Sparse representation and its applications to airborne platforms

Srinivasan, Abirami

2021

Srinivasan, A. (2021). Sparse representation and its applications to airborne platforms. Doctoral thesis, Nanyang Technological University, Singapore.
<https://hdl.handle.net/10356/148844>

<https://hdl.handle.net/10356/148844>

<https://doi.org/10.32657/10356/148844>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 09:11:08 SGT

Sparse Representation and its Applications to Airborne Platforms

Abirami Srinivasan

School of Electrical and Electronic Engineering

A thesis submitted to Nanyang Technological University in partial
fulfillment 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.

12 Apr 2021

Date

Abirami Srinivasan

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of 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 accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

12 Apr 2021

Date

A handwritten signature in black ink, appearing to read 'Anamitra Makur', written in a cursive style.

Anamitra Makur

Authorship Attribution Statement

This thesis contains material from 1 paper accepted at conferences in which I am listed as an author.

Some of the contents in Chapter 5 are published in A. Srinivasan, A. Vijayakumar and A. Makur, Compressive Sensing-based On-board Intrusion Detection for a Surveillance Drone, IEEE VTS Asia Pacific Wireless Communications Symposium, APWCS, Aug 2019.

The contributions of the co-authors are as follows:

- Prof A. Makur is my supervisor and he has guided me and reviewed my manuscript.
- Dr. A. Vijayakumar is my co-supervisor at Airbus and she has guided me and reviewed the manuscript.

12 Apr 2021

Date

Abirami Srinivasan

Acknowledgements

This thesis would not be what it is without the guidance, keen inputs and support of my supervisor, Prof Anamitra Makur. He has been ever willing to offer his help with suggestions and will always be someone I look up to for his calm demeanour and patience. I am eternally indebted to my co-supervisor, Dr. Asha Vijayakumar for being extremely supportive, kind and being there always to lift me up when my spirits fell. Conversations with her spanning the entire spectrum from technical discussions to light-hearted chats will always be my cherished memories. I have been fortunate to have been guided by the most patient and encouraging supervisors.

I am extremely grateful to the Economic Development Board (EDB), Singapore for sponsoring this Industrial PhD Programme (IPP) and enabling me to work towards this. I am thankful to Airbus for having given me the opportunity to undertake this Industrial PhD Programme (IPP) with NTU. I am thankful to Cyrille, Francois and Julian at Airbus for their support and push to see this thesis through. Special thanks to Roger for having been the kind person that he is and being understanding of the time-sharing I needed to do for the research and the project work. Thanks to Supriti and Archana for the happy times at Airbus. I am thankful to Mukund, Sathiya and Sujit for the most interesting and helpful discussions on Compressed Sensing and more.

My eternal pillars of support and strength, Porna, Puppy, Tree and Gee, thank you for always being there for me! Much love and thanks to my other family at Smule, Vimmo and

Nisha for being the stress-buster the last 3 years with their music and love! My dearest family, Appa, Amma, Sivaram, Sarada and the newest entrant, Ahaana - there is never a dull moment with all of you, thanks for all the love and the faith in me! No amount of thanks can suffice for Ravi and my darling daughter, Kavya for giving me the space and time to deal with the stress and the pressure to complete this. Special love to Kavya for having tolerated me these last few years while I was battling with work. Sometimes, all I needed was her smile at the end of the day to cheer me up!

Contents

Acknowledgement	i
List of Figures	ix
List of Tables	x
Summary	xi
1 Introduction	1
1.1 Sparsity	2
1.2 Applications	3
1.3 Contributions	3
1.4 Outline of the Thesis	5
2 Literature Review	6
2.1 Dictionary	6
2.2 Dictionary Learning Algorithms	8
2.2.1 K-SVD	8
2.2.2 Sequential Generalization of K-Means, SGK	9
2.2.3 Discriminative Dictionary Learning Algorithms	10
2.3 Compressed Sensing (CS)	11
2.3.1 Signal Recovery	11

2.3.2	Sensing Matrices	12
2.3.3	Reconstruction Algorithms	15
2.4	Summary and Motivation of Proposed Works	18
3	Sparse Representation-based Anomaly Detection	19
3.1	Anomaly Detection	19
3.1.1	Anomaly Detection Techniques	20
3.1.2	One-Class Classification	21
3.1.3	Sparse Representation-based Methods	21
3.2	New Data-driven Approaches to Training for Anomaly Detection	22
3.2.1	l_∞ Formulation	23
3.2.2	Baseline Retraining	25
3.2.3	Weighted SGK	26
3.3	Dataset 1: Numenta Anomaly Benchmark (NAB)	27
3.3.1	Training and Anomalous Data	27
3.3.2	Dictionary Training and Retraining	28
3.3.3	Anomaly Detection: Choice of Metrics and Thresholds	29
3.3.4	Performance Comparison of Baseline Retraining with Weighted SGK Re- training	31
3.4	Dataset 2: AIRBUS Dataset	34
3.4.1	Training data	34
3.4.2	Validation Data	34
3.5	Data Analysis	35
3.5.1	Preliminary Analysis	35
3.5.2	Sparse Representation of Data set with a Trained Dictionary	36
3.6	Performance of Dictionary Training Algorithms	37
3.6.1	Baseline Retraining	37
3.6.2	Weighted SGK Retraining	38

3.6.3	Comparison of Baseline Retraining with Weighted SGK Retraining	40
3.7	Performance of Anomaly Detection	41
3.7.1	Anomaly Detection Algorithm	41
3.7.2	Results and Discussion	43
3.7.3	Comparison with State-of-the-art Anomaly Detection Algorithms	48
3.8	Summary	50
4	Sparse Representation-based Processing in Synthetic Aperture Radar	51
4.1	Synthetic Aperture Radar	51
4.1.1	Raw SAR Data and its Processing	53
4.1.2	Advantages of Sparse Representation in SAR Processing	55
4.2	Applications of Sparse Representation in SAR: A Background	56
4.3	Exploring Phase in SAR with Point Targets	59
4.3.1	Scenario: One Point Target	60
4.3.2	Scenario: Two point targets	60
4.4	Compressed Sensing-based Low Rank and Sparse Decomposition (LRSD) of SAR	
data		63
4.4.1	Simulations: Synthetic Experiment	64
4.4.2	Proposed Method of CS-based LRSD	65
4.4.3	Proposed Algorithm	68
4.4.4	Simulations and Results: Real Data	69
4.4.5	CS-based LRSD for Speckle Noise Processing	74
4.5	Summary	76
5	Compressive Sensing-based On-board Intrusion Detection for a Surveillance	
	Drone	78
5.1	Background	79
5.1.1	Leveraging CS for On-board Processing	80

5.2	Implementation	81
5.2.1	System Design	81
5.2.2	The Detection Scheme	82
5.2.3	Implementation on Testbed	85
5.3	Results and Discussion	85
5.3.1	Detection Accuracy	86
5.3.2	Encoding Time	87
5.4	Compressed-sensed Domain Processing from the viewpoint of Edge Computing Analytics	87
5.5	CS-based Reconstruction	89
5.5.1	Proposed Alternate Implementations for OMP	89
5.5.2	Analysis of Mean Squared Error (MSE) across proposed methods	91
5.5.3	Simulations and Results	94
5.5.4	Summary	95
6	Conclusions and Future Directions	98
6.1	Conclusions	98
6.2	Future Directions	100
	Author's Publications	101
	References	102

List of Figures

2.1	Illustration to show the relaxation of l_0 -norm to l_1 -norm	12
3.1	Anomalous Sequence	28
3.2	Anomalous Sequence	29
3.3	Anomaly Detection Metric: Average Representation Error for Baseline Retraining	30
3.4	Anomaly Detection Metric: Maximum Sparsity for Baseline Retraining	31
3.5	Anomaly Detection Metric: Average Representation Error for Weighted SGK Retraining	32
3.6	Anomaly Detection Metric: Maximum Sparsity for Weighted SGK Retraining	33
3.7	Signal: Normal and Anomaly	35
3.8	Statistical Features of Training Data: Mean and Standard Deviation	36
3.9	Frequency Analysis of Training Data	37
3.10	Basic Statistical Features of Training Data	38
3.11	Basic Statistical Features of Training Data	39
3.12	Error Plot for various Weights Matrix used	40
3.13	Representation Error in Sparse Coding	41
3.14	Distribution of Sparsity of Training sequences in Baseline Retraining	42
3.15	Distribution of Sparsity of Training sequences in Weighted SGK Retraining	43
3.16	Distribution of Sparsity of Validation sequences in Baseline Retraining	44
3.17	Distribution of Sparsity of Validation sequences in Weighted SGK Retraining	45

3.18 Performance of <i>TestA</i>	46
3.19 Performance of <i>TestB</i>	47
3.20 Performance with 3 Normal sequences in the training set	47
4.1 Geometry of SAR platform	52
4.2 Raw SAR signal	55
4.3 Range Compressed data	55
4.4 Azimuth Compressed data	56
4.5 Variation of Phase across Range bins at one location along the Azimuth	61
4.6 Phase of the received signal from Two Point Targets, $n = 2$, $\delta x = 30m$	62
4.7 Phase of the received signal from Two Point Targets, $n = 2$, $\delta x = 60m$	62
4.8 Isolation of the Jitter contributed by the 2^{nd} target	63
4.9 Noisy Synthetic raw SAR data	65
4.10 Range-Compressed noisy SAR data	66
4.11 Azimuth-Compressed noisy SAR data	66
4.12 Range/Azimuth compressed sparse Component after LRSD and SAR processing: Target1(1320, 210), Target2(1480, 270)	67
4.13 Compression in SAR with Proposed Method	68
4.14 Original SAR Image reconstructed from the raw data (with SAR processing)	70
4.15 Low Rank Component reconstructed after SAR processing:: $\lambda_{max} = 0.95$, $MR =$ 0.95	71
4.16 Sparse component reconstructed after SAR processing: $\lambda_{max} = 0.95$, $MR = 0.95$	72
4.17 Low Rank Component reconstructed after SAR processing:: $\lambda_{max} = 1.5$, $MR =$ 0.95	73
4.18 SAR Image reconstructed from the CS-encoded SAR data ($MR = 0.95$	74
4.19 Speckle-suppressed Low Rank component reconstructed after SAR processing: $\lambda_{max} = 0.95$, $MR = 0.95$	76

4.20	Speckled Sparse component reconstructed after SAR processing: $\lambda_{max} = 0.95$, $MR = 0.95$	77
5.1	Illustration of the Test Scenario	82
5.2	Schematic Flow of the Intrusion Detection	83
5.3	Encoding Time	88
5.4	Comparison of errors for various measurements in a 32-sparse Signal with $N =$ 256 averaged over 100 trials	93
5.5	Comparison of errors across SNR for $N = 256$, $M = 100$, $m = 32$ averaged over 100 trials	93
5.6	Error Convergence for Various Measurements in a m -sparse Signal with $N = 256$ averaged over 100 trials	95
5.7	Error Convergence for Various Measurements in a m -sparse Signal with $N = 256$ averaged over 100 trials	96
5.8	Computational Times for Various Measurements in a m -sparse Signal with $N =$ 256 averaged over 100 trials	97

List of Tables

3.1	Detection Accuracy	32
3.2	<i>TestA</i> : Detection Accuracy	45
3.3	<i>TestB</i> : Detection Accuracy	46
3.4	Comparison of Performance with State-of-the-art Algorithms	50
4.1	Summary of state-of-the-art in sparse representation-based SAR	59
4.2	Synthetic SAR signal specifications	64
4.3	Image Quality Comparison	73
5.1	Detection Accuracy	86
5.2	Detection Accuracy for Varying Measurement Ratios	87
5.3	Alternate proposed approaches for OMP implementation; $A^\dagger$ is the pseudoinverse of matrix A	90
5.4	Comparison of Computational Times	94

Summary

Many natural signals can be concisely represented in a different domain. These signals, are then said to be sparsely represented. Approximation of a signal with an overcomplete system gives a sparse representation of the signal based upon redundant “dictionaries”. The replacement of the original signal with a sparse representation does not only result in data compression, but also provides an alternative look into the inherent structure of the natural signals with the overcomplete dictionaries. Sparse representation finds application in various fields such as signal processing, image processing, pattern recognition, etc. Numerous algorithms are increasingly being proposed for sparse representation. The advantages of sparse representation can be exploited well for processing associated with signals generated on airborne platforms. The work discussed in this thesis explores the use of sparse representation in signal processing related to airborne platforms such as drones, helicopters and Synthetic Aperture Radar (SAR). Compressed Sensing is a direct relative of sparse representation, and has grown increasingly popular in the recent years. Compressed Sensing (CS) has been emerging as an effective sampling scheme for signal acquisition and efficient sensor design which not only achieves data compression but also enables various forms of signal detection and estimation with vastly reduced data. CS relies on the inherent “sparsity”, or compressibility of a signal in some basis and operates on far fewer samples than traditional sampling schemes. The signals, thus sensed, can be reconstructed accurately, with high probability, using convex optimization techniques such as l_1 -optimization or more efficient greedy recovery algorithms such as Orthogonal Matching Pursuit. This new sampling scheme finds applications in various areas of signal processing.

The application of sparse representation-based dictionary in anomaly detection of accelerometer data obtained from helicopters is explored in this thesis. Sparse-representation based anomaly detection uses the sparsity of the signals to train a dictionary. A Sequential Generalization of K -means (SGK), a dictionary training algorithm is used towards this end. The training data which are classified “normal”, typically contains a wide set of signals of which some show significant deviation from the more common statistical features of the rest. In such a scenario, a dictionary is required to be retrained to be able to represent the signals reasonably across the sample space. A modification to the SGK, the “Weighted Sequential Generalization of K-Means” is proposed here to be able to represent the signals more sparsely. The performance of the Weighted SGK and the SGK in retraining are compared with the validation set of the data. It is observed that Weighted SGK outperforms SGK in the detection performance for the anomalies in the helicopter data. Weighted SGK is also seen to outperform some classical machine learning approaches.

Synthetic Aperture Radar (SAR), that is used in remote sensing for high resolution imagery by utilizing flight path to synthesize a large aperture, generates large amount of data that are transmitted to the ground station where it is further processed. CS exhibits the potential to reduce this data that needs to be transmitted through the downlink. CS-based Low Rank and Sparse Decomposition (LRSD) is proposed as a solution to tackle the data rate on the downlink. Proposed LRSD has two-fold advantage of compression of raw data as well as useful image information isolated in one of the two components, either the low-rank or the sparse. The quality of the images are compared among those obtained by the uncompressed data, CS-compressed data and CS-compressed-LRSD-decomposed data. It is shown that the metrics used to indicate certain visual aspects of the images gave reasonable results for the CS-based LRSD scheme, as compared to the uncompressed and the traditional CS-compressed data. The use of inherent nature of the phase in SAR signals is a rather unexplored area and is of potential interest.

Surveillance systems are widely used not only for military but also for everyday civilian pur-

poses. Intrusion-detection schemes are of interest in every surveillance system. Mobile flying platforms such as drones are being used effectively as an autonomous surveillance system that can not only detect an intrusion on-board, but also make an informed decision to either track the intruder or transmit the required information to the ground station. Efficient techniques to implement a surveillance system on-board continue to be explored. A CS-based scheme for intrusion detection on-board the AR Parrot Drone is implemented. It is observed that a CS-based system can make efficient use of on-board resources by exploiting the properties of the compressed data space. It is also capable of efficient usage of the available bandwidth by transmitting frames to the ground station only if an intrusion is detected. Performance metrics, namely, encoding time and detection accuracy were evaluated. The detection accuracy across data sets that were tested was found to be excellent. The encoding time using CS on-board outperformed JPEG encoding time for similar region-of-interests (ROIs).

Chapter 1

Introduction

Over the past decade, there has been a paradigm shift from “not-really-sufficient” amount of data produced by sensors (or “the sensor bottleneck”) to “too-overwhelming-to-store-and-process” (or “the data deluge”) amount of data. Circumventing the data deluge situation require quick-paced advancements in the design of sensors, compression, storage devices, communication systems and signal processing algorithms. Often, signals of interest contain information that is much lesser than the amount of data generated. In other words, the information content in signals reside in a space that is of lower dimension and a subset of the higher dimension space of the raw data [1]. Signals with low-dimensional structure may be modeled in various ways. When the raw data samples can be transformed to a domain where a much lesser number of coefficients can define the signal, the model is termed as “sparse signal” model and forms the area of interest in this thesis.

Sparse models form the crux of compression and processing algorithms such as JPEG. With growing interest in the field of representation domain, various transforms were developed to visualize the signals more compactly. Transforms such as Fourier Transform, Discrete Cosine Transform, Wavelets, to name a few, when applied to appropriate signals, can give rise to concise representations. Such concise or sparse representation can be constructed by the decomposition of signals over a few basis waveforms comprising the “dictionary” [2].

