [422] M. Rudelson and R. Vershynin, “On sparse reconstruction from Fourier and Gaussian measurements,” *Commun. Pure Appl. Math.*, vol. 61, no. 8, pp. 1025–1045, 2008.

-
- [423] D. L. Donoho and J. Tanner, “Exponential bounds implying construction of compressed sensing matrices, error-correcting codes, and neighborly polytopes by random sampling,” *IEEE Trans. Inf. Theory*, vol. 56, no. 4, pp. 2002–2016, 2010.
- [424] F. Wen, L. Pei, Y. Yang, W. Yu, and P. Liu, “Efficient and Robust Recovery of Sparse Signal and Image Using Generalized Nonconvex Regularization,” *IEEE Trans. Comput. Imaging*, vol. 3, no. 4, pp. 566–579, Dec. 2017, doi: 10.1109/TCI.2017.2744626.
- [425] K. Lee, D. J. Kriegman, and J. Ho, “Acquiring Linear Subspaces for Face Recognition under Variable Lighting,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, pp. 684–698, 2005, doi: 10.1109/TPAMI.2005.92.
- [426] J. Gan, G. Wen, H. Yu, W. Zheng, and C. Lei, “Supervised feature selection by self-paced learning regression,” *Pattern Recognit. Lett.*, vol. 132, pp. 30–37, 2020.
- [427] N. Bi, J. Tan, J.-H. Lai, and C. Y. Suen, “High-dimensional supervised feature selection via optimized kernel mutual information,” *Expert Syst. Appl.*, vol. 108, pp. 81–95, 2018.
- [428] Y. Zhu, X. Zhang, R. Hu, and G. Wen, “Adaptive structure learning for low-rank supervised feature selection,” *Pattern Recognit. Lett.*, vol. 109, pp. 89–96, 2018.
- [429] K. Liu, X. Yang, H. Yu, J. Mi, P. Wang, and X. Chen, “Rough set based semi-supervised feature selection via ensemble selector,” *Knowledge-Based Syst.*, vol. 165, pp. 282–296, 2019.
- [430] S. Chen, Y. Wang, C.-T. Lin, W. Ding, and Z. Cao, “Semi-supervised feature learning for improving writer identification,” *Inf. Sci. (Ny)*, vol. 482, pp. 156–170, 2019.
- [431] Z. Liu, Z. Lai, W. Ou, K. Zhang, and R. Zheng, “Structured optimal graph based sparse feature extraction for semi-supervised learning,” *Signal Processing*, vol. 170, p. 107456, 2020.
- [432] C. Tang *et al.*, “Unsupervised feature selection via latent representation learning and manifold regularization,” *Neural Networks*, vol. 117, pp. 163–178, 2019.
- [433] Y. Zhang, Q. Wang, D. Gong, and X. Song, “Nonnegative Laplacian embedding guided subspace learning for unsupervised feature selection,” *Pattern Recognit.*, vol. 93, pp. 337–352, 2019.
- [434] Z. Manbari, F. AkhlaghianTab, and C. Salavati, “Hybrid fast unsupervised feature selection for high-dimensional data,” *Expert Syst. Appl.*, vol. 124, pp. 97–118, 2019.
- [435] D. Huang, J.-H. Lai, and C.-D. Wang, “Robust ensemble clustering using probability