1.1 Sparsity

The key concept underlying sparse representation is that the signal is sparse (or having very few nonzero values). It is assumed that any signal $\mathbf{x} \in \mathbb{R}^n$ can be represented using an overcomplete dictionary $D \in \mathbb{R}^{n \times K}$, which contains K atoms, or representative signals or as it is termed in mathematics, basis vectors $\{\mathbf{d}_j\}_{j=1}^K$. The signal $\mathbf{x}$ can be written as a linear combination of these basis vectors as $\mathbf{x} = D\mathbf{s}$, subject to $\|\mathbf{s}\|_0 \ll n$. $\mathbf{s} \in \mathbb{R}^K$ contains the sparse coefficients of the signal $\mathbf{x}$. Solutions of interest are those that satisfy

$$\arg \min_{\mathbf{s}} \|\mathbf{s}\|_0 \text{ such that } \|\mathbf{x} - D\mathbf{s}\|_0 \leq \epsilon, \quad (1.1)$$

where ϵ is the upper bound on the error. These problems are combinatorial in nature and difficult to solve. Hence, convex relaxations are used to solve the above. Real world signals such as sound and image can be stored in compressed form, in terms of their projection on suitable basis. With an appropriately chosen basis, a large number of representation coefficients are zero or negligible and the signal is said to be compressible.

When there exist limitations in the size of data that can be generated in data capturing devices, or when measurements are expensive or time-consuming, there is a need for a technique which can “measure only what should be measured”. Compressive (or Compressed) Sensing (CS) has seen an increased interest in the last decade mainly due to the rising demand in fast, efficient and inexpensive signal processing algorithms for applications and devices. Even though the traditional Shannon-Nyquist paradigm [3] calls for a sampling frequency that is twice the magnitude of the maximum frequency in the signal, the CS paradigm, relying on finding sparse solutions to underdetermined linear systems, can reconstruct signals from far fewer samples than is possible using the Nyquist sampling rate. CS operates under the assumption that signal acquisition and compression can be carried out simultaneously and that a sparse signal can be sampled at its “information rate” using non-adaptive linear measurements [4] [5].

1.2 Applications

Sparse representation methods have been utilized in various applications of computer vision, image processing, machine learning, to mention a few. The techniques have proved very useful in the often explored applications such as image denoising, classification, object detection/-tracking and signal processing. Dictionary learning is one of the most classically used sparse representation methods. Dictionary learning methods exploiting sparse representation are capable of performing better than the dictionaries based on pre-defined transform functions. A sparse representation-based dictionary can be effectively learned from the data and hence will be the best transform domain for the data.

A contributing factor to the data deluge scenario is the sudden increase in the airborne platforms over the past few years. Airborne sensors, that is, sensors on airborne vehicles/platforms such as aircraft and helicopters, generate plenty of data. These are stored and once they are on the ground, required to make decisions on health of the systems and smooth functioning. The number of Unmanned Aerial Vehicles (UAVs/drones) in the atmosphere have gone up recently. The data captured by them and transmitted back to the ground station for the purpose of imaging, detection, tracking and monitoring is innumerable. There is an increasing use of drones in personal, commercial as well as military applications. Synthetic Aperture Radar (SAR) is an airborne imaging platform which achieves high resolution imaging of a 2D/3D scene by the utilization of a synthetic aperture created by the airborne platform [6]. Most SAR systems are limited in size, weight, power and cost. These systems generate a constantly increasing volume of data that require to be transmitted to a ground station for processing. Thus, sparse representation and compressed sensing shows promise and potential use in these areas.

1.3 Contributions

In order to bridge the gap between the large amount of data generated on board airborne platforms and the need to store/transmit just the required information, the objective of this

research is to explore the sparse representation-based techniques. The proposed techniques are discussed for their feasibility in real world data scenario. Real world data incorporate challenges in terms of unpredictable behaviour of noise and lack of a predetermined structure. Data-dependent dictionaries and sparse representation-based analysis and detection techniques prove very useful in such situations. The specific contributions of the thesis are listed below:

- The thesis contributes alternative approaches to traditional dictionary training that can effectively train a dictionary which can represent data more sparsely. Among the approaches proposed, is the Weighted Sequential Generalization of K -means (Weighted SGK) which is a modification to SGK [7]. The Weighted SGK, similar to the SGK is a sequential update algorithm that modifies the dictionary already obtained using SGK by preferential weighting to those signals that are the “training outliers” in the normally behaved training set. It is shown that the proposed algorithm performs better in the application of anomaly detection, as compared to the baseline training algorithm. An alternate mathematical formulation with the use of l_∞ -norm is also proposed for dictionary learning that would give a better dictionary representation for the outliers.
- The thesis also contributes a CS-based scheme for low rank and sparse decomposition for SAR data. This achieves compression of raw data as well as useful image information isolated in one of the two components, either the low-rank or the sparse. The quality of the images are compared among those obtained by the uncompressed data, CS-compressed data and CS-compressed-LRSD-decomposed data. The nature of the phase in the SAR signals is also analyzed which reveals the potential of sparse sensing in the phase component of SAR to build the target scene entirely for point targets.
- The thesis demonstrates the use of the compressed sensed domain (CS domain) and its implementation onboard a surveillance drone for intrusion detection. An intrusion detection scheme is proposed which exploits the compressive measurement domain without relying on full scale signal reconstruction. The proposed scheme is shown to extract the

required information (in this case, the event detection of intrusion) from far lesser samples than would be required for a complete CS-based reconstruction-driven decision. CS encoding rate for various region-of-interests (ROIs) is also shown to be superior to a traditional JPEG encoding rate. A more efficient implementation of the Orthogonal Matching Pursuit (OMP) is also proposed which shows significant savings in computational time.

1.4 Outline of the Thesis

The thesis comprises six chapters. Chapter 1 introduces the basic concept of sparsity, its applications and the sparse representation-based works presented in the thesis. Chapter 2 reviews the fundamentals and relevant existing works in sparse-representation and CS. Chapter 3 proposes multiple approaches for sparse representation-based dictionary training and their application to anomaly detection. It proposes an alternative strategy, the Weighted Sequential Generalization of K -means (Weighted SGK) for improved performance in anomaly detection. Chapter 4 introduces the fundamentals in SAR processing and sparse representation-related works in SAR. It proposes a CS-based low rank and sparse decomposition for SAR data and discusses its impact on the respective image components. Chapter 5 discusses the implementation of CS-based intrusion detection on-board drones and demonstrates the advantages of computational efficiency in the CS domain. Chapter 6 concludes the thesis with directions to potential future work.

Chapter 2

Literature Review

Compression is often used to resolve the challenges associated with storage and computation costs involved in dealing with real-world high-dimensional data. This aims at finding the most concise representation of a signal that is able to deal with acceptable distortion margins. Transform Coding is one of the most popular techniques for signal compression which works on finding a basis that provides sparse or compressible representations for signals in a class of interest. A sparse signal is one which is of length n , but can be represented with $m \ll n$ non-zero coefficients. A compressible representation can very well approximate the signal with only m non-zero coefficients. Both sparse and compressible signals can be represented with high fidelity. Sparse approximation preserves only the values and locations of the largest coefficients of the signal to represent the sparse and compressible signals.

2.1 Dictionary

Signals can often be well-approximated as a linear combination of just a few elements from a known basis or “dictionary”. The signal is said to be sparse when this representation is exact. Sparse signal models provide a mathematical scheme to represent high-dimensional signals that contain relatively little information in their original domain. Given a dictionary $D \in \mathbb{R}^{n \times K}$ containing basis atoms $\mathbf{d}_k \in \mathbb{R}^n$ for $k = 1, \dots, K$, the goal of sparse representation

is to represent a signal $\mathbf{x} \in \mathbb{R}^n$ as a linear combination of a small number of atoms $\mathbf{x} = D\mathbf{s}$, where $\mathbf{s} \in \mathbb{R}^K$ is the sparse representation vector and $\|\mathbf{s}\|_0 = m : m \ll n$. Dictionaries that better fit such a sparsity model, can either be chosen from a defined set of linear transforms such as Fourier, Cosine and Wavelet or can be trained on a set of training signals [2]. Given a set of training signals, a trained D will always produce a better sparse representation in comparison to traditional parametric bases [7]. This is because, for a set of training signals $X = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]$, D is trained to minimize the representation error,

$$D, S = \arg \min_{D, S} \|E\|_F^2 = \arg \min_{D, S} \|X - DS\|_F^2 \quad (2.1)$$

with a constraint that $S = [\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_N]$ are the sparse representations of $\mathbf{x}_i$. $\|E\|_F$ is the Frobenius-norm of the matrix. As the error minimization depends both on S and D , the solution is obtained iteratively by alternating between sparse coding (for X) and dictionary update (for D). Since the solution of l_0 -norm regularization is usually a NP-hard problem, utilizing a convex relaxation strategy to replace l_0 -norm regularization is an often-utilized method for sparse coding stage in dictionary learning. Further information on algorithms for sparse coding is discussed in Section 2.3.3. Dictionary learning methods, generally, iteratively update the sparse approximation and the dictionary alternatively. Dictionary learning algorithms can be divided into categories based on presence of structures, such as parallel and serial. These can also be divided into categories based on different norms used in the penalty term, i.e., l_0 -norm relaxation based methods [8]. Further, based on the learning mode, all dictionary learning algorithms can be divided into (i) Supervised Dictionary Learning and (ii) Unsupervised Dictionary Learning. The differentiator between the two classes, Supervised and Unsupervised, depends on whether the class label is used in the dictionary learning. Unsupervised dictionary learning methods have been implemented for multiple applications such as image compression, image denoising, feature coding of image representation, to mention a few [8]. Supervised Dictionary Learning methods use sparse representation for signals belonging to various classes with respect to a

shared dictionary and multiple decision functions as described in [9].

2.2 Dictionary Learning Algorithms

One of the earliest works on dictionary learning algorithms proposed for sparse representation is found in [10]. A few algorithms used for dictionary learning, such as K-SVD [11], SGK [7] and discriminative K-SVD [12] are discussed below.

2.2.1 K-SVD

K-SVD [11] is one of the most well known unsupervised dictionary learning algorithm which is an extension of the method of directions (MOD) algorithm [13]. It is based on the l_0 -norm penalty. The objective function of K-SVD is:

$$\arg \min_{D, S} \|X - DS\|_F^2 \text{ s.t. } \|\mathbf{s}_i\|_0 \leq m, i = 1, 2, \dots, N \quad (2.2)$$

where $X \in \mathbb{R}^{n \times N}$ is the matrix that consists of all training signals, $D \in \mathbb{R}^{n \times K}$ is the learned dictionary, $S \in \mathbb{R}^{K \times N}$ is the matrix of sparse coefficients. This joint optimization problem is solved by alternatively optimizing D and S iteratively. With fixed S , D is obtained by solving:

$$\hat{D} = \arg \min_D \|X - DS\|_F^2 \quad (2.3)$$

where $\hat{D} = XS^\dagger = XS^T(SS^T)^{-1}$. When D is directly estimated in this manner, this is called the Method of Optimal Directions (MOD). K-SVD sequentially updates the atoms in the dictionary, instead of the entire matrix being updated. K-SVD is shown to be better than MOD in terms

of accuracy and speed in [11]. Equation 2.3 is expressed in K-SVD as:

$$\begin{aligned}
\hat{D}, \hat{X} &= \arg \min_{D, X} \|X - DS\|_F^2 \\
&= \arg \min_{D, X} \left\| X - \sum_{j=1}^{j=K} \mathbf{d}_j S_j^T \right\|_F^2 \\
&= \arg \min_{D, X} \left\| (X - \sum_{j \neq l} \mathbf{d}_j S_j^T) - \mathbf{d}_l S_l^T \right\|_F^2
\end{aligned} \tag{2.4}$$

Here S_j is the j^{th} row of S . Singular Value Decomposition (SVD) is used to find the closest rank-1 matrix that approximates the error at iteration, t , $E_k^{(t)} = X - \sum_{j \neq k} \mathbf{d}_j^{(t)} S_j^{(t)}$ subject to $\|\mathbf{d}_k^{(t+1)}\| = 1$. SVD operates on $E_k^{(t)} = U \Delta V^T$. The first column of U is used to represent $\mathbf{d}_k^{(t+1)}$, and $\hat{S}_k^{(t)}$ is given by the first column of V multiplied by the first diagonal entry of Δ . Both $\mathbf{d}_k$ and S_k are updated in the dictionary update stage. K-SVD is widely used in image processing for compression, denoising and filling in missing pixels.

2.2.2 Sequential Generalization of K-Means, SGK

SGK is another example of an unsupervised dictionary learning algorithm. While MOD is suitable for a range of sparse representation applications irrespective of conditions imposed on sparse structure and dictionary, it may demand more computational resources [7]. Algorithms like K-SVD that process sequentially would need lesser resources. SGK sequentially updates the dictionary atoms by fixing the updated sparse coefficients at the dictionary update stage. Thus, the solution to the sequential update problem as in Equation 2.4 is reformulated as:

$$\mathbf{d}_{k+1}^{(t+1)} = \arg \min_{\mathbf{d}_k} \left\| E_k^{(t)} - \mathbf{d}_k S_k^{(t)} \right\|_F^2 \tag{2.5}$$

where the solution is obtained similar to MOD as:

$$\mathbf{d}_{k+1}^{(t+1)} = E_k^{(t)} S_k^{(t)T} (S_k^{(t)} S_k^{(t)T})^{-1} \tag{2.6}$$

This process is repeated for all K atoms sequentially. The complexity of SGK is shown to be much less than K-SVD by an order in [7]. SGK finds application in image compression, denoising, and image inpainting [14].

2.2.3 Discriminative Dictionary Learning Algorithms

Discriminative Algorithms are examples of supervised dictionary learning algorithms. These set of algorithms utilize a joint representation and classification scheme that achieves the purpose of learning the most discriminative sparse overcomplete encoding and optimal classifiers. This is achieved by combining the classification and representation error into the objective function [15].

$$\hat{D}, \hat{W}, \hat{\mathbf{s}} = \arg \min_{D, W, \mathbf{s}} \|X - D\mathbf{s}\|_2 + \gamma \|H - W\mathbf{s}\|_2 + \beta \|W\|_2 \text{ subject to } \|\mathbf{s}\|_0 \leq m \quad (2.7)$$

where $\hat{D}, \hat{W}, \hat{\mathbf{s}}$ are the estimates for the unknowns, X is the set of input signals, D is the dictionary, $\mathbf{s}$ is the sparse coefficients, H is the label of the training set, W the parameter of the classifier, and γ and β are scalars that lend relative weights to the terms. The solution for Equation 2.7 is obtained by iteratively solving for D , W and $\mathbf{s}$.

Discriminative K-SVD (D-KSVD), an example of the above class, works on the extension of K-SVD algorithm by simultaneously enabling the performance of a linear classifier and the representational power of the dictionary by the same optimization technique. The D-KSVD algorithm finds the dictionary and solves for the classifier using a derivative of the K-SVD algorithm. [12] discusses the use of D-KSVD in face recognition. Label Consistent K-SVD (LC-KSVD) is proposed in [16] to learn a discriminative dictionary for image classification. LC-KSVD utilizes the supervised information to learn the dictionary, unifies the dictionary and optimal linear classifier construction simultaneously, and obtains the learned dictionary and the optimal classifier.

Apart from the methods mentioned so far, dictionary learning can also be used for semi-

supervised instances where all training data are not labeled or only a single class of data is available, such as in the applications of anomaly detection [17].

2.3 Compressed Sensing (CS)

Sparsity of a signal can also be exploited in its compressed acquisition. The pivotal idea underlying CS [18] is to recover a high-dimensional sparse signal from a dimension-reduced space. Suppose $\mathbf{x} = \{x_i\}_{i=1}^N$ to be the signal that is to be recovered. A priori known, this signal is either, (i) inherently sparse which means that $\|\mathbf{x}\|_0 \leq m \ll N$ where $\|\mathbf{x}\|_0$ represents the l_0 -norm which gives the number of non-zero coefficients in $\mathbf{x}$, or (ii) is sparse in a basis such that $\mathbf{x} = \psi\boldsymbol{\theta}$ where $\|\boldsymbol{\theta}\|_0 \leq m$, ψ is a matrix composed of elements of the orthonormal basis, and m is termed the *sparsity* of the signal. While this is the theoretical definition of sparsity, very few real-world signals are exactly sparse. So, the real-world signals are termed as *approximately* sparse, which means the signal can actually be approximated by a few significant coefficients.

The CS problem is mathematically expressed as:

Let A be a $M \times N$ matrix where $M \leq N$, called the *measurement* or the *sensing* matrix. Then $\mathbf{y} = A\mathbf{x}$ are the measurements. Given $\mathbf{y}$, the recovery problem is as follows: Recover $\mathbf{x}$ such that $\mathbf{y} = A\mathbf{x}$ or recover $\boldsymbol{\theta}$ such that $\mathbf{y} = A\psi\boldsymbol{\theta}$. This leads to an underdetermined system of linear equations. For simplicity of discussions, the subsequent sections will address inherently sparse signals. The explanation for compressible signals in a proper basis is merely an extension to the following discussion.

2.3.1 Signal Recovery

The signal $\mathbf{x}$ is to be recovered by:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ subject to } \mathbf{y} = A\mathbf{x} \quad (2.8)$$

Equation 2.8 can be solved only by combinatorial search and hence is NP-hard. To circumvent this problem, Donoho *et al* proposed a solution by the relaxation of the l_0 -norm to the l_1 -norm which can then be solved by convex optimization [19]. Figure 2.1 explains how the shape of the l_1 -ball is a better promoter of the sparsity than the l_2 -ball.

Figure 2.1: Illustration to show the relaxation of l_0 -norm to l_1 -norm

The conditions that rule the situation where ‘ $\|x\|_0 = \|x\|_1$ ’ holds the key to CS. The discussion in the following sections explain the conditions that are necessary to ensure accurate signal recovery.