- trajectories,” *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 5, pp. 1312–1326, 2015.
- [436] Y.-M. Xu, C.-D. Wang, and J.-H. Lai, “Weighted multi-view clustering with feature selection,” *Pattern Recognit.*, vol. 53, pp. 25–35, 2016.
- [437] D. Huang, J. Lai, and C.-D. Wang, “Ensemble clustering using factor graph,” *Pattern Recognit.*, vol. 50, pp. 131–142, 2016.
- [438] D. Huang, C.-D. Wang, and J.-H. Lai, “Locally weighted ensemble clustering,” *IEEE Trans. Cybern.*, vol. 48, no. 5, pp. 1460–1473, 2017.
- [439] D. Huang, C.-D. Wang, H. Peng, J. Lai, and C.-K. Kwok, “Enhanced ensemble clustering via fast propagation of cluster-wise similarities,” *IEEE Trans. Syst. Man, Cybern. Syst.*, 2018.
- [440] D. Huang, C.-D. Wang, J.-S. Wu, J.-H. Lai, and C.-K. Kwok, “Ultra-scalable spectral clustering and ensemble clustering,” *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 6, pp. 1212–1226, 2019.
- [441] G. Moretti, R. Sprugnoli, S. Menini, and S. Tonelli, “ALCIDE: Extracting and visualising content from large document collections to support humanities studies,” *Knowledge-Based Syst.*, vol. 111, pp. 100–112, 2016.
- [442] W. Yan, B. Zhang, S. Ma, and Z. Yang, “A novel regularized concept factorization for document clustering,” *Knowledge-Based Syst.*, vol. 135, pp. 147–158, 2017.
- [443] C.-H. Chen, “A non-parametric feature assessment mechanism by identifying representative neighbors for image clustering,” *Knowledge-Based Syst.*, vol. 67, pp. 328–341, 2014.
- [444] X. Qian, M. Li, Y. Ren, and S. Jiang, “Social media based event summarization by user--text--image co-clustering,” *Knowledge-Based Syst.*, vol. 164, pp. 107–121, 2019.
- [445] M. Dash, K. Choi, P. Scheuermann, and H. Liu, “Feature selection for clustering-a filter solution,” in *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, 2002, pp. 115–122.
- [446] V. Roth and T. Lange, “Feature selection in clustering problems,” in *Advances in neural information processing systems*, 2004, pp. 473–480.
- [447] S. Yang, S. Yan, C. Zhang, and X. Tang, “Bilinear analysis for kernel selection and nonlinear feature extraction,” *IEEE Trans. neural networks*, vol. 18, no. 5, pp. 1442–1452, 2007.

-
- [448] D. Cai, C. Zhang, and X. He, “Unsupervised feature selection for multi-cluster data,” in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010, pp. 333–342.
- [449] Z. Li, Y. Yang, J. Liu, X. Zhou, and H. Lu, “Unsupervised feature selection using nonnegative spectral analysis,” in *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.
- [450] M. Qian and C. Zhai, “Robust unsupervised feature selection,” in *Twenty-third international joint conference on artificial intelligence*, 2013.
- [451] D. Han and J. Kim, “Unsupervised simultaneous orthogonal basis clustering feature selection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5016–5023.
- [452] S. Wang, J. Tang, and H. Liu, “Embedded unsupervised feature selection,” in *Twenty-ninth AAAI conference on artificial intelligence*, 2015.
- [453] F. Nie, W. Zhu, and X. Li, “Unsupervised feature selection with structured graph optimization,” in *Proceedings of the Thirtieth AAAI conference on artificial intelligence*, 2016, pp. 1302–1308.
- [454] H. Liu, M. Shao, and Y. Fu, “Consensus guided unsupervised feature selection,” in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2016, pp. 1874–1880.
- [455] J. Guo, Y. Guo, X. Kong, and R. He, “Unsupervised feature selection with ordinal locality,” in *2017 IEEE International Conference on Multimedia and Expo (ICME)*, Jul. 2017, pp. 1213–1218, doi: 10.1109/ICME.2017.8019357.
- [456] D. Huang, X. Cai, and C.-D. Wang, “Unsupervised feature selection with multi-subspace randomization and collaboration,” *Knowledge-Based Syst.*, vol. 182, p. 104856, 2019, doi: <https://doi.org/10.1016/j.knosys.2019.07.027>.
- [457] D. L. Donoho, “For most large underdetermined systems of linear equations the minimal ℓ_1 -norm solution is also the sparsest solution,” *Commun. Pure Appl. Math. A J. Issued by Courant Inst. Math. Sci.*, vol. 59, no. 6, pp. 797–829, 2006.
- [458] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt, “Support vector method for novelty detection,” in *Advances in neural information processing systems*, 2000, pp. 582–588.