2.3.2 Sensing Matrices

The main design requirement for the sensing matrix, A is to facilitate the unique identification of the signal, $\mathbf{x}$ from its measurements $\mathbf{y} = A\mathbf{x}$. Suppose the sparsity of the signal is m , the number of measurements we require is $M > K$. To help in our discussion of the properties of a sensing matrix, we define relevant terms and state the following theorems.

Definitions

- The spark, $\text{spark}(A)$ of a given matrix is the smallest number of columns of A that are linearly independent.

- The coherence $\mu(A)$ is the largest absolute inner product of the columns of A taken pairwise:

$$\mu(A) = \max_{1 \leq i \neq j \leq N} \frac{|\langle A_i, A_j \rangle|}{\|A_i\|_2 \|A_j\|_2} \quad (2.9)$$

$$\mu(A) \in \left\{ \sqrt{\frac{N-M}{M(N-1)}}, 1 \right\}. \text{ When } N \gg M, \mu(A) \geq 1/\sqrt{M}$$

- A matrix A has the (m, δ_m) -restricted isometry property $\{(m, \delta_m)\text{-RIP}\}$, if, $\forall \mathbf{x} \in \Sigma_m$ (where Σ_m is the space of m -sparse signals),

$$(1 - \delta_m) \|\mathbf{x}\|_2^2 \leq \|A\mathbf{x}\|_2^2 \leq (1 + \delta_m) \|\mathbf{x}\|_2^2 \quad (2.10)$$

with δ_m called the isometry constant. The RIP ensures that all submatrices of A of size $M \times m$ are distance-preserving [20].

Theorem 1. If $\text{spark}(A) > 2m$, then for each measurement vector $\mathbf{y} \in \mathbb{R}^M$, there exists at most one signal $\mathbf{x} \in \Sigma_m$ such that $\mathbf{y} = A\mathbf{x}$.

The $\text{spark}(A) \in [2, M + 1]$ so that Theorem 1 gives $M \geq 2m$. Even though this theorem ensures unique m -sparse signals, the calculation of the spark of matrix is combinatorial and hence, it is preferred to use coherence of the matrix that can be easily calculated [21].

Theorem 2. If $m < \frac{1}{2} \left\{ 1 + \frac{1}{\mu(A)} \right\}$ then for each measurement vector $\mathbf{y} \in \mathbb{R}^M$, there exists at most one signal $\mathbf{x} \in \Sigma_m$ such that $\mathbf{y} = A\mathbf{x}$. This provides an upper bound on m that guarantees uniqueness using coherence: $m = \mathcal{O}(\sqrt{M})$ [21].

These theorems ensure uniqueness when $\mathbf{y}$ is error-free. System hardware introduces errors during acquisition in the form of noise during sensing, resulting in $\mathbf{y} = A\mathbf{x} + \mathbf{n}$. In the presence of noise, the theorems that address uniqueness fall short and there is the need for a new criterion, called the *restricted isometry property* (RIP) that dictates the distance between two measurement vectors $\mathbf{y} = A\mathbf{x}$ and $\mathbf{y}' = A\mathbf{x}'$ to be proportional to the distance between the original signals, $\mathbf{x}$ and $\mathbf{x}'$. This would guarantee that, for small $\mathbf{n}$, two sparse vectors that are

far apart do not generate the same measurement vector.

Theorem 3. If A has unit-norm columns, and coherence, μ , then A has the (m, δ_m) -RIP with $\delta_m \leq (m - 1)\mu$ [22].

Random matrices of size $M \times N$ derived from random distributions that are independent and identically distributed, have the (m, δ_m) -RIP with high probability if

$$M = \mathcal{O} \left\{ \frac{m}{\delta_m^2} \log\left(\frac{N}{m}\right) \right\} \quad (2.11)$$

The ideal sensing matrix has a low mutual coherence, and a small RIP constant. Matrices whose entries are derived from a random distribution such as Gaussian or Bernoulli ensure high probability of reconstruction. However, these are quite difficult to achieve and there are no deterministic methods to calculate the RIP constant of a matrix. Thus, the definite structure of a structured matrix can be exploited to overcome these issues [21].

Random Matrices

Random matrices are easy to construct and ensure high probability of reconstruction. But, their storage is expensive and hence, their implementation is very difficult [21]. The recovery becomes infeasible when the signal is of large dimension, as the size of the measurement matrix also increases. The entries of a random Bernoulli matrix take the values $\frac{+1}{\sqrt{M}}$ or $\frac{-1}{\sqrt{M}}$ with equal probability. The entries of a Gaussian matrix are independent and follow a normal distribution $\approx \mathcal{N}(0, \frac{1}{M})$. Such random matrices satisfy the RIP with high probability and ensure sparse recovery using l_1 -minimization [21]. These matrices do not enable efficient matrix multiplication that may speed up the reconstruction because of lack of specific structure. Thus, they are disadvantageous for large data.

Structured Sensing Matrices

From a computational and an application oriented view, it is desirable to have measurement matrices with structure. It is not straightforward to prove recovery guarantees for deterministic matrices, hence random matrices with structure were proposed [23]. In this thesis, we use structured sensing matrices. The random partial Fourier matrix, which consists of randomly chosen rows of the discrete Fourier matrix was studied very early in the research on CS [24]. The Fast Fourier transform (FFT) algorithm can be efficiently used to compute partial Fourier matrix in $\mathcal{O}(N \log(N))$ operations. Comparing this to the expression in Equation 2.11, the FFT operations do imply a complexity gain. Other types of structured random matrices are circulant and Toeplitz Matrices [25]. Another structured sensing matrix introduced in [26] are called the Signature-Row block (SRB) matrices. Hadamard codes are constructed in a strategic manner using signature blocks that arrange patterns into groups that share a certain local signature. These codes can help to analyze a scene (as in imaging) without going through the computationally intensive reconstruction by computing the signal-to-noise ratio (SNR) of different blocks and inferring signatures that are most prominent. This new approach of construction of patterns during the sensing enhances recovery and performance.

2.3.3 Reconstruction Algorithms

There are many algorithms proposed for the recovery of the signal from its compressive measurements, from the same family as the sparse coding algorithms. The sensing matrix is analogous to the dictionary. They are broadly classified into (i) convex optimization-based (ii) greedy algorithms and (iii) combinatorial algorithms.

Convex Optimization-based

The following linear program:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \text{ subject to } \mathbf{y} = A\mathbf{x} \quad (2.12)$$

gives exact signal recovery, provided RIP holds. If measurements are affected by noise, i.e., if $\mathbf{y} = A\mathbf{x} + \mathbf{n}$ where $\mathbf{n}$ is measurement noise, then the optimization problem in Equation 2.12 is modified to

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \text{ subject to } \|A\mathbf{x} - \mathbf{y}\|_2^2 \leq \varepsilon \quad (2.13)$$

for an empirically chosen $\varepsilon > 0$. Equation 2.12 is referred to as the classical Basis Pursuit (BP) problem and Equation 2.13 is called the Basis Pursuit Denoising (BPDN). Although these problems are convex programs with simple structures, solution requires dedicated algorithms because classical methods, such as interior-point algorithms for linear and quadratic programming, are very inefficient on them. In order to deal with this, many algorithms are proposed and continue being developed. One of the earliest first-order methods applied were the projected gradient methods [27] where the authors expressed Equation 2.13 as a box-constrained quadratic program and implemented a gradient projection method. The most widely used first-order methods for solving Equation 2.13 is the iterative thresholding [28] and its variants. Another sparse reconstruction algorithm called SpaRSA [29] was also developed, where the main problem is divided into sub-optimization problems. Such sub-optimization problems are solved more rapidly than the original problem.

Greedy Algorithms

This category of algorithms iteratively approximate the coefficients and supports of the original signal. These algorithms are usually computationally very efficient and easy to implement [18]. Their performance guarantees are comparable to those of the l_1 -based minimization algorithms. One of the most popular greedy algorithms is the *Orthogonal Matching Pursuit*

(OMP). It was first introduced in [30] and various modifications/improvements have been proposed thereafter such as the Regularized Orthogonal Matching Pursuit, Stagewise Orthogonal Matching Pursuit[31], etc. OMP chooses the atom having the maximum correlation with the residue or the error, at each iteration. Thus, at each iteration, the algorithm attempts to estimate the signal, $\mathbf{y}$ with incremental information. The estimation error is termed the residue. The columns of matrix, A are assumed to be l_2 normalized.

The OMP is explained as below:

OMP Algorithm

Input:

$A = \{\mathbf{a}_i\}_{i=1}^N \in \mathbb{R}^{M \times N}$: $\mathbf{y} \in \mathbb{R}^M$: Error threshold, ε

Initialization:

$j = 0$: $\mathbf{x}^0 = 0$: Residual, $\mathbf{r}^0 = \mathbf{y}$: Support, $S^0 = \emptyset$

Steps:

While not converged do:

1. Iterate: $j = j + 1$
2. Choose i_0 such that $\min_c \|\mathbf{c}\mathbf{a}_{i_0} - \mathbf{r}^{j-1}\|_2 \leq \min_c \|\mathbf{c}\mathbf{a}_i - \mathbf{r}^{j-1}\|_2 \forall i$
3. $S^j = S^{j-1} \cup i_0$
4. $\mathbf{x}^j = \arg \min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{y}\|_2$ subject to $\text{supp } \mathbf{x} = S^j$
5. $\mathbf{r}^j = \mathbf{y} - A\mathbf{x}^j$

Until $\|\mathbf{r}^j\|_2 \leq \varepsilon$

Output:

Approximate solution $\mathbf{x}^j$

The combinatorial algorithms, which are designed for structured samples of the original signal, are not as widely used as the convex optimization-based and greedy algorithms. Sparse Representation and CS finds application in multiple areas of signal processing as explained in

2.4 Summary and Motivation of Proposed Works

The concept of dictionary and the utilization of dictionary for different applications is introduced. K-SVD and SGK learning algorithms are discussed. SGK, in terms of complexity is seen to be a superior algorithm to K-SVD. Sparse representation-based dictionary training is used in a range of applications for unsupervised and supervised scenarios such as image processing and classification. Dictionary learning also finds potential use in semi-supervised scenarios. One such application is anomaly detection and the motivation is to propose an algorithm that can perform better than SGK to accomplish the task. Data-driven dictionary learning algorithms are better suited for the required task. With nature of the data known *a priori*, improved objective formulations can be exploited.

Compressed Sensing, an acquisition scheme that exploits sparse representation is reviewed. CS, with its inherent feature to use less number of samples for measurement, is a very attractive scheme for airborne platforms such as drones and imaging platforms such as SAR. The use of CS in a surveillance drone application is exploited and its advantages demonstrated. SAR platform generates innumerable data for processing. Sparse representation and CS, with its ability to reduce the number of samples required for faithful reconstruction and other features like its capability to handle missing data is a relevant area of research and promising in the context of SAR.

Chapter 3

Sparse Representation-based Anomaly Detection

Anomaly Detection is a much sought-after task in most functioning systems. Anomaly detection refers to the problem of finding patterns in data that do not conform to expected behavior. These non-conforming patterns are often referred to as anomalies, outliers, exceptions, aberrations, surprises, or peculiarities in different application domains. Anomaly Detection is widely used in a diverse set of applications such as intrusion detection in cyber security, fraud detection, fault detection in critical systems and in military surveillance and defence.

3.1 Anomaly Detection

A common need when analyzing real-world data-sets is determining which instances stand out as being dissimilar to all others. Such instances are known as anomalies. The goal of anomaly detection is to determine all such instances in a data-driven manner. Anomalies can be caused by errors in the data but often are indicative of a new, previously unknown, underlying process. An anomaly is an observation that deviates so significantly from other observations as to arouse suspicion that it was generated by a different mechanism [32]. Anomalies [33] can be broadly categorized as:

- Point anomalies: A single instance of data is anomalous if it is too far off from the rest of the instances.
- Contextual anomalies: The abnormality is context specific, such as those observed in time-series data.
- Collective anomalies: A collective set of data instances contribute to anomaly detection.

3.1.1 Anomaly Detection Techniques

Simple Statistical Methods

The simplest approach to identifying irregularities in data is to flag the data points that deviate from common statistical properties of a distribution [33], including mean, median, quantiles, etc. Traversing statistical properties over time-series data will necessitate the usage of windowing to smooth the variations in time, such as a low-pass filter. The low pass filter allows for simple anomaly identification; however for cases where the signal exhibits behaviour that are not captured by statistical features, other methods will have to be employed. For example, a noisy signal that blurs the boundary between a normal and abnormal behaviour, or where the distinction between normal and abnormal changes with time, other methods are needed.

Machine Learning Methods

There are multiple machine-learning based anomaly detection methods [33] such as:

- Density-Based: Normal data points occur around a dense neighborhood while anomalies are farther away. Ex., K-Nearest Neighbour (K-NN)
- Clustering-based: Data points that are similar tend to belong to similar groups or clusters, as determined by their distance from local centroids. Ex., K-means
- Support Vector(SVM)-based: SVM learns a soft boundary in order to cluster the normal data instances using the training set, which then, using the testing instance, adapts itself

to identify the abnormalities that fall outside the learned region.

Deep Learning Methods

In the broader field of machine learning, the recent years have witnessed a proliferation of deep neural networks, with unprecedented results across various application domains. Deep learning is a subset of machine learning that achieves good performance and flexibility by learning to represent the data as a nested hierarchy of concepts within layers of the neural network. Deep learning outperforms the traditional machine learning as the scale of data increases [32].

3.1.2 One-Class Classification

One-class classification (OCC) is different from the typical multi-class classification problem because the negative class is either absent or inadequately sampled. The problem of classifying positive (or target) cases in the absence of knowledge of the negative cases (or anomalies) has gained increasing attention in recent years [34]. OCC achieves identification of objects of a specific class amongst all instances, by primarily learning from a training set containing only the instances of that class. This is more challenging than the traditional classification problem, which attempts to distinguish between two or more classes with the training set containing objects from all the classes. Density-estimation, reconstruction by describing structure of the data and boundary methods are the different methods to achieve OCC [35]. OCC without any constraint may design a universal classifier that captures much more space than the training set. A remedy is OCC learning/training with some constraint. Sparse representation is chosen as this constraint. So, OCC with sparse representation is expected to design a tighter classifier than one with no constraint, which would tightly capture the training set of normal sequences.

3.1.3 Sparse Representation-based Methods

Techniques for anomaly detection using sparse representation proves to be effective in specific time series where the sparsity of a signal can be exploited. In [17], the authors use incoherent

dictionary learning method and sparse representation of a signal for specific time series describing the analysed network traffic. The use of Alternating Direction Method of Multipliers (ADMM) to recover simultaneously the sparse representations and the deviated components for the entire collection are discussed in [36]. The authors in [37] propose to exploit convolutional sparse models to learn a dictionary of filters from a training set of normal images and use this to detect anomalies in regions that are characterized by structures that do not adhere to those of the normal images. The work in [38] discusses the use of sparse representation and dictionary learning for anomaly detection in telemetry time series. In Sparse Representation-based methods, the majority of the data vectors are assumed to conform to a sparse representation model, whereas the anomaly is caused by an unknown subset of the data vectors, which significantly deviate from the model.

3.2 New Data-driven Approaches to Training for Anomaly Detection

We propose several ideas of an enhanced and modified sparse-representation based dictionary training for the purpose of anomaly detection. To this end, we use the sequential generalization of k-means (SGK) algorithm for the dictionary training to represent the “normal sequences”, i.e., the training data. The use of an overcomplete dictionary for sparse representation of data is used extensively in dictionary learning. In anomaly detection, the training data set is *a priori*, known to definitively show normal behaviour. A dictionary is “learnt” to represent the known set of data sequences to a reasonable representation error margin.

In any typical dictionary design, the sum of the squared error $\|X - DS\|_F^2$ of the training set $X = \{\mathbf{x}_i\}$ is minimized. Usually, the training set $\{\mathbf{x}_i\}$ consists of a large number of “normally-behaved” sequences $\{\mathbf{x}_j^{(n)}\}$ plus a few “outliers” $\{\mathbf{x}_1^{(o)}, \mathbf{x}_2^{(o)}, \dots\}$. Since $\{\mathbf{x}_j^{(n)}\}$ are large in number, the normally-behaved sequences contribute more towards the sum of the squared error. Therefore, D converges to a dictionary that represents $\{\mathbf{x}_j^{(n)}\}$ well, but produces a large

representation error $\left\| \mathbf{x}_1^{(o)} - D\mathbf{s}_1 \right\|_2^2$ for the outliers such as $\mathbf{x}_1^{(o)}$. Such a dictionary D , referred in this work as the “baseline dictionary”, does not perform well in representation error threshold based anomaly detection.

Anomaly detection is achieved by comparing the representation error $\|\mathbf{x} - D\mathbf{s}\|_2^2$ (subject to the sparsity constraint on $\mathbf{s}$) against a threshold. If $\|\mathbf{x} - D\mathbf{s}\|_2^2$ exceeds the threshold, an anomaly is detected. However, since $\left\| \mathbf{x}_1^{(o)} - D\mathbf{s}_1 \right\|_2^2$ is large, the representation error of an anomaly $\|\mathbf{x} - D\mathbf{s}\|_2^2$ may well be smaller than $\left\| \mathbf{x}_1^{(o)} - D\mathbf{s}_1 \right\|_2^2$. If the threshold is chosen larger than $\left\| \mathbf{x}_1^{(o)} - D\mathbf{s}_1 \right\|_2^2$, then the anomaly is not detected. On the other hand, if the threshold is chosen smaller than $\left\| \mathbf{x}_1^{(o)} - D\mathbf{s}_1 \right\|_2^2$, then the outlier is mis-classified as anomaly.

In the following, in order to improve the performance of the baseline dictionary D , we propose modifications that takes into account the outliers in the training set.

3.2.1 l_∞ Formulation

Instead of minimizing the usual sum of squared error $\|X - DS\|_F^2$, a *minmax* optimization may be used that minimizes the maximum (absolute) error element in $E = X - DS$. In other words, if $\vec{E}$ is the vectorized form of the error matrix, E , then the dictionary training minimizes the l_∞ norm, $\left\| \vec{E} \right\|_\infty$:

$$D = \arg \min_D \left\| (X - DS) \right\|_\infty \quad (3.1)$$

It may be shown that the above *minmax* problem may be written as an overdetermined $\|l\|_\infty$ -norm problem as follows:

$$(X_{n \times N} - D_{n \times M} S_{M \times N})^T \rightarrow S_{N \times M}^T D_{M \times n}^T - X_{N \times n}^T$$

$$S_{M \times N}^T : \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \\ \vdots \\ \mathbf{s}_N \end{bmatrix} \xrightarrow{\text{Convert to}} A_{nN \times Mn} = \begin{bmatrix} \mathbf{s}_1 & \mathbf{0} & \cdot & \cdot & \cdot & \mathbf{0} \\ \mathbf{0} & \mathbf{s}_1 & \cdot & \cdot & \cdot & \mathbf{0} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \mathbf{s}_1 \\ \mathbf{s}_2 & \mathbf{0} & \cdot & \cdot & \cdot & \mathbf{0} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \mathbf{s}_N & \mathbf{0} & \cdot & \cdot & \cdot & \mathbf{0} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix}$$

$$D_{M \times n}^T : \begin{bmatrix} \mathbf{d}_1 & \mathbf{d}_1 & \cdot & \cdot & \cdot & \mathbf{d}_n \end{bmatrix} \xrightarrow{\text{Vectorize}} \mathbf{z}_{Mn \times 1} = \begin{bmatrix} \mathbf{d}_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{d}_n \end{bmatrix}$$

$$X_{N \times n}^T : \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \cdot \\ \cdot \\ \mathbf{x}_N \end{bmatrix} \xrightarrow{\text{Vectorize}} \mathbf{b}_{nN \times 1} = \begin{bmatrix} \mathbf{x}_1^T \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{x}_n^T \end{bmatrix}$$

The dictionary design transformed as shown above is equivalent to the overdetermined system of equations $A\mathbf{z} = \mathbf{b}$ that minimizes the $\|A\mathbf{z} - \mathbf{b}\|_\infty$. Then, $S^T D^T - X^T \equiv A_{nN \times Mn} \mathbf{z}_{Mn \times 1} - \mathbf{b}_{nN \times 1}$ is a vector of sample-wise error [39]. Existing algorithms to solve the overdetermined system rely on the information that, at optimum, the residual vector $r(\mathbf{z}) = A_{mo \times no} \mathbf{z}_{no \times 1} - \mathbf{b}_{mo \times 1}$ has at least $no + 1$ components equal in absolute value and maximal. For full rank systems,

there always exists a solution with exactly $no + 1$ elements of the residual vector equal in absolute value and maximal [40]. The crux of Polya’s method lies in the fact that at optimum, the solution to the minimization problem is equivalent to the solution of a few relevant $no + 1$ row submatrix of A [41]. It determines this crucial subset by iteratively solving the minimization problem for increasing values of p in $\|r(\mathbf{z})\|_p$. Another method, called the Steiefel’s Method iteratively computes the minimum infinity norm solution for a succession of subsystems size $(no + 1) \times n$, by using the exchange method [41].

The disadvantage of the above formulation is that, l_∞ -norm refers to the maximum (absolute) error sample in the training set X rather than the outlier $\mathbf{x}_1^{(o)}$. It is possible that normally-behaved sequences $\{\mathbf{x}_j^{(n)}\}$ may have a noisy sample or two that has the maximum error. l_∞ -norm fails to address the large representation error of outliers, and addresses the large representation error of samples instead.

3.2.2 Baseline Retraining

When the normal sequences themselves show a wide range of representation error with the trained baseline dictionary, the training sequences that show the largest error (outliers) can be chosen as a new data set to retrain the already available baseline dictionary. During retraining, the number of normally-behaved sequences $\{\mathbf{x}_j^{(n)}\}$ are reduced to a number comparable to the number of outliers $\{\mathbf{x}_1^{(o)}, \mathbf{x}_2^{(o)}, \dots\}$ in the training set, such that the representation errors of the outliers contribute significantly to the sum of the squared error. As a result, the retrained dictionary D assigns some of the atoms to represent the outliers. This new dictionary, thus obtained, can represent the entire data set better. This scheme is called baseline retraining. It is important to maintain a balance between the number of normally-behaved sequences and the number of outliers during retraining. Too many normally-behaved sequences do not change the baseline dictionary much. When the number of outliers increases, it changes the baseline dictionary to such an extent that the normally-behaved sequences are no longer well represented. Results for varying numbers are presented in the Data Analysis section. Repeated retraining

may also be used to obtain a desired outcome.

3.2.3 Weighted SGK

In order for the representation errors of the outliers to contribute significantly, these errors may be weighted. The outliers in the training set can be represented even more sparsely if the dictionary is trained relative to the error each sequence contributes to the sum of the squared error in the algorithm. A larger weight should be chosen for a larger error (outlier). This scheme is called weighted training. In the following, we propose to introduce weights in the sequential generalization of k-means algorithm for dictionary design. The modified algorithm is referred as Weighted SGK. We refer to the SGK algorithm in Section 2.2.2 in which the dictionary update step is as follows:

$$\mathbf{d}_{k+1}^{(t+1)} = \arg \min_{\mathbf{d}_k} \left\| E_k^{(t)} - \mathbf{d}_k S_{rowk}^{(t)} \right\|_F^2 \quad (3.2)$$

where the solution is obtained as:

$$\mathbf{d}_{k+1}^{(t+1)} = E_k^{(t)} S_{rowk}^{(t)T} (S_{rowk}^{(t)} S_{rowk}^{(t)T})^{-1} \quad (3.3)$$

and $E_k^{(t)} = X - \sum_{j \neq k} \mathbf{d}_j^{(t)} S_j^{(t)}$ is the error in representing the training set X using all atoms except $\mathbf{d}_k^{(t)}$ of the dictionary at iteration t . In conventional SGK, the dictionary atom $\mathbf{d}_k^{(t)}$ is updated based on $E_k^{(t)}$. In weighted SGK, we consider the weighted error $E_k^{(t)} W_k^{(t)}$ and introduce the weights to Equation 3.3 with:

$$\mathbf{d}_{k+1}^{(t+1)} = E_k^{(t)} W_k^{(t)} S_{rowk}^{(t)T} (S_{rowk}^{(t)} S_{rowk}^{(t)T})^{-1} \quad (3.4)$$

where, $W_k^{(t)}$ is a $N \times N$ diagonal matrix with the diagonal elements representing the factor to which the errors in $E_k^{(t)}$ are scaled, that can be reasonably designed according to the data. i^{th} column of $E_k^{(t)}$ refers to the error vector corresponding to the training sequence $\mathbf{x}_i$. $W_k^{(t)}$ scales

the i^{th} column by w_{ii} , the i^{th} diagonal element of $W_k^{(t)}$. Thus, error of each training sequence is independently weighted. If the weight w_{ii} is chosen proportional to the representation error of each training sequence, then training sequences with larger error will be emphasized. The errors thus scaled according to their magnitude are used in the dictionary update stage. A mathematical formulation to the optimization for this kind of sparse representation-based dictionary learning could be viewed as a *minmax* of the representation error of the data sequences with the dictionary discussed in Section 3.2.1.

Let $E_{n \times N} = X - DS$ be the error matrix at any given iteration. Then $\mathbf{1}_{1 \times n} E_{n \times N}$ denotes the vector of representation errors of N training sequences. Therefore, a possible choice of the weight matrix is $W_{N \times N} = \text{diag}(\mathbf{1}_{1 \times n} E_{n \times N})$. Results for various choices of the weight matrix are described in the Data Analysis section.

3.3 Dataset 1: Numenta Anomaly Benchmark (NAB)

The objective of Numenta Anomaly Benchmark (NAB) [42] is to provide a comprehensive environment of tools to test and measure different anomaly detection algorithms on streaming data. The NAB dataset comprises time-series data where each row contains a time stamp and a single scalar value. NAB claim to include all types of streaming data anomalies, include a variety of data metrics, and challenges such as noise and establishing new normal patterns. Apart from the real data, NAB also consists of artificially generated data that tests anomalous behaviour not represented in the real data and also data without anomalies. This artificial data is used here to test the performance of the algorithms.

3.3.1 Training and Anomalous Data

The normal sequences in the NAB data consists of 4 time-series sequences of length 4033. The test data consists of 6 anomalous sequences of length 4033. Figure 3.1 shows examples of a normal and anomalous sequences.

Figure 3.1: Anomalous Sequence

3 normal sequences are used to train the baseline dictionary itself by using 350 continuous samples from each sequence.

3.3.2 Dictionary Training and Retraining

The initial baseline training is achieved using 3 normal sequences from the data set with 350 samples from each sequence. The training data is generated by dividing the sequences in to shorter windows, using a rectangular window length of 10 with an overlap of 3 samples. The sparsity of each windowed sequence is set to a maximum of half the size of the window. An overcomplete dictionary is designed using 8 times the length of the windowed sequence, i.e., 80 atoms and the size of the dictionary is 10×80 . The dictionary obtained using the initial

training (with 2 normal sequences) is then used for sparse coding the other normal sequences and the average representational error for each sequence across all windows is shown in Figure 3.2. Sequence 3, which shows the highest representational error and Sequence 1 are picked for

Figure 3.2: Anomalous Sequence

retraining the baseline dictionary to maintain a balance between the data and to avoid skewing the retraining in favour of only the sequences with larger errors. The dictionary is retrained again using SGK and Weighted SGK.

3.3.3 Anomaly Detection: Choice of Metrics and Thresholds

With the dictionary obtained after retraining, all training sequences are sparse coded using OMP (as in Section 2.3.3) and the following metrics calculated for each sequence:

- (1) Mean of Absolute Errors for every window in a sequence, met_{T1}
- (2) Maximum Number of dictionary atoms used to represent a window in each sequence or the sparsity, met_{T2} .

The thresholds estimated for the anomaly detection itself are a statistical representation of the above, $\mu(met_{Ti}) + \sigma(met_{Ti})$ where μ and σ are the mean and standard deviation of each metric. All test sequences are represented using the retrained dictionary and each sequence is classified as an anomaly or normal depending on whether the metrics calculated as per (1) and (2) exceed the thresholds. The first condition refers to a large mean error of representation. The second condition refers to a large sparsity.

Figure 3.3 and Figure 3.4 show the plots of met_{T1} and met_{T2} for baseline retraining. All the anomalies are captured with both the metrics together. Figure 3.5 and Figure 3.6 show the plots for met_{T1} and met_{T2} for Weighted SGK retraining. The performance of both the methods are seen to be similar for this dataset. Both methods classify a similar set of sequences as anomalies.

Figure 3.3: Anomaly Detection Metric: Average Representation Error for Baseline Retraining

Figure 3.4: Anomaly Detection Metric: Maximum Sparsity for Baseline Retraining

3.3.4 Performance Comparison of Baseline Retraining with Weighted SGK Retraining

The actual metric used to determine the performance itself is the distance from the ideal precision and recall desired which is $(1, 1)$, where $Precision = \frac{\text{Number of correctly detected anomalies}}{\text{Total number of detections}}$ and $Recall = \frac{\text{Number of correctly detected anomalies}}{\text{Total number of anomalies}}$. The lesser the distance of $(Precision, Recall)$, from the ideal, $(1, 1)$, the better is the performance. This is similar for both methods, as seen from the plots. Table 3.1 shows the detection performance of both methods.

The similar performance of the algorithms can be attributed to the fact that the anomalous sequences have a certain window of anomalous behaviour within the entire sequence and the nature of the signal (as seen in Figure 3.1) within the anomalous window is similar to that of

Figure 3.5: Anomaly Detection Metric: Average Representation Error for Weighted SGK Re-training

Table 3.1: Detection Accuracy

Retraining	Precision	Recall	Distance from Ideal
Baseline	0.8571	1.00	0.1429
Weighted-SGK	0.8571	1.00	0.1429

Figure 3.6: Anomaly Detection Metric: Maximum Sparsity for Weighted SGK Retraining

the normal sequence. Both the dictionaries represent the sequences in a similar manner, both with respect to representational error and sparsity. The anomalous sequences not captured by the met_{T1} are captured by the met_{T2} . However, if the normal and anomalous datasets are not so well structured with the behaviour difficult to analyze, and there are significant number of outliers within the normal dataset itself, the Weighted SGK-based retraining can prove to be more beneficial, as shown in Section 3.4.

3.4 Dataset 2: AIRBUS Dataset

AIRBUS platforms (aircraft, helicopters, satellites, etc.) generate a huge quantity of sensor data that is collected. Almost all data collected from the platforms is considered as normal. The occurrences of faults and failures are very rare and it is not desirable to wait for hundreds of faults to occur before a system could be trained to anticipate such faults. Thus, the interest is to look for changes in the behavior of the systems monitored through the sensors installed on the platforms in a semi-supervised context. Airbus Data Challenge 2019 was announced with the goal of anomaly detection for in-flight sensor data [43]. The data consists of two sets: Training and Validation. All sequences are sampled at 1024 Hz and are 61440 samples long.

3.4.1 Training data

The training dataset is composed of 1677 one-minute-sequences of accelerometer data measured on test helicopters at various locations, in various angles (X, Y, Z), on different flights. All data was multiplied by a factor so that absolute values are meaningless, but no other normalization procedure was carried out. All sequences are considered as normal and should be used to learn normal behaviour of accelerometer data.

3.4.2 Validation Data

The validation dataset is composed of 594 one-minute-sequences of accelerometer data measured on test helicopters at various locations, in various angles (X, Y, Z). Locations and angles may or may not be identical to those of the training dataset. Sequences are to be tested with the normal behaviour learnt from the training data to detect abnormal behaviour. The amount of abnormal sequences in the validation dataset is a priori unknown. The nature or behaviour of the anomaly is not known and the underlying reason for the cause is also unknown.

Figure 3.7: Signal: Normal and Anomaly

3.5 Data Analysis

3.5.1 Preliminary Analysis

A preliminary analysis of the training data was done as part of data preparation. Figure 3.8 shows the statistical mean and standard deviation of each sequence in the training data. Figure 3.9 shows the frequency components in a few sequences from the training set. The sequences that show larger variance are different from those that have large mean in the data set. Figure 3.10 and Figure 3.11 show the distribution of the mean and the standard deviation of the sequences in the training set. While the mean of the sequences show a trend approximating the Gaussian distribution, the standard-deviation itself shows a distinctly different trend, which

could be approximated to the Rayleigh Distribution. Those sequences that show larger variance compared to most could be potential outliers in the training set, on initial analysis. From analysis that follow later, it is seen that they are the outliers in the normal data set. The frequency analysis of the normal data do not show a similar pattern or a sparsity in a known domain, such as the frequency domain. This motivates the need to explore a data-dependent domain in which the signals might be sparse.

Figure 3.8: Statistical Features of Training Data: Mean and Standard Deviation

3.5.2 Sparse Representation of Data set with a Trained Dictionary

Each sequence in the training set is windowed using a rectangular window of length 100 samples and an overlap length of 10 samples. So, with a subsequent signal length of 100 for the data that is used to train the dictionary, we fix an upper limit of 50 for the sparsity ($n/2$) and an error goal of 0.009 for the sparse coding stage in the dictionary training. We train a dictionary of size 100×800 for the data. The number of dictionary atoms was fixed based on empirical trials, for which the lowest MSE was obtained when SGK was run for 150 iterations.

Figure 3.9: Frequency Analysis of Training Data

3.6 Performance of Dictionary Training Algorithms

12 sequences are chosen randomly to form a short training set for the dictionary training. We run SGK to design the trained dictionary for this data, and obtain the “baseline dictionary”. Performance improvements are desired with this dictionary as the “initial dictionary” in subsequent retraining. The entire training set (of 1677 sequences) is then represented sparsely using the baseline dictionary and the error in representing each sequence is calculated. It is observed that some of the sequences have a large error on representation as compared to a large majority. These sequences are “outliers” in the training set.

3.6.1 Baseline Retraining

With the initial “baseline dictionary” that was obtained, two new data sets of different sizes were formed for baseline retraining based on the number of outliers chosen from the represented sequences using the baseline dictionary. (x, y) denotes the representation of the number of outliers, x and number of commonly-behaved sequence, y . *TestA* is run with $x + y = 12$ and *TestB* is run with $x + y = 20$. All parameters for retraining remain the same as that of the

Figure 3.10: Basic Statistical Features of Training Data

baseline training.