-
- [459] J. Kenney, “Particle swarm optimization,” in *Proc. IEEE International Conference on Neural Networks, 1995*, 1995, pp. 1942–1948.
- [460] E. Mezura-Montes and C. A. C. Coello, “Constraint-handling in nature-inspired numerical optimization: past, present and future,” *Swarm Evol. Comput.*, vol. 1, no. 4, pp. 173–194, 2011.
- [461] M. E. H. Pedersen, “Good parameters for particle swarm optimization,” *Hvass Lab., Copenhagen, Denmark, Tech. Rep. HLI001*, 2010.
- [462] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann, “The balanced accuracy and its posterior distribution,” in *Pattern Recognition (ICPR), 2010 20th International Conference on*, 2010, pp. 3121–3124.
- [463] C. Lu, J. Shi, and J. Jia, “Online robust dictionary learning,” in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, 2013, pp. 415–422.
- [464] T. Fawcett, “An introduction to ROC analysis,” *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, 2006.
- [465] L. Wu, A. Ganesh, B. Shi, Y. Matsushita, Y. Wang, and Y. Ma, “Robust photometric stereo via low-rank matrix completion and recovery,” in *Asian Conference on Computer Vision*, 2010, pp. 703–717.
- [466] I. R. Shafarevich and A. Remizov, *Linear algebra and geometry*. Springer Science & Business Media, 2012.
- [467] M. Zhou, Y. Ji, Y. Ding, J. Ye, S. S. Young, and J. Yu, “Non-Lambertian Surface Shape and Reflectance Reconstruction Using Concentric Multi-Spectral Light Field,” *arXiv Prepr. arXiv1904.04875*, 2019.
- [468] S. Barsky and M. Petrou, “The 4-source photometric stereo technique for three-dimensional surfaces in the presence of highlights and shadows,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1239–1252, 2003.
- [469] J. Sun, M. Smith, L. Smith, S. Midha, and J. Bamber, “Object surface recovery using a multi-light photometric stereo technique for non-Lambertian surfaces subject to shadows and specularities,” *Image Vis. Comput.*, vol. 25, no. 7, pp. 1050–1057, 2007.
- [470] T. Ono, H. Kubo, K. Tanaka, T. Funatomi, and Y. Mukaigawa, “Practical BRDF reconstruction using reliable geometric regions from multi-view stereo,” *Comput. Vis. Media*, vol. 5, no. 4, pp. 325–336, 2019.

-
- [471] N. Meng, H. K.-H. So, X. Sun, and E. Lam, "High-dimensional dense residual convolutional neural network for light field reconstruction," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019.
- [472] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 3166–3173.
- [473] Y. Wei, F. Wen, W. Zhu, and J. Sun, "Geodesic saliency using background priors," in *European Conference on Computer Vision*, 2012, pp. 29–42.
- [474] H. Y. T. Ngan, G. K. H. Pang, and N. H. C. Yung, "Automated fabric defect detection—a review," *Image Vis. Comput.*, vol. 29, no. 7, pp. 442–458, 2011.
- [475] R. M. Haralick, "Statistical and structural approaches to texture," *Proc. IEEE*, vol. 67, no. 5, pp. 786–804, 1979.
- [476] A. P. S. Chauhan and S. C. Bhardwaj, "Detection of bare PCB defects by image subtraction method using machine vision," in *Proceedings of the World Congress on Engineering*, 2011, vol. 2, pp. 6–8.
- [477] D.-M. Tsai and C.-H. Chiang, "Automatic band selection for wavelet reconstruction in the application of defect detection," *Image Vis. Comput.*, vol. 21, no. 5, pp. 413–431, 2003.
- [478] D. P. Brzakovic, P. R. Bakic, N. S. Vujovic, and H. Sari-Sarraf, "A generalized development environment for inspection of web materials," in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, 1997, vol. 1, pp. 1–8.
- [479] F. Pernkopf and P. O’Leary, "Image acquisition techniques for automatic visual inspection of metallic surfaces," *NDT E Int.*, vol. 36, no. 8, pp. 609–617, 2003.
- [480] B. Zhang *et al.*, "Principles, developments and applications of computer vision for external quality inspection of fruits and vegetables: A review," *Food Res. Int.*, vol. 62, pp. 326–343, 2014.
- [481] A. Kumar, "Computer-vision-based fabric defect detection: a survey," *Ind. Electron. IEEE Trans.*, vol. 55, no. 1, pp. 348–363, 2008.
- [482] P. M. Mahajan, S. R. Kolhe, and P. M. Patil, "A review of automatic fabric defect detection techniques," *Adv Comput Res*, vol. 1, no. 2, pp. 18–29, 2009.
- [483] B. Dhande and M.R.Dhotre, "A Review of Fabric Defect Detection Techniques," *Int. J. Comput. Sci. Trends Technol.*, 2014.