3.6.2 Weighted SGK Retraining

While weighted SGK by itself could have been applied on the baseline dictionary, weighted SGK in combination with baseline retraining is observed to perform better. This scheme is referred as weighted SGK retraining. So, *TestA* and *TestB* is run with the introduction of the weights matrix. In the weighted SGK retraining, we use the weights matrix W to scale the errors in the dictionary update stage, which shifts the dictionary behaviour in the direction of the outliers so they can also be reasonably well-represented. Simulations were run for the weighted SGK retraining for various weights that can scale the errors proportional to the magnitude, which

Figure 3.11: Basic Statistical Features of Training Data

generates a different dictionary for each weights matrix used. The training data are then sparse-coded using each of this dictionary, and the corresponding error in representation calculated. Figure 3.12 shows the magnitude of the errors across the sequences for each weights matrix used. The weights matrix used are as follows: (a) Normalized root-mean square error (RMSE) for the errors at each iteration in the training set used for dictionary training, (b) Normalized RMSE constrained in the range of $[1, 2]$ so that the scaled errors range is smaller as compared to (a) and, (c) Normalized MSE.

The performance of (a) normalized RMSE matrix shows a reasonably good performance as compared to the others, and we thus choose this as the weights matrix in weighted SGK retraining.

Figure 3.12: Error Plot for various Weights Matrix used

3.6.3 Comparison of Baseline Retraining with Weighted SGK Retraining

The representation error in the complete set of sequences of the training set are calculated with the retrained dictionary obtained with both baseline retraining and the weighted SGK retraining. Figure 3.13 shows the comparison in the representation error in each sequences obtained using both the dictionaries. It is observed that error to represent the outlier sequences in the training set are lower using Weighted SGK retraining as compared to baseline retraining and the error for the more common sequences, though marginally larger, do not change as much in the Weighted SGK. Figure 3.14 and Figure 3.15 show the distribution of the number of

atoms used in the dictionary to represent the sequence, i.e., the sparsity of the sequences. Both distributions show a unimodal behaviour, thus representing the training sequences in a similar manner, although the Weighted SGK retraining represents them more sparsely. The validation data set is then sparse-coded using both the retrained dictionaries and the distribution of the sparsity in the sequences shown in Figure 3.16 and Figure 3.17. A distinction is seen in the distribution shown in both. While one represents a unimodal distribution, the other represents a bimodal distribution in sparsity with two significant peaks. This can be viewed as two different clusters in the dataset and a significant feature to distinguish between the two.

Figure 3.13: Representation Error in Sparse Coding

3.7 Performance of Anomaly Detection

3.7.1 Anomaly Detection Algorithm

With the dictionary obtained after retraining, all training sequences are sparse coded using OMP and the following metrics calculated for each sequence:

- (1) Mean of Absolute Errors for every window in a sequence, met_{T1}

Figure 3.14: Distribution of Sparsity of Training sequences in Baseline Retraining

- (2) Average Number of dictionary atoms used to represent each sequence or the sparsity, met_{T2}
- (3) Maximum of the errors across all windows in each sequence, met_{T3} .

The thresholds estimated for the anomaly detection itself are a statistical representation of the above, $\mu(met_{Ti}) + k_i\sigma(met_{Ti})$ where μ , σ and k_i are the mean, standard deviations and a weighting constant respectively fixed empirically for each metric. All validation sequences are represented using the retrained dictionary and each sequence is classified as an anomaly or normal depending on whether the metrics calculated as per (1), (2) and (3) exceed the threshold for met_{T1} , fall below the threshold for met_{T2} or exceed the threshold for met_{T3} . The first condition refers to a large mean error of representation. The second condition refers to a

Figure 3.15: Distribution of Sparsity of Training sequences in Weighted SGK Retraining

small sparsity (such as a constant value with sparsity 1). The third condition refers to a large maximum error. With the knowledge of the ground truth available for the validation set, the performance of the algorithm is evaluated. The metric used to determine the performance itself is the distance from the ideal precision and recall desired which is $(1, 1)$. The lesser the distance of $(Precision, Recall)$, from the ideal, $(1, 1)$, the better is the performance.

3.7.2 Results and Discussion

Figure 3.18 and Figure 3.19 show the performance of *TestA* and *TestB* respectively.

We observe that the weighted SGK retraining performs better than the baseline retraining in all cases for the *TestA*, while for a range of outliers, it performs better for *TestB*. It is

Figure 3.16: Distribution of Sparsity of Validation sequences in Baseline Retraining

interesting to note that both perform the best when the number of normally behaved sequences, $y = 3$. Figure 3.20 represents the performance of the algorithm for varied number of sequences with a fixed number of normally-behaved sequences set to $y = 3$ as this gives the optimal performance for both *TestA* and *TestB*. We see that in all cases, the weighted SGK retraining performs better than the baseline retraining. Table 3.2 and Table 3.3 show the accuracy in terms of precision and recall for the cases where the weighted SGK retraining outperformed the baseline retraining.

We hereby note that the dictionary is trained using very few sequences from the training data set ($\sim 1\%$) itself. The performance is reasonably good and improvements from the baseline dictionary are obtained with a similarly small data set too. The time for execution for each

Figure 3.17: Distribution of Sparsity of Validation sequences in Weighted SGK Retraining

Table 3.2: *TestA*: Detection Accuracy

$x + y = 12$	Baseline Retraining		Weighted SGK Retraining	
No. of Outliers	Precision	Recall	Precision	Recall
5	0.9513	0.8552	0.9228	0.8855
6	0.9511	0.8519	0.9393	0.8855
7	0.9401	0.8451	0.9228	0.9259
8	0.9309	0.8620	0.9615	0.9259
9	0.9373	0.8552	0.9617	0.9293
10	0.9270	0.8552	0.9679	0.9125
11	0.9407	0.8552	0.9606	0.9024

Figure 3.18: Performance of *TestA*

Table 3.3: *TestB*: Detection Accuracy

$x + y = 20$	Baseline Retraining		Weighted SGK Retraining	
No. of Outliers	Precision	Recall	Precision	Recall
15	0.9712	0.9091	0.9329	0.9360
16	0.9853	0.9024	0.9685	0.9327
17	0.9889	0.8990	0.9789	0.9360
18	0.9853	0.9024	0.9681	0.9192
19	0.9781	0.9024	0.9577	0.9158

Figure 3.19: Performance of $TestB$

Figure 3.20: Performance with 3 Normal sequences in the training set

iteration of both baseline retraining and Weighted SGK retraining algorithms is comparable, though the latter takes twice the number of iterations to plateau as compared to the former. While established deep learning techniques for anomaly detection promise good performance, the training time could be significantly larger since they typically use the entire data set for training. Weighted SGK-based retraining outperforms baseline retraining for the AIRBUS helicopter data because of the nature of the accelerometer data and the wide range of normal behaviour that is expected.

3.7.3 Comparison with State-of-the-art Anomaly Detection Algorithms

In order to demonstrate the superior performance of the proposed sparse-representation based retraining methods, we compare their performance to some of the state-of-the-art anomaly detection methods. The authors in [44] discuss various state-of-the-art algorithms for anomaly detection in univariate time-series under the broad categories of statistical, machine learning and deep learning approaches. The authors in [45] discuss the use of Gaussian Mixture Models (GMM) based clustering in digital flight data to detect flights with anomalous data patterns. Recent work in [46] also discusses the use of GMM for clustering data obtained from flight data recorders. Hence, we choose a classical machine learning-based approach such as Isolation Forest as mentioned in [44] and a clustering approach based on GMM for comparative study of the performance.

Feature Extraction

Application of the classical methods directly on the time series data or even the differenced data gives highly unsatisfactory results, unlike the proposed sparse representation-based methods that operate directly on the raw data. Hence, feature extraction is necessary for the the classical methods. A set of 24 features were extracted for each signal in the training set of the accelerometer data from both the temporal as well as the frequency domain such as

energy of the signal, autocorrelation peaks, number of zero-crossings, entropy, wavelet-related features, etc. These features were then used for training and threshold-based classification. Both Isolation Forest and GMM-based methods were implemented using [47] on the Python platform. Hyperparameters were tuned for both the Isolation Forest as well as the GMM using a part of the validation data to obtain the least distance from the ideal precision and recall. The rest is used to test the performance.

Comparison in Performance

The comparative results are shown in Table 3.4. The entire training set of 1677 sequences is used to train both the Isolation Forest and the GMM model. There is no outlier based retraining involved in the classical approaches. We present the best performance of the Weighted SGK-based retraining and Baseline Retraining as given by $TestB$ for $y = 3$. Weighted SGK-based retraining outperforms all the other methods followed by baseline retraining. However, it is observed that the metrics chosen for the thresholds in the proposed sparse-representation based retraining $\mu(met_{Ti}) + k_i\sigma(met_{Ti})$ is a sufficient statistic for the threshold with $(k_1, k_2, k_3) = (1, -1, 1)$. Even though the performance improves with the use of validation data to optimize the values of k_i , a basic use of $(1, -1, 1)$ gives a $(Precision, Recall) = (0.9676, 0.9097)$ with a distance from the ideal as 0.0099. These values can be obtained with very few validation sequences. Thus, even with minimal parameter tuning, Weighted SGK-based retraining performs better than the classical methods. This is very beneficial for anomaly detection in particular since prior knowledge of the nature of anomalies is unknown. The Weighted SGK-based retraining captures the nature of the training data in the most concise manner and is represented by just the sparsity and the representation errors. The baseline dictionary and the retrained dictionary use only ($\sim 1 - 2\%$) of the training sequences, with the intelligent choice of the outlier representation. However, the performance of the classical machine learning approaches is dependent on the features extracted, the size of the training set and the hyperparameter tuning based on the validation set.

Table 3.4: Comparison of Performance with State-of-the-art Algorithms

Approach	Precision (P)	Recall (R)	Distance of (P,R) from the ideal (1,1) $(1 - P)^2 + (1 - R)^2$
Isolation Forest [44]	0.8760	0.9130	0.0231
GMM [47]	0.9260	0.9090	0.0137
Baseline Retraining (Proposed)	0.9889	0.8990	0.0103
Weighted SGK Retraining (Proposed)	0.9789	0.9360	0.0045

3.8 Summary

Several modifications are proposed to improve the performance of baseline dictionary. The proposed Weighted SGK retraining outperforms the proposed baseline retraining and the classical machine learning approaches in terms of the classification performance. The proposed weighted SGK retraining, a modified and enhanced improvement to the SGK algorithm, modifies the dictionary update step in the dictionary learning algorithm to better represent signals that do not fall in the more “common” set of the training data. This modification enabled a better representation-basis for the signals and achieved better classification results with the validation set.

Chapter 4

Sparse Representation-based Processing in Synthetic Aperture Radar

High-resolution imagery required for environmental monitoring, earth-mapping, military systems, etc., must be acquired at night and in all kinds of weather. Synthetic Aperture Radar (SAR) was designed specifically for such applications. SAR systems exploit long-range propagation characteristics of radar signals and the sophisticated information processing capability of modern digital electronics to provide high resolution imagery. SAR complements optical imaging systems because it is not restricted by the time of day or atmospheric conditions. SAR technology provides terrain structural information to geologists, oil spill information on water to environmentalists, reconnaissance information to military operations, to name a few.

4.1 Synthetic Aperture Radar

Consider an airborne SAR imaging platform moving perpendicular to the aircraft velocity as shown in Figure 4.1 (from [48]). Typically, SAR produces a two-dimensional (2-D) image [6]. One dimension in the image is called range (or cross-track) and is a measure of the “line-of-

sight” distance from the radar to the target. Range measurement and resolution in SAR are achieved similar to other radars: range is determined by measuring the time from transmission of a pulse to receiving the echo from a target and, in the simplest SAR, range resolution is determined by the transmitted pulse width, i.e., narrow pulses yield fine range resolution.

Figure 4.1: Geometry of SAR platform

The other dimension is called azimuth (or along-track or cross-range) and is perpendicular to range. SAR’s ability to produce fine azimuth resolution is one of the reasons which differentiates it from other radars. In order to obtain fine azimuth resolution, a physically large antenna is needed to focus the transmitted and received energy into a sharp beam. The sharpness of the beam defines the azimuth resolution. Optical systems require large apertures to obtain fine imaging resolution. Airborne radar can collect data while flying this distance, and then process the data as if it came from a physically long antenna. The antenna is “synthesized” by flying a distance and is known as the synthetic aperture. A narrow synthetic beamwidth results from the long synthetic aperture, which yields finer resolution than is possible from a smaller physical antenna. A target’s position along the flight path determines the Doppler

frequency of its echoes. As the aircraft flies a distance (the synthetic aperture), echoes are resolved into a number of Doppler frequencies. The target's Doppler frequency determines its azimuth position. Thus, the SAR employs range as well as azimuth processing to image a target.

4.1.1 Raw SAR Data and its Processing

SAR sensors typically make use of frequency modulated pulsed waveforms for transmission, the “chirp” signals [49]. The instantaneous frequency of the waveform varies linearly with time, t , according to $f r_i = k_r t$ where k_r is known as the chirp rate. τ is the pulse time length, yielding the bandwidth $B = k_r \tau$. This is followed by the echo window time during which the radar “listens” to the scattered echoes and stores them on-board the platform.

The received echoes form a two-dimensional data matrix of complex samples, where each complex sample is given by its real and imaginary part, thus representing an amplitude and phase value, and is called the “phase-history” samples. The first dimension corresponds to the range direction (or fast-time): a range vertical consists of the complex received signal samples after being amplified, down converted to base band, digitized and stored in memory. The radar acquires a range vertical during the distance it travels in the pulse repetition interval (PRI) thus forming the second dimension of the data matrix, known as azimuth or slow-time. The return signal from the illuminated scene are sampled both in fast-time and slow-time.

The SAR signal model as described in [50] is briefly discussed here. The SAR signal from a reflector located at $(x, y) = (X_c + x_i, Y_c)$, where (X_c, Y_c) are the co-ordinates of the center-point of the target area in the range and cross-range domain respectively, is:

$$s_{0i}(t, u) = p \left[t - \frac{2\sqrt{(X_c + x_i)^2 + (Y_c - u)^2}}{c} \right] \quad (4.1)$$

for $u \in (-\infty, \infty)$. (t, u) is the two-dimensional coordinate frame in fast-time and slow-time. Equation 4.1 is termed the SAR reference SAR signal at the reference range bin $x = x_i$. The

fast-time Fourier transform of this SAR signal is

$$s_{0i}(\omega, u) = P(\omega) \exp \left[-j2k\sqrt{(X_c + x_i)^2 + (Y_c - u)^2} \right] \quad (4.2)$$

Thus, the two-dimensional Fourier transform of the reference SAR signal $s_{0i}(t, u)$ with respect to (t, u) is:

$$S_{0i}(\omega, k_u) = P(\omega) \exp \left[-j\sqrt{4k^2 - k_u^2}(X_c + x_i) - jk_u Y_c \right] \quad (4.3)$$

where $k = \omega/c$, the wavenumber and k_u is the spatial frequency for azimuthal synthetic aperture, u or the slow-time Doppler domain. In order to form the SAR image at the range bins x_i 's where $x_i \in [-X_0, X_0]$, the reconstruction of the target reflectivity function, $f(x, y)$ at a fixed range $x = x_i$ is given by:

$$f(x_i, y) = \iint_{k_u, \omega} P^*(\omega) \exp \left(j\sqrt{4k^2 - k_u^2}X_c + jk_u Y_c \right) S(\omega, k_u) \times \exp \left(\sqrt{j4k^2 - k_u^2}x_i + jk_u y \right) d\omega dk_u \quad (4.4)$$

where $P^*(\omega)$ is the complex conjugate of $P(\omega)$. Equation 4.4 can be expressed as:

$$f(x_i, y) = \iint_{k_u, \omega} S_{0i}^*(\omega, k_u) S(\omega, k_u) d\omega \times \exp(jk_u y) dk_u \quad (4.5)$$

Thus, for image formation, the Fourier transform of the SAR signal is first matched filtered with the Fourier transform of the reference SAR signal at its range bin, x_i . The matched filtered signal is summed over the available fast-time frequencies to yield the marginal Fourier transform of the target function, at the range x_i . The target function $f(x_i, y)$ is formed by the inverse Fourier transform of the marginal Fourier transform. All the individual range bin outcomes are put together to form a two-dimensional image of the target function. This is termed the range-stacking digital reconstruction for SAR. This is one of the methods for reconstruction.

Raw SAR data as is, does not give any useful information. It yields meaningful information only after signal processing. Thus the complete processing can be visualized as two separate

matched filter operations along the range and azimuth dimensions: (i) The first step is to compress the transmitted chirp signals to a short pulse by using the transmitted chirp as the reference function. The result is a range compressed image, which reveals only information about the relative distance between the radar and any point on the ground. (ii) Azimuth compression follows this step, i.e., the signal is convolved with its reference function, which is the complex conjugate of the response expected from a point target on the ground. This is shown in Figure 4.2, Figure 4.2 and Figure 4.4, for a synthetic data generated from 4 point targets of different reflectivities. This processing takes place at the ground station, to which the SAR platform transmits large amount of the raw data. The result of the processing is the reflectivity functions of the targets at the reconstructed (*range, azimuth*) co-ordinates.

Figure 4.2: Raw SAR signal

Figure 4.3: Range Compressed data

4.1.2 Advantages of Sparse Representation in SAR Processing

The reflected SAR signal generates a vast amount of data that needs to be transmitted on the downlink to the ground station. Traditionally used compression techniques include Block-Adaptive Quantization (BAQ) and Vector Quantization (VQ) [51]. However, both BAQ and VQ have drawbacks such as poor Signal-to-noise ratio (SNR) and increased computational complexity respectively. So, there is always a demand for new and efficient compression techniques

Figure 4.4: Azimuth Compressed data

to reduce data being transmitted on the downlink. Thus, sparse representation and CS, with its ability to reduce the number of samples required for faithful reconstruction and other features like its capability to handle missing data looks attractive and promising in the context of SAR.

4.2 Applications of Sparse Representation in SAR: A Background

This section is termed “sparse reconstruction-based” rather than CS-based, since this will cover work done with respect to SAR on sparse reconstruction which may or may not be compressive sensing-based. One of the earliest works in radar processing via compressed sensing is described in [52]. M.A. Herman and T. Strohmer describe a technique that makes use of a deterministic signal that is “incoherent” with the measurement matrix, that does away with matched filtering and achieves higher resolution than conventional radar processing. In [53], the authors propose the design of a CS undersampling scheme for SAR by (i) random slow-time undersampling, where random changes to the PRI can break up the periodicity of the aliasing effects and (ii) jittered slow-time undersampling, where regular undersampling is perturbed by random noise.

The SAR image reconstruction is proposed via a modified optimization problem:

$$\mathbf{v}_{rec} = \arg \min_{\mathbf{v}'} \|\mathbf{v}'\|_1 + \alpha TV(|\Psi \mathbf{v}'|) \text{ subject to } \|S - \Phi \Psi \mathbf{v}'\|_2 \leq \varepsilon \quad (4.6)$$

for some empirical $\alpha \leq 0$, $\varepsilon > 0$ and TV is the total variation defined as:

$$TV(\mathbf{x}) = \|\nabla(\mathbf{x})\|_1 \quad (4.7)$$

$\mathbf{v}_{rec}$ is the reconstructed estimate of the reflectivity map, S is the undersampled measurements, Φ is the measurement matrix, Ψ is the basis in which the data is assumed to be sparse and ε is a threshold dependent on the noise in the measurements. ∇ is a discrete approximation to the 2-D gradient operator.

In [54], S.J. Wei *et al.* propose a CS-based reconstruction for SAR which applies to strong scatterers while the weak ones are considered as noise. Here, the minimization of the l_1 - norm promotes the reconstruction of strong scatterers. In [55], the authors propose a CS-based adaptive l_p -norm algorithm (where $0 < p < 1$) for high-resolution in radars in which the algorithm dynamically estimates the optimal value of p and also accomplishes reconstruction. [56] addresses the challenge of choosing an appropriate basis and representation scheme for the sparse representation of the SAR data. The authors present simulated results for data that are sparse in wavelet-basis. In [57], S. Kelly *et al.* describe a SAR image formation framework based on iterative inversion algorithms that also enables for phase-error corrections and partial SAR data. S. Ugur and O. Arikan [58] propose a novel SAR processing technique based on CS for autofocused image reconstruction. It achieves image reconstruction as well as optimal correction of phase errors simultaneously.

[59] discusses the Principal Component Analysis (PCA) as a sparsifying transform for raw SAR data to which traditional CS is applied. They present simulated results on synthetic data. Cetin *et al.* in [60] discuss an overview of the work done in sparsity-driven SAR imaging. They discuss a particular sparsity-driven formulation called “feature-enhanced imaging” which

is given by minimizing the cost functional:

$$J(\mathbf{v}') = \|\mathbf{r} - H\mathbf{v}'\|_2^2 + \lambda_1 \|\mathbf{v}'\|_1 + \lambda_2 \|\nabla|\mathbf{v}'|\|_1 \quad (4.8)$$

where $\mathbf{v}'$ denotes the underlying, complex-valued reflectivity image, H is the mathematical observation model for the SAR, $\mathbf{r}$ represents the phase history measurements. The parameters $\lambda_1, \lambda_2 > 0$ determine which sparsity term is more dominant: (i) the first-one leading to “point-enhanced” imaging and (ii) the second-term leading to “region-enhanced” imaging. [61] and [62] propose the use of low-rank and sparse decomposition (LRSD) of the SAR images for image analysis.

From the above discussion, we see that there is no definite quantification of sparsity for raw SAR data and there is no basis in which all SAR data could be sparse. Thus, sparsity of SAR data is still an open problem and new frameworks and algorithms continue to be proposed towards this end. The authors in [63] propose a CS-based imaging framework for sparse and non-sparse scenario using downsampled raw data. The SAR reconstruction is formulated as double parameter optimization and solution obtained using second order convex approximation. The results using RADARSAT-1 [64] raw data show that the method is insensitive to regularization parameters. Table 4.1 gives the summary of all the prior work done (which have been discussed in the earlier sections) with information that highlights the specific theoretical contributions of each work, type of data that have been addressed and whether any noise was addressed in the work.

Recent works in sparse representation-based SAR classification include a Bayesian sparse representation (BSR) model for SAR image classification [66]. Image classification is achieved by design of frameworks for sub-categories, the BSR model formulation and the decisions for classification. The authors in [67] propose a method to design dictionaries for each dimension in the SAR image, as opposed to a single dictionary in a conventional sparse model, to achieve target recognition in a SAR image. A label-dependent sparse representation algorithm is used

Reference	Contribution	Theoretical Contribution	Type of Data	Noise addressing
[52]	Does not deal with SAR. Achieves higher resolution with CS than conventional radar processing	CS theory used for scene reconstruction instead of the traditional matched filtering. Upper bound on sparsity	Synthetic	Simulated AWGN
[53]	SAR Target scene reconstruction through aperture undersampling	Measurement matrix incorporated in the data acquisition model.	Mostly synthetic and Real SAR data (MSTAR)	Synthetic AWGN. Empirical trials shown better to deal with speckle
[54]	SAR Target scene reconstruction through CS	Measurement matrix incorporated in the data acquisition model	Synthetic data	Synthetic AWGN
[55]	SAR High-resolution imaging through CS	Achieved through adaptive l_p - norm minimization	Mostly Synthetic data one ex of real data	Not explicitly mentioned
[56]	Processing of partial SAR data advanced image formation for mostly sparse bright pixels	Fast iterative back projection algorithms for aperture undersampled data	Public SAR datasets	No
[57]	Sparse representation for SAR data	Choice of appropriate basis iterative solution for the optimization	Synthetic data one real dataset	No
[58]	SAR target scene reconstruction based on novel approach	Approach based on random filter CS	Synthetic data	AWGN
[59]	Conventional CS used on SAR data	None. PCA used as sparsifying basis	Synthetic data	No
[60]	Survey on sparsity-driven SAR imaging	1. Feature-enhanced optimization 2. Autofocusing and Phase error retrieval	Synthetic Data Real datasets	No
[65]	SAR imaging through matrix completion	Matrix completion of range-compressed data, phase-compensated by minimizing rank	Synthetic Data one real dataset	No
[61]	LRSD for SAR data	Joint decomposition and image reconstruction	Synthetic data One real dataset	No
[62]	LRSD of SAR data	LRSD for moving target with background subtraction	Synthetic and real data	No

Table 4.1: Summary of state-of-the-art in sparse representation-based SAR

to recognize SAR target configuration in sparse domain [68]. The work discussed in [69] aims to achieve SAR target classification through joint sparse representation of multi-level dominant scattering images. [70] describes a method to achieve SAR target recognition through linear representation over global and local dictionaries. All the recent work mentioned above make use of the MSTAR data set [71] for experimental results.

4.3 Exploring Phase in SAR with Point Targets

While some SAR reconstruction schemes neglect the phase of the received signal, in this section we explore how phase information may be explored for detection of point targets in SAR. Let $p(t) = a(t)e^{j\beta t + j\alpha t^2}$ be the transmitted pulse, where $a(t)$ denotes the rectangular pulse amplitude, and $e^{j\beta t + j\alpha t^2}$ denotes the chirp signal with chirp parameters α and β . The reflected signal from the targets in the range domain, for one position along the azimuth in SAR is of the form:

$$s(t) = \sum_n \sigma_n a\left(t - \frac{2x_n}{c}\right) \exp\left\{j\beta\left(t - \frac{2x_n}{c}\right) + j\alpha\left(t - \frac{2x_n}{c}\right)^2\right\} \quad (4.9)$$

where, σ_n is the reflectivity coefficient for the n^{th} target, and x_n is the range dimension position coordinate for the n^{th} target, and c is the speed of light. The phase of the reflected signal for multiple targets may be analyzed to find the position coordinates x_n .

4.3.1 Scenario: One Point Target

The phase for one target is:

$$\phi_1(t) = \alpha(t - 2x_1/c)^2 + \beta(t - 2x_1/c) \quad (4.10)$$

The nature of the phase trend for a single point target is quadratic across range bins for a single position of the azimuth, as shown in Figure 4.5. $\phi_1(t)$ is a time-shifted version of the phase of $p(t)$. If, the phase of the transmitted chirp is $\phi_0(t)$, the convolution of the phases can be expressed by:

$$f(t) = \phi_1(t) \otimes \phi_0(t) \quad (4.11)$$

The location of the maximum of the convolved phases, $f(t)$, gives the time shift. The estimated time shift may be used to find x_1 .

4.3.2 Scenario: Two point targets

The expressions below give the received signal for two points with unit (or, equal) reflectivity coefficients, positioned at locations x_1 and $x_1 + \delta x$ along the range dimension.

$$s(t) = \exp \left[j \left\{ \alpha \left(t - \frac{2x_1}{c} \right)^2 + \beta \left(t - \frac{2x_1}{c} \right) \right\} \right] \left[1 + \exp \left\{ j \alpha \frac{4}{c} \delta x \left(\frac{1}{c} (2x_1 + \delta x) - t \right) + j \beta \frac{-2\delta x}{c} \right\} \right] \quad (4.12)$$

The phase of two closely spaced targets is of the form in Figure 4.6. From Equation 4.12, it is observed that the phase is essentially the superposition of the phase of the chirp waveform and the chirp phase modulated with a sinusoid of the form $\exp(j\frac{2}{c}\delta x(2\alpha t + \beta))$. The δx^2 term can be neglected if δx is assumed to be sufficiently small. β and α are known while the unknown

Figure 4.5: Variation of Phase across Range bins at one location along the Azimuth

parameter is δx . This unknown is seen to modify the “jitter” superimposed on the quadratic chirp phase.

The trend of the jitter in the Figure 4.6 and Figure 4.7 is seen to conform with our analysis: As we increase δx , the frequency of the sinusoid increases and hence, the frequency of the jitter itself. $\phi_1(t)$ may be fitted with the time-shifted chirp phase as in Equation 4.11, to remove the phase component due to the first target and the residue (jitter) may be isolated as in Figure 4.8. The period of the jitter may be estimated to find δx . This can be extended to multiple targets scenario. Equation 4.13 gives the jitter observed in the 3-targets scenario, positioned at x_1 , $x_1 + \delta x$ and $x_1 + 2\delta x$ (for simplicity). The jitter $w(t)$ is observed to be a modulated form of the time-delayed chirp with unknown parameters x_1 and δx .

$$w(t) = \exp \left(j \left\{ \alpha \left(t - \frac{2x_1}{c} \right)^2 + \beta \left(t - \frac{2x_1}{c} \right) \right\} \right) \exp \left\{ -j\beta 4 \frac{\delta x}{c} + j\alpha \left(\frac{4}{c^2} (4\delta x^2 + 4x_1\delta x) - 4 \frac{t}{c} (2\delta x) \right) \right\} \quad (4.13)$$

This equation gives the phase component due to the remaining two targets. The contribution from each target can be removed by employing source separation techniques.

Signals reflected from the target scene are received at each position along the azimuth domain. Sparse sensing can be exploited along the azimuth domain to operate on the phase of the received signals. In a noisy scenario, it can be effectively used to reconstruct the point target scene parameters. This could be extended to the reflected signal in 2-D as well.

Figure 4.6: Phase of the received signal from Two Point Targets, $n = 2$, $\delta x = 30m$

Figure 4.7: Phase of the received signal from Two Point Targets, $n = 2$, $\delta x = 60m$

Figure 4.8: Isolation of the Jitter contributed by the 2nd target

4.4 Compressed Sensing-based Low Rank and Sparse Decomposition (LRSD) of SAR data

Presence of noise limits the performance of sparse representation based SAR reconstruction schemes. This work proposes using low rank and sparse decomposition (LRSD) to reduce noise. Like sparsity, low rank is also an atomic property of a matrix such that the matrix may be represented using low dimensions. Rank minimization is achieved using the nuclear norm. Let Y be a matrix that is the sum of a low rank matrix L and a sparse matrix S . Since sparsity and low rank are different atomic properties, it is possible to decompose Y into these components. Candes *et al.* [72] propose a “robust principal component analysis” (RPCA) framework where the authors describe conditions under which a matrix (which is essentially a superposition of a low-rank and sparse component) can be decomposed exactly by solving a

convex program called *Principal Component Pursuit* which minimizes a weighted combination of the nuclear norm and the l_1 norm. The problem to be solved is:

$$\min \|L\|_* + \lambda \|S\|_1 \quad s.t. \quad L + S = Y \quad (4.14)$$

where L , S are the low-rank and sparse components to be recovered from Y , λ is the weighting regularization parameter, Y is of size $n_1 \times n_2$ and $\|\cdot\|_*$ is the nuclear norm of the matrix. In [72], the authors also propose a choice for $\lambda = 1/\sqrt{\max(n_1, n_2)}$ as a good rule of thumb, which can be appropriately tweaked for best results. The algorithm used to solve Equation 4.14 is the *Augmented Lagrangian Method* (ALM). The authors propose to use singular value thresholding (SVT) by alternating directions to efficiently solve the problem. However, this algorithm is not very computationally efficient for large sized data.

4.4.1 Simulations: Synthetic Experiment

In this section we illustrate the use of LRSD for the reconstruction of synthetic SAR data. In this example, no compressed sensing is used since all elements of Y are used in the reconstruction. We generate a synthetic SAR signal with two point targets of relatively low reflectivity coefficients and the following specifications given in Table 4.2. We generate a synthetic SAR

PRF	300 Hz
Platform speed	200 m/s
Carrier frequency	4.5 GHz
Chirp Pulse Duration	2.5 μ s
Bandwidth	100 MHz

Table 4.2: Synthetic SAR signal specifications

signal reflected from two point targets and superimpose that with a low-rank noise. Figure 4.9, Figure 4.10 and Figure 4.11 show the visualizations of the noisy raw SAR data, the range-compressed noisy data and the azimuth-compressed noisy data respectively. Now, we apply the LRSD algorithm proposed in [72], obtain the low-rank and sparse components and reconstruct

their corresponding SAR-processed images. Figure 4.12 shows the range-azimuth processed image of the sparse component obtained as a result of the LRSD. Figure 4.11 is visually quite unclear and the second point target can hardly be noticed in it whereas, Figure 4.12 is visually far superior and both targets are well localized because the low-rank component effectively removed the “noise” that was introduced in the signal.

Figure 4.9: Noisy Synthetic raw SAR data

In order to achieve compressed sensing, Y should be some measurements of $L + S$ rather than $L + S$ itself.

4.4.2 Proposed Method of CS-based LRSD

The work in [73] proposes a CS-based LRSD from compressive measurements. The proposed algorithm formally achieves the decomposition for *a priori* known rank of L and sparsity of S . The authors in [74] propose a variational framework to the problem introduced by Candes *et al.* [72] by formulating the problem as :

$$\min_{L,S} \|A_t(L, S) - Y\|_2^2 \text{ s.t. } \max(\|L\|_*, \lambda_{max}\|S\|_1) \leq \tau \quad (4.15)$$

Figure 4.10: Range-Compressed noisy SAR data

Figure 4.11: Azimuth-Compressed noisy SAR data

where A_t is a linear operator, $\lambda_{max} > 0$ and τ is the error tolerance. The above problem is solved using Newton's Method. The approach is agnostic to the linear operator A_t . So, we adapt the A_t to be a linear transformation on the measurement matrix, A that generates compressive measurements, Y . The method is shown in Figure 4.13. As discussed in [73], the

Figure 4.12: Range/Azimuth compressed sparse Component after LRSD and SAR processing: Target1(1320, 210), Target2(1480, 270)

extension of CS to the LRSD problem requires that the conditions for (i) CS recovery (ii) low rank minimization and (iii) feasible robust PCA, are met. Requirements of a measurement matrix that guarantees CS recover are discussed in Section 2.3. If $\mathcal{A} : \mathbb{R}^{m_1 \times m_2} \rightarrow \mathbb{R}^p$ is a linear map and $m_1 \leq m_2$, then for every integer lr with $1 \leq lr \leq m_1$, the lr -restricted isometry constant be the smallest constant, $\delta_{lr}(\mathcal{A})$ such that

$$(1 - \delta_{lr}(\mathcal{A})) \|L\|_F \leq \|L\| \leq (1 + \delta_{lr}(\mathcal{A})) \|L\|_F \quad (4.16)$$

holds for all matrices L of rank at most lr . This is called the lr -restricted RIP [75]. The authors in [75] show that matrices derived from nearly isometric random variables, such as Gaussian and Bernoulli, will obey the lr -restricted RIP. In order to deal with large-sized data, such as in SAR, we propose to use the structurally random matrix discussed in [76]. $A = \sqrt{\frac{N}{M}} QFR$ where $R \in N \times N$ is a uniform random permutation matrix, $F \in N \times N$ is an orthonormal matrix such as FFT and $Q \in M \times N$ is a subsampling matrix that selects a random subset of rows from FR . Under mild conditions, majority of the entries in $A\psi$ where ψ is an orthonormal

Figure 4.13: Compression in SAR with Proposed Method

basis, behave like Gaussian random variables. Thus, A can be considered to be a member of the nearly isometric random family of variables, if ψ is assumed to be the identity matrix. For Robust PCA to have unique solutions, there must exist a degree of incoherence between L and S . This would depend on the data of interest.