-
- [484] M. Habib, R. H. Faisal, M. Rokonzaman, F. Ahmed, and others, “Automated Fabric Defect Inspection: A Survey of Classifiers,” *arXiv Prepr. arXiv1405.6177*, 2014.
 - [485] G. K. Gajanan, “Review of Defect Detection and Classification Methods for Fabric,” *Int. J. Sci. Eng. Res.*, vol. 5, no. 3, pp. 698–704, 2014.
 - [486] D. Mishra, “A Survey-Defect Detection and Classification for Fabric Texture Defects in Textile Industry,” *Int. J. Comput. Sci. Inf. Secur.*, vol. 13, no. 5, p. 48, 2015.
 - [487] X. Xie, “A review of recent advances in surface defect detection using texture analysis techniques,” *ELCVIA Electron. Lett. Comput. Vis. Image Anal.*, vol. 7, no. 3, 2008.
 - [488] N. Neogi, D. K. Mohanta, and P. K. Dutta, “Review of vision-based steel surface inspection systems,” *EURASIP J. Image Video Process.*, vol. 2014, no. 1, pp. 1–19, 2014.
 - [489] P. S. Malge and R. S. Nadaf, “A survey: automated visual pcb inspection algorithm,” in *International Journal of Engineering Research and Technology*, 2014, vol. 3, no. 1 (January-2014).
 - [490] S.-H. Huang and Y.-C. Pan, “Automated visual inspection in the semiconductor industry: A survey,” *Comput. Ind.*, vol. 66, pp. 1–10, 2015.
 - [491] K. R. Davidson and S. J. Szarek, “Local operator theory, random matrices and Banach spaces,” *Handb. Geom. Banach spaces*, vol. 1, no. 317–366, p. 131, 2001.
 - [492] M. Ledoux, *The concentration of measure phenomenon*, no. 89. American Mathematical Soc., 2001.
 - [493] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” in *European conference on computational learning theory*, 1995, pp. 23–37.
 - [494] T. Doan and J. Kalita, “Predicting run time of classification algorithms using meta-learning,” *Int. J. Mach. Learn. Cybern.*, pp. 1–15, 2016.
 - [495] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, “Meta-learning with memory-augmented neural networks,” in *International conference on machine learning*, 2016, pp. 1842–1850.
 - [496] T. Bosc, “Learning to Learn Neural Networks,” *arXiv Prepr. arXiv1610.06072*, 2016.
 - [497] Y. Chen, M. W. Hoffman, S. G. Colmenarejo, M. Denil, T. P. Lillicrap, and N. de Freitas, “Learning to Learn for Global Optimization of Black Box Functions,” *arXiv Prepr. arXiv1611.03824*, 2016.

-
- [498] G. F. C. Campos, S. Barbon Jr, and R. G. Mantovani, “A Meta-learning Approach for Recommendation of Image Segmentation Algorithms,” 2016.
- [499] S. N. das Dôres, L. Alves, D. D. Ruiz, and R. C. Barros, “A meta-learning framework for algorithm recommendation in software fault prediction,” in *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, 2016, pp. 1486–1491.
- [500] D. García-Saiz and M. Zorrilla, “A meta-learning based framework for building algorithm recommenders: An application for educational arena,” *J. Intell. Fuzzy Syst.*, vol. 32, no. 2, pp. 1449–1459, 2017, doi: 10.3233/JIFS-169141.
- [501] R. Caiuta, A. Pozo, and S. R. Vergilio, “Meta-learning based selection of software reliability models,” *Autom. Softw. Eng.*, pp. 1–28, 2016.
- [502] M. ELMahgiubi, “An Efficient Framework for Automatic Algorithm Selection using Meta-Learning,” 2016.
- [503] A. Zabashta, I. Smetannikov, and A. Filchenkov, “Rank Aggregation Algorithm Selection Meets Feature Selection,” in *Machine Learning and Data Mining in Pattern Recognition*, Springer, 2016, pp. 740–755.
- [504] T. Doan and J. Kalita, “Algorithm selection using performance and run time behavior,” in *International Conference on Artificial Intelligence: Methodology, Systems, and Applications*, 2016, pp. 3–13.
- [505] N. Pise and P. Kulkarni, “Algorithm selection for classification problems,” in *SAI Computing Conference (SAI)*, 2016, 2016, pp. 203–211.
- [506] S. Poursoltan and F. Neumann, “A Feature-Based Prediction Model of Algorithm Selection for Constrained Continuous Optimisation,” *arXiv Prepr. arXiv1602.02862*, 2016.
- [507] B. Wooley, S. Bridges, J. Hodges, and A. Skjellum, “Scaling the data mining step in knowledge discovery using oceanographic data,” in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, 2000, pp. 85–92.
- [508] D. G. Ferrari and L. N. De Castro, “Clustering algorithm selection by meta-learning systems: A new distance-based problem characterization and ranking combination methods,” *Inf. Sci. (Ny)*, vol. 301, pp. 181–194, 2015.
- [509] M. Wistuba, N. Schilling, and L. Schmidt-Thieme, “Two-Stage Transfer Surrogate Model