4.4.3 Proposed Algorithm

1. $A = QFR$, where F is the FFT matrix.
2. $Y = AX$, where X is the input data.
3. Adapt the linear operator A_t in the LRSD algorithm to the A , by passing the adjoint of A .
4. Empirically determine the value of λ_{max} that gives the most useful decomposition.

5. Rearrange any estimate of the original vector obtained during the runtime of the algorithm into a 2-D matrix of size compliant with the original raw data
6. During the reconstruction of the data, use A^T
7. Outputs: L and S

4.4.4 Simulations and Results: Real Data

We consider an example of real SAR data set collected by the BYU/Artemis microSAR system [77] that was flown as part of the Characterization of Arctic Sea Ice Experiment 2009 (CASIE-09). This dataset [78] consists of a 13 sec portion of one segment of the flight. The proposed CS-based LRSD framework is used with a part of this data. We present results for *measurement ratio* (MR), $M/N = 0.95$, which is 95% of the samples taken into the CS domain.

The low-rank component shown in Figure 4.15 essentially captures most of the textural information that can be seen in the original SAR image shown in Figure 4.14. The proposed reconstruction has less noise compared to Figure 4.14 since the sparse component is removed. Figure 4.16 shows the reconstruction from the sparse component, which essentially is noise. This is an example where one of the components show the necessary information. The earlier synthetic example showed how the useful information was contained in the sparse part. This is not a contradiction. For point targets, the SAR data was indeed sparse, so the noise synthetically added was low-rank to distinguish it from the sparse signal. Thus, the information that is required can be obtained by either the sparse or the low rank component, depending on the nature of the data. Another example of the low rank decomposition is shown in Figure 4.17 for a different value of the weighting parameter, λ_{max} . Figure 4.18 shows the SAR image processed from the data that was reconstructed using SpaRSA, the CS-based reconstruction algorithm discussed in Section 2.3.3. The MR decides the extent to which the image can be faithfully reconstructed. The example here is shown for a $MR = 0.95$. For lower MR , the peak-signal-to-noise ratio (PSNR) of the image with respect to the original deteriorates and more and more

Figure 4.14: Original SAR Image reconstructed from the raw data (with SAR processing)

information is lost from the image. However, for SAR, even as less as 5% compression in data amounts to a significant saving in storage as well as the data that needs to be transmitted on the downlink.

Image Quality Analysis

Metrics to compare performance of compression algorithms [79] can be applied at the (i) data domain, which would primarily target data fidelity after decompression and (ii) image domain, which would deal with features of the image and its quality. The proposed method achieves a simultaneous decompression and decomposition of the compressed raw data and focuses on analysis of the image after SAR processing. Hence, metrics in the quality of image are more

Figure 4.15: Low Rank Component reconstructed after SAR processing:: $\lambda_{max} = 0.95$, $MR = 0.95$

relevant here. There are established metrics to compare subjective image quality of SAR images:

- Image Contrast (IC): A metric to describe the quality of a SAR image. Higher the image contrast ratio, higher is the crispness of the image. A low image contrast ratio suggests a washed-out image. $IC = \frac{\sigma_{image}}{\mu_{image}}$, is thus the ratio of the average intensity of a distributed clutter background to the average intensity of a no-return background [80].
- Average Local Contrast (LC): Local Contrast ratio is the contrast ratio of a small part of an image. Average Local Contrast, over multiple small regions can be representative of a global contrast in the image. The global contrast is essentially indicative of visual

Figure 4.16: Sparse component reconstructed after SAR processing: $\lambda_{max} = 0.95$, $MR = 0.95$

image perception [81].

$$LC = \frac{1}{w \times h} \sum_{i=1}^{w \times h} lc_i \quad (4.17)$$

where, lc_i is the local contrast for pixel i , and is computed as the average magnitude of the difference between each pixel and its immediate neighbours.

The comparison of these metrics for the images seen earlier (Figure 4.14, Figure 4.18 and Figure 4.15) are given in Table 4.3. Though these are image-specific metrics to compare quality of different images, the higher image contrast in CS-based LRSD processing concurs with the visual analysis earlier. The average local contrast in all images is almost similar, which is also promising because the one obtained with compressed data compares well with the original

Figure 4.17: Low Rank Component reconstructed after SAR processing:: $\lambda_{max} = 1.5$, $MR = 0.95$

Table 4.3: Image Quality Comparison

Image obtained with:	Image Contrast	Average Local Contrast
Uncompressed	0.0193	0.0372
CS-compressed	0.0191	0.037
CS-based LRSD	0.0202	0.0372

image obtained without compression.

Figure 4.18: SAR Image reconstructed from the CS-encoded SAR data ($MR = 0.95$)

4.4.5 CS-based LRSD for Speckle Noise Processing

The earlier analysis in Section 4.4.2 discusses the noise removal from SAR raw data if the noise distribution were assumed to be additive. In most SAR images, however the coherent nature of the radar illumination causes the grainy appearance or the “speckle effect”. The speckle is caused by the multiplicative noise [82]. The authors in [83] review and compare the multiple filters for speckle processing in SAR images. Multiplicative noise in SAR images is generally transformed to additive with the use of logarithmic transformation. [84] discusses an algorithm for SAR image denoising using clustering and sparse representation. The authors in [85] propose the use of non-local priors with sparse representation for suppression of speckle in SAR images. An algorithm for SAR image denoising based on sparse representation using

pre-trained dictionary model is proposed in [86].

We extend the CS-based concept proposed in 4.4.2 to speckle noise processing. We propose the speckle noise removal using low rank and sparse decomposition with the separation of phase component as a prior. The distribution of the multiplicative noise is assumed to be non-negative and have a mean of 1. The nature of the SAR dataset in [77] is such that the raw data is real and the phase is either π or $-\pi$. We hereby propose an algorithm for the case of raw SAR data where the phase takes binary values.

Prior: $X_{org} = \phi V$, where ϕ is the phase or the sign of the data and V is the magnitude of the data.

Assumption: $X_{noisy} = X_{org}N$, where $N \geq 0$ and has mean of 1

- Estimate the phase(sign) $\hat{\phi}$ from X_{noisy} , $\hat{\phi} = \text{sign}(X_{noisy})$, since $N \geq 0$
- $X_l = \frac{X_{noisy}}{\hat{\phi}}$, $X_l \geq 0$
- Transform to logarithmic domain: $\log X_l = \log \phi \hat{\phi} + \log V + \log N$
- $X_t = \log X_l$ as input to CS-based LRSD processing as discussed in Section 4.4.2: $X_t = X_{lr} + X_{sp}$
- Transform to exponential domain: $\hat{V} = e^{X_{lr}}$ and $\hat{N} = e^{X_{sp}}$ for the magnitude and noise estimates
- Estimate $\hat{X}_{org} = \hat{\phi} \hat{V}$

Any zero values in the data before transformation to the logarithmic domain are converted to a positive value (of 1). The algorithm is tested with the BYU/Artemis dataset in [78]. The low rank component after speckle suppression is shown in Figure 4.19. The image is observed to be visually much clearer than the low rank components obtained earlier using the additive noise removal. The textural features are observed to be better preserved after the speckle suppression. Figure 4.20 shows the speckled sparse component after the decomposition and subsequent SAR image processing.

Figure 4.19: Speckle-suppressed Low Rank component reconstructed after SAR processing: $\lambda_{max} = 0.95$, $MR = 0.95$

4.5 Summary

BAQ is the most popular algorithm for the compression of SAR data. FFT-BAQ is known to outperform the BAQ, but leads to increased on-board computational complexity. CS achieves simple encoding, as in the CS-encoded data obtained with the proposed CS-based LRSD using the precomputed structured random matrix. This is very useful on-board SAR platforms, to compress and transmit the data, with all the computational complexity offloaded to the decoder side at the ground station. CS techniques could outperform other conventional compression algorithms for SAR depending on the nature of data. If the nature of data is inherently sparse, in applications such as isolation of strongly reflecting scatterers, or if the data is sparse in some

Figure 4.20: Speckled Sparse component reconstructed after SAR processing: $\lambda_{max} = 0.95$, $MR = 0.95$

transform domain, and if the data is textural, CS-based methods can be utilized. CS-based LRSD with phase separation prior is seen to be effective in speckle suppression.

Chapter 5

Compressive Sensing-based On-board Intrusion Detection for a Surveillance Drone

Surveillance systems are widely used not only for military but also for everyday civilian purposes. Intrusion-detection schemes are of interest in every surveillance system. Mobile flying platforms such as drones are being used effectively as autonomous surveillance systems that can not only detect an intrusion on-board, but also make an informed decision to either track the intruder or transmit the required information to the ground station. Efficient techniques to implement a surveillance system on-board continue to be explored. We implement a compressive sensing-based scheme for intrusion detection on-board the AR Parrot Drone. CS-based scheme makes efficient use of resources available on-board, and with the intrusion being detected on-board, it also makes efficient usage of the bandwidth available while transmitting the frame information to the ground-station. Performance metrics, namely, encoding time and detection accuracy, were evaluated on a Cubieboard that emulates the AR Parrot drone platform.

5.1 Background

Demand for security systems has been growing to meet the need for increased surveillance from intruders in the premises of offices, organizations, residential houses and so on. Most surveillance systems widely used are capable of capturing information, storing them and transmitting the data to a processing station where the data are processed and a decision taken, or some cases even depend on manual decision-making. The more recent systems focus on learning and decision-making abilities without the use of an external intervention.

A video surveillance system is designed and target detection and identification abilities are implemented in [87]. Frame difference and background subtraction are used to achieve the task of detection. Frame difference enables to capture the intricate differences between the frames. Background subtraction is deployed to isolate the target from the background. The authors in [88] implement a smart surveillance system with embedded Complex Event Processing (CEP) technology to assist the intrusion detection by correlating raw data extracted from different sources. They use multiple classifiers to predict the event occurrences from the raw data sequence pattern acquired from the door sensors and surveillance camera with template matching algorithm. The authors in [89] survey the different wireless sensor network techniques used in border surveillance and intrusion detection. Many human intruder detection systems utilize various sensors such as the geophone, microphone, hydrophone, infrared (IR) and surveillance camera. The authors propose the possibility to integrate the whole system as a multi-sensing paradigm to provide effective surveillance and detection of human intruders. All of the above works utilize data captured from stationary sensors.

Unmanned Aerial Vehicles can provide a mobile platform for effective intrusion detection. Japanese company Secom has launched a security service that utilizes this technology to provide an innovative security solution, as cited by one of the world's leading news sources in the security and technology sphere [90]. Secom has built a UAV equipped with a surveillance camera and a light emitting diode to detect intruders and suspicious objects.

5.1.1 Leveraging CS for On-board Processing

The task of technology-aided intrusion detection can be achieved with various kinds of systems; such as cameras and radars with each having its own advantages and limitations. Imaging radars entail extensive computations and overheads for intrusion detection and provide limited information. The features of the proposed on-board/real-time processing are: *(i)* Bandwidth efficiency: the scheme for intrusion detection is implemented on-board. Hence, only the frames marked as “intruded” are transmitted to the ground station, as opposed to detection being done at the ground station where every frame would need to be transmitted *(ii)* Real-time detection: since each frame is not transmitted to the ground, the latency of detection at the ground station and in the round-trip of transmission is avoided *(iii)* Efficient on-board processing: less power requirements is critical because higher the power consumed, lesser is the battery-life and lesser is the time of flight of the drone.

Surveillance systems with High Definition (HD) cameras may need manual intervention once an intrusion is detected. In video applications, especially surveillance, the goal is usually not to exactly reconstruct the compressed video, but mostly an inference problem with respect to detection, tracking, recognition or classification. Even though this kind of detection, estimation or classification can be done subsequently after the reconstruction, it is extremely beneficial if all of the above could be achieved prior to reconstruction since that would translate to saving of computational resources and time. One of the earliest works in this area, the compressive-sensed domain is [91], where the authors propose an approach for background subtraction using compressive measurements without reconstruction. Through this approach, they show that it is possible to recover silhouettes of foreground objects by learning the compressive-sensed representation of the background image, thereby making it unnecessary to learn the background image itself. Davenport et al. in [92] show that a wide range of statistical inference tasks, such as signal detection and estimation, can be achieved through the CS representation of the signal without having to ever reconstruct the signal itself.

Limited processing power on-board small drones is a bottleneck. Typical compression and

detection algorithms would involve intensive processing. In order to tackle this issue and to improve utilization of the bandwidth, CS is proposed for the intrusion detection. The encoding block of CS is simple, as compared to the usual compression schemes. The gain on the simplicity and ease in encoding is offset with the computationally-intensive decoding and reconstruction, but this load falls on the decoder. The decoder at the ground station can be assumed to have sophisticated computational resources and so, infrequent and on-demand reconstruction in the case of an event detection would not be a cause for concern.

5.2 Implementation

The Parrot AR Drone 2.0 is a quadricopter capable of being remotely controlled. It consists of a Parrot P6 processor (32 bits, ARM9-core at 468 MHz) and a camera that can operate at a frame-rate of 15 frames per second [93]. The software that runs on-board the drone is not accessible to users. Though a SDK is provided to remotely communicate with the drone, it does not permit any on-board processing. In order to prove the feasibility of CS-based schemes on-board, the drone firmware is halted to reduce other costs. All access to devices as in any Linux system, was possible with the flight control and navigation shut down. Thus, the drone is operated in the stationary mode.

5.2.1 System Design

The surveillance system design consists of a detection module that implements an encoder, a detection algorithm and a decoder. The on-board encoder receives data from the HD video camera. The illustration of the scenario under test is shown in Figure 5.1. In order to reduce the processing requirement, detection is carried out on a region-of-interest (ROI) rather than on the entire frame. A fixed region is marked within the frame as the ROI. The encoder converts the data within the ROI to the compressed sensed domain by using a Gaussian random matrix. This encoded data is used by the detection algorithm to make the decision for intrusion. If an

intrusion is detected, the entire frame is encoded and the data captured is transmitted as UDP packets to the ground station where the decoder reconstructs the data using an l_1 -optimization algorithm [18]. The schematic flow of the detection algorithm is shown in Figure 5.2. Here, M is chosen ad-hoc to be $0.5N$ such that the number of measurements is only 50% of the pixels, which enables optimal reconstruction. Though lower measurement ratios could yield similar or slightly lower reconstruction accuracy, 50% of the pixels is used since there is no real concern for bandwidth availability in the scenario. The encoded frames can be transmitted either via (i) Frame-wise Transmission, where each frame captured by the camera is transmitted to the ground station or (ii) Selective Frame-wise transmission, where frames are transmitted only when there is a significant difference from the previous frame. We refer to [94], which discusses the details of CS encoding of frames and various modes of operation of the AR drone. However, for the system under consideration which implements intrusion-detection (explained in Section 5.2.2), the intrusion decision is made on-board and only the frames marked as “intruded” are transmitted to the ground station, hence bandwidth and frame-rates are not really a concern.

Figure 5.1: Illustration of the Test Scenario

5.2.2 The Detection Scheme

The concept of using data in the compressed-sensed domain for estimation and detection is proposed in [95]. A ROI of a smaller size is marked within the frame of interest, i.e., the area in need of the surveillance. The difference in ROI of the current frame to the initial/o-

original frame is encoded to the compressed-sensed domain using a Gaussian random matrix that compresses the data, similar to background subtraction in [91]. This encoded data is then used to calculate the statistical feature, standard deviation. The difference in standard deviation of CS-encoded successive ROIs is compared against a threshold that is empirically chosen for the data of interest, and a decision of intrusion appropriately made. The empirically set threshold is the same across all data sets tested. The performance of the detection algorithm is tested and found to be robust to a wide range of thresholds. This could be attributed to the reason that the algorithm deals with the difference in statistical features of the compressive measurements of successive ROIs. Once an intrusion is detected, the entire frame is encoded using a SRM and then transmitted to the ground station, where the encoded data is reconstructed using a CS-based reconstruction algorithm such as sparse reconstruction-based separable approximation (*SpaRSA*) [29]. *SpaRSA* is much faster and is suitable for solving large scale optimization problems.