- for Automatic Hyperparameter Optimization,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2016, pp. 199–214.
- [510] B. Wang, “Machine learning techniques for brain signal analysis with applications on seizure detection and brain-computer interfaces,” *AI Matters*, vol. 2, no. 3, pp. 18–19, 2016.
- [511] C. Félix, C. Soares, and A. Jorge, “Can Metalearning Be Applied to Transfer on Heterogeneous Datasets?,” in *International Conference on Hybrid Artificial Intelligence Systems*, 2016, pp. 332–343.
- [512] N. R. Waytowich, V. J. Lawhern, A. W. Bohannon, K. R. Ball, and B. J. Lance, “Spectral transfer learning using information geometry for a user-independent brain-computer interface,” *Front. Neurosci.*, vol. 10, 2016.
- [513] D. H. Wolpert, “Stacked generalization,” *Neural networks*, vol. 5, no. 2, pp. 241–259, 1992.
- [514] C.-Y. Low and A. B.-J. Teoh, “Stacking-based Deep Neural Network: Deep Analytic Network on Convolutional Spectral Histogram Features,” in *2017 IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 1592–1596.
- [515] C. Zhao, X. Wan, G. Zhao, and Y. Yan, “Spectral--spatial classification of hyperspectral images using trilateral filter and stacked sparse autoencoder,” *J. Appl. Remote Sens.*, vol. 11, no. 1, p. 16033, 2017.
- [516] Y. Cao, H. Wang, M. Moradi, P. Prasanna, and T. F. Syeda-Mahmood, “Fracture detection in x-ray images through stacked random forests feature fusion,” in *Biomedical Imaging (ISBI), 2015 IEEE 12th International Symposium on*, 2015, pp. 801–805.
- [517] D. R. Pereira, R. J. Pisani, A. N. de Souza, and J. P. Papa, “An Ensemble-Based Stacked Sequential Learning Algorithm for Remote Sensing Imagery Classification,” *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, 2017.
- [518] Z.-H. Zhou, *Ensemble methods: foundations and algorithms*. CRC press, 2012.
- [519] R. E. Schapire, “The strength of weak learnability,” *Mach. Learn.*, vol. 5, no. 2, pp. 197–227, 1990.
- [520] L. Breiman, “Bagging predictors,” *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.
- [521] B. Efron and R. J. Tibshirani, *An introduction to the bootstrap*. CRC press, 1994.
- [522] T. G. Dietterich, “Ensemble methods in machine learning,” in *International workshop on*

- multiple classifier systems*, 2000, pp. 1–15.
- [523] L. Breiman, “Stacked regressions,” *Mach. Learn.*, vol. 24, no. 1, pp. 49–64, 1996.
- [524] P. Smyth and D. Wolpert, “Stacked density estimation,” in *NIPS*, 1997, pp. 668–674.
- [525] Y. Freund and R. E. Schapire, “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting,” *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997, doi: 10.1006/jcss.1997.1504.
- [526] T. Ojala, M. Pietikainen, and D. Harwood, “Performance evaluation of texture measures with classification based on Kullback discrimination of distributions,” in *Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on*, 1994, vol. 1, pp. 582–585.
- [527] Y. Xu, H. Ji, and C. Fermuller, “A projective invariant for textures,” in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*, 2006, vol. 2, pp. 1932–1939.
- [528] A. Oliva and A. Torralba, “Modeling the shape of the scene: A holistic representation of the spatial envelope,” *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001.
- [529] A. Oliva and A. Torralba, “Building the gist of a scene: The role of global image features in recognition,” *Prog. Brain Res.*, vol. 155, pp. 23–36, 2006.
- [530] C. H. Chen and G.-K. Young, “A study of texture classification using spectral features,” 1982.
- [531] C. Cortes and V. Vapnik, “Support-vector networks,” *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.
- [532] J. Demšar, “Statistical comparisons of classifiers over multiple data sets,” *J. Mach. Learn. Res.*, vol. 7, no. Jan, pp. 1–30, 2006.