Figure 5.2: Schematic Flow of the Intrusion Detection

Embedded Implementation of the Sensing Matrix

The challenging task in the implementation of a CS-based algorithm is the design of the sensing matrix at the encoder that enables hardware-friendly and optimal fast encoding. The sensing matrix for the ROI is of smaller dimension while the sensing matrix for the entire frame is of larger dimension. A Gaussian random matrix is used here, which usually restricts the use of

large-sized data. However, a smaller-sized ROI is used within the entire frame which is efficiently encoded by a Gaussian random matrix. The Gaussian random matrix for the required ROI size is pre-stored on the platform for quick access. For the encoding of the entire frame, in case an intrusion is detected, structured random matrix (SRM) discussed in Section 4.4.2 is used for its ease and ability to handle large data. In the SRM implementation, the input is prandomized by flipping the sample signs and then transform it using a Fast Fourier Transform (FFT) before picking up the M random measurements. The use of FFT makes SRM more convenient to use for large-sized data as it reduces the time required to encode. The selection vector to pick up the random measurements is stored as a lookup table in files. Multiplication by a Gaussian random matrix of size $M \times N$ requires $\mathcal{O}(MN)$ operations, while multiplication by a SRM of size $M \times N$ requires $\mathcal{O}(N \log N)$ operations. Though a Gaussian random matrix and a structured matrix are used to encode the data, there are other sensing matrices which can be used too such as deterministic sensing matrix [96].

Power Consumption

The on-board firmware is very loaded on the processor and consumes $\sim 47\%$ of the CPU power on an average, when the drone is not flying. If an intrusion is detected, the frame is required to be compressed and transmitted to the ground station. At 10 *fps*, CS encoding of 256×256 sized frames using SRM consumes $\sim 44\%$ of the CPU for 50% measurement ratio. The detection scheme utilizes a much smaller ROI of the size 70×60 . The encoding involves a multiplication of the ROI with a Gaussian random matrix is efficiently implemented. The detection scheme itself does not contribute much to the computational time, as described in Section 5.3.2. Thus, it is reasonable to claim that the power consumed for the CS-based detection itself is much lesser compared to the power consumed during the CS encoding of the entire frame. The conditional transmission of frames in the event of intrusion reduces the amount of power used as compared to the compression and transmission of multiple frames to the ground station for detection.

5.2.3 Implementation on Testbed

We emulate the on-board system using a testbed consisting of a Cubieboard which is a powerful armboard with AllWinner A10 SoC. It features a 1 GHz ARM Cortex A8 CPU, with 1GB of DDR3 RAM at 480 MHz. It is connected to the system emulating the ground station via an ethernet cable supporting $1 \times 10/100$ speeds. The frames captured are of size 256×256 pixels. A ROI of a smaller size such as 70×60 is marked within the frame of interest. This size is chosen to suit the region of interest within the surveillance scene. The chosen ROI is encoded to the compressed-sensed domain using a Gaussian random matrix that compresses the data to 50%.

5.3 Results and Discussion

We test our CS-based detection algorithm on 5 different datasets recorded at the premises of Airbus, Singapore. These datasets comprise intrusion-free or clean sequences of a surveillance scene of interest interspersed with “intruders” attempting to intrude in to the area of interest (marked by the ROI). These datasets were recorded using the HD video camera on-board the AR Parrot Drone 2.0 at a rate of 15 frames per second [93]. The AR Drone camera produces a raw HD image of size 1280×720 , these frames are then down-sampled to 256×256 image to convert the data into a single UDP packet for transmission to the ground station. Each data set is a video sequence of duration 25 – 30 seconds with a different subject or intruder, which were then used as test sets on-board the Cubieboard platform.

Performance Metrics: In order to evaluate the “correctness” or the use of our scheme, we propose to use the following metrics:

(i) *Detection Accuracy:* This determines if all the frames that are known to have an intrusion have been correctly identified. It also accounts for frames that are wrongly marked as “intruded” too. It is given by:

$$DA = \frac{NF - WF - MF}{NF} \quad (5.1)$$

where DA is the detection accuracy, NF is the total number of frames in the dataset, WF is the number of frames wrongly detected and MF is the number of missed frames for detection. It is expressed in percentage of the total frames for each dataset under consideration.

(ii) *Encoding Time*: Encoding time is the time taken to encode or compress the higher-dimensional data into the smaller-dimensional space. The primary challenge in any encoder is to enable fast encoding. Here, we compare the CS-based encoding times against a JPEG-based encoding scheme for different sizes of ROI. For this purpose, we use a JPEG encoder available open-source [97].

5.3.1 Detection Accuracy

The “detection accuracy” is calculated as a proportion of the correctly identified frames over the entire dataset. The detection accuracy for the algorithm is given in Table 5.1. It can be

Table 5.1: Detection Accuracy

Dataset	Detection Accuracy (%)
1	99.76
2	100
3	98.75
4	98.48
5	99.52

observed from the results that the algorithm performs at a high level of accuracy for the datasets. It is interesting to note that there were no missed detections and the accuracy was only affected by a small proportion of wrong detections. This is desirable in any security system, because while false detection results only in wastage of time (in investigating the assumed intrusion), missed detection is potentially much more harmful. We also tested the detection accuracy for varying measurement ratios for the ROIs. The results of the comparison are given in Table 5.2. We see that the detection algorithm performs well even if measurement ratio is further reduced. This is because the ROI preserves the statistical properties in the compressed-

Table 5.2: Detection Accuracy for Varying Measurement Ratios

Measurement Ratio	Detection Accuracy (%)
0.5	99.52
0.25	99.52
0.20	99.52
0.10	99.52
0.05	99.52

sensed domain even for low measurement ratios. This further reduction in measurement ratios can lead to reduction in CS encoding time, hence the overall time taken to detect an intrusion. Measurement ratios can be further reduced down to a lower limit as discussed in [95] if the processing requirements demand that.

5.3.2 Encoding Time

The “encoding time”, as expected, accounts for the major overhead ($\sim 90\%$) in the detection algorithm itself. We compare the encoding time for the CS-based detection scheme to traditional JPEG encoding time for various sizes of the ROI. The figure below shows the improvement of CS-based encoding time over JPEG encoding time. It is observed that as the size of the ROI increases, the improvement in encoding time for the CS-based encoding reduces. For the scenario of interest, CS-based encoding outperforms JPEG encoding as the area of interest is only a fraction of the entire frame itself.

5.4 Compressed-sensed Domain Processing from the viewpoint of Edge Computing Analytics

Edge computing is a new concept that has emerged in the last few years to bring the services and utilities of cloud computing closer to the end-user [98]. It is characterized by quick processing

Figure 5.3: Encoding Time

time and low latency. Edge computing is envisioned to enable many future technologies like 5G, Internet of Things (IoT) and vehicle-to-vehicle communications. It is a potential solution to delay-sensitive applications. An example is the Mobile-edge Computation Offloading (MECO), which transfers time and hence, power intensive computations to clouds located at the edge of cellular networks [99]. Offloading the power intensive computations elsewhere forms the crux of the CS-based scheme discussed in this Chapter. The CS-based technique described in this work, in essence, can be viewed as an edge-computing solution. The authors in [100] discuss a complete CS-based IoT platform in which acquisition and reconstruction are CS-based and the CS-based reconstruction is implemented on the edge computing platform. However, we take a step further to propose that acquisition and decision for real-time tasks such as event-detection and tracking can be completely CS-based and be achieved on the same platform or on a edge computing platform. As discussed earlier, even reconstruction can be time-intensive and there are a wide range of problems that can be solved efficiently without reconstruction, purely from compressive measurements. Thus, we propose the use of compressed-sensed domain as an effective edge computing technique.

5.5 CS-based Reconstruction

Surveillance cameras transmit videos that are processed and analyzed at a processing station. As seen in Chapter 2, CS can reduce the number of samples to the information rate and ensure faithful reconstruction. With the encoding simplified to a multiplication of the sensing matrix and the data samples, the reconstruction at the ground station or the processing end becomes computationally intensive. Greedy algorithms, as discussed in Chapter 2 enable quicker computation. The algorithm for OMP is discussed in Section 2.3.3. The OMP, implemented with a pseudoinverse as a straightforward solution to the least squares, could be very inefficient and consume a lot of computational power. OMP works on greed by estimating the most correlated column of the measurement matrix with the residual at each iteration. It estimates the sparse signal of interest at every iteration (illustrated in the Algorithm under Section 2.3.3) with a closed-form solution by computing a pseudoinverse of the subset of the columns of the sensing matrix under the calculated support. This operation is typically computationally intensive.

5.5.1 Proposed Alternate Implementations for OMP

The proposed implementation achieves orthogonalization of every new column selected at the start of the iteration against the existing atoms by *modified Gram-Schmidt* (GS) method [101]. A similar technique with the modified GS-method is discussed in [102] for an efficient QR factorization. Though in principle, the Gram-Schmidt process can be used to solve the least squares problem as it yields a QR decomposition, it is numerically unstable [101]. Hence the modified Gram Schmidt method is preferred for the orthogonalization.

The steps of the OMP algorithm, as described in Section 2.3.3 are as follows:

Input:

$$A = \{\mathbf{a}_i\}_{i=1}^N \in \mathbb{R}^{M \times N} : \mathbf{y} \in \mathbb{R}^M: \text{Error threshold, } \varepsilon$$

Initialization:

$j = 0 : \mathbf{x}^0 = 0 : \text{Residual, } \mathbf{r}^0 = \mathbf{y} : \text{Support, } P^0 = \emptyset : V = \emptyset \in M \times M$

Steps:

While not converged do:

1. Iterate: $j = j + 1$
2. Choose i_0 such that $\min_c \|\mathbf{c}\mathbf{a}_{i_0} - \mathbf{r}^{j-1}\|_2 \leq \min_c \|\mathbf{c}\mathbf{a}_i - \mathbf{r}^{j-1}\|_2 \forall i$
 $\mathbf{b} = A^T \mathbf{r}^{j-1} : i_0 = \arg \max_k \{\mathbf{b}_k\}$
3. $P^j = P^{j-1} \cup i_0$
4. $\mathbf{x}^j = \arg \min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{y}\|_2$ subject to $\text{supp } \mathbf{x} = P^j$

Until: $\mathbf{r}^j \leq \varepsilon$

Output: $\mathbf{x}^j$

The ways to compute the least squares solution in step 4 in the simplest form of the OMP algorithm and the proposed improvements, “OMPFast” and “OMPRe” are described in Table 5.3.

Steps	OMP	OMPFast	OMPRe
Orthogonalization:		$k = 1 : \mathbf{v}_j = \mathbf{a}_j$ Do: $V \rightarrow \mathbf{v}_j = \mathbf{v}_j - \langle \mathbf{v}_j, \mathbf{v}_k \rangle \mathbf{v}_k$ $k = k + 1$: Until: $k < j$ $\mathbf{v}_j = \frac{\mathbf{v}_j}{\ \mathbf{v}_j\ _2}$	$k = 1 : \mathbf{v}_j = \mathbf{a}_j$ Do: $V \rightarrow \mathbf{v}_j = \mathbf{v}_j - \langle \mathbf{v}_j, \mathbf{v}_k \rangle \mathbf{v}_k$ $k = k + 1$: Until: $k < j$ $\mathbf{v}_j = \frac{\mathbf{v}_j}{\ \mathbf{v}_j\ _2}$ $k = 1$
(Re)orthogonalization:			Do: $V \rightarrow \mathbf{v}_j = \mathbf{v}_j - \langle \mathbf{v}_j, \mathbf{v}_k \rangle \mathbf{v}_k$ $k = k + 1$: Until: $k < j$ $\mathbf{v}_j = \frac{\mathbf{v}_j}{\ \mathbf{v}_j\ _2}$
Least Squares Solution	$\hat{\mathbf{x}}^j = A_{\in P}^\dagger \mathbf{y}$ $\mathbf{r}^j = \mathbf{y} - A_{\in P} \hat{\mathbf{x}}^j$	$\hat{\mathbf{x}}^j = V_{\in P}^T \mathbf{y}$ $\mathbf{r}^j = \mathbf{y} - A_{\in P} \hat{\mathbf{x}}^j$	$\hat{\mathbf{x}}^j = V_{\in P}^T \mathbf{y}$ $\mathbf{r}^j = \mathbf{y} - A_{\in P} \hat{\mathbf{x}}^j$

Table 5.3: Alternate proposed approaches for OMP implementation; $A^\dagger$ is the pseudoinverse of matrix A .

The inverse of the orthogonalized set of columns in a real matrix is essentially its transpose.

Claim: V is left-invertible and $V^{-1} = V^T$

This is substantiated by the Proof 5.5.1.

Proof 5.5.1 *At the end of an iteration k in the orthogonalization stage:*

Given: $V \in \mathbb{R}^{M \times z}$, $z \in [1, 2, \dots, M]$ and $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_z\}$ are orthonormal, as guaranteed by the modified GS.

Thus, $\{\mathbf{v}_z\}$ are linearly independent, $z \leq M$ and the rank of V is z . Hence, the left-inverse of V exists.

$$V^T V = \begin{bmatrix} \mathbf{v}_1^T \\ \mathbf{v}_2^T \\ \vdots \\ \mathbf{v}_z^T \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \dots & \mathbf{v}_z \end{bmatrix} = I_{z \times z}$$

Thus, $V^{-1} = V^T$. QED

The OMPFast algorithm is based entirely on the modified GS method. The vectors mutually orthogonalized by the modified GS might not be as close to orthonormal as required. In this scenario, the V obtained through orthonormalization could be input for a rerun of the orthonormalization. This is called the modified GS with “reorthogonalization” [101] and used to implement OMPRe in this work. Since V from the first run is nearly an isometry, the second run will modify the vectors to be even close to orthonormal.

5.5.2 Analysis of Mean Squared Error (MSE) across proposed methods

The mean squared error (MSE) is a function of the pseudo-inverse of the measurement matrix, as indicated by the least squares solution step in Table 5.3, $\hat{\mathbf{x}}^j = A_{\in P}^\dagger \mathbf{y}$. Thus, the final error in $\mathbf{x} - \hat{\mathbf{x}}$ is a reflection of the error in the calculation of the pseudo-inverse of A . The differentiator in the proposed methods is the technique in the calculation of the pseudo-inverse. To evaluate the numerical impact on the MSE, we consider a small perturbation in the measurement matrix, A . The authors in [103] consider the effect of perturbation in pseudo-inverses. For any A and B , with $B = A + E$, $\|B^\dagger - A^\dagger\| \leq \mu \max(\|A^\dagger\|_2^2, \|B^\dagger\|_2^2) \|E\|$ where μ depends on the norm

chosen. A theoretical performance analysis of the accumulation of the error in the pseudo-inverse calculation over multiple iterations as in OMP is not straightforward and hence, the error is analyzed by simulating a small perturbation in the form of AWGN in A and using that in the reconstruction. The error in the pseudo-inverse is calculated as:

$$\delta E = \left\| A_{\in P, \text{noisy}}^\dagger - A_{\in P, \text{noiseless OMP}}^\dagger \right\|_F \quad (5.2)$$

where $A_{\in P, \text{noisy}}^\dagger$ is the pseudo-inverse of A calculated in each implementation of the OMP for the noisy scenario and $\|\cdot\|_F$ is the Frobenius-norm of the matrix. The error in the pseudo-inverse of A is calculated by the accumulated error averaged over 100 iterations in each implementation of the OMP as compared to the noiseless scenario. Figure 5.4 shows the comparison of errors across the different implementations for a particular value of noise variance. The accumulated errors for the OMPFast and OMPRe are similar and coincide while that of the original implementation of OMP increase with the increase in number of measurements. Since OMPFast and OMPRe works by orthogonalization of a newly chosen column against the rest, this would attribute to the suppression of the noise itself up to a certain noise floor. The accumulated error is more or less similar for the initial few measurement values across all the implementations because the reconstruction error is high and the number of measurements are not sufficient for an accurate reconstruction of the signal. But, with $M > 70$, the error in the pseudo-inverse calculation increases for OMP while it reduces for the OMPFast and OMPRe.

Figure 5.5 shows the plot of the accumulated errors in the pseudo-inverse for $N = 256$, $M = 100$ and $m = 32$, across varying SNR. Again, the errors for the OMPFast and OMPRe are similar while for that of OMP is much higher. However, the error is largely insensitive to the magnitude of the noise across all implementations. Thus, the proposed implementations of OMPFast and OMPRe are more robust to perturbations in the measurement matrix, A .

Figure 5.4: Comparison of errors for various measurements in a 32-sparse Signal with $N = 256$ averaged over 100 trials

Figure 5.5: Comparison of errors across SNR for $N = 256$, $M = 100$, $m = 32$ averaged over 100 trials

5.5.3 Simulations and Results

The “OMPFast” algorithm was implemented by the author as part of an in-house Compressive Sensing Diagnostic tool at Airbus Group Innovations (AGI), Singapore. The OMPFast algorithm was used for sparse coding of a 2-D non-synthetic, non-stationary signal, and the times for computation are tabulated in Table 5.4, for both low-frequency and high-frequency data. The method, thus implemented enabled a computational time savings of 69 – 97% depending on the nature of the data.

Table 5.4: Comparison of Computational Times

Sample Data	OMP (sec)	OMPFast (sec)
Low Frequency	0.78	0.24
High Frequency	1841.3	46.8

Simulations are run for m -sparse signals with length of the signal, $N = 256$. Figure 5.6 shows the error convergence trends of OMP and OMPFast for m -sparse signals with varying measurements. The MSE in both cases are equal, which implies that the left-invertible semi-orthogonal matrix approximation agrees well with the conventional Moore-Penrose inverse of the matrix [104]. Figure 5.7 shows the error convergence for OMPRe too. Though this trend is not seen always, reorthogonalization can only strengthen the orthonormal columns and reduce the error, as seen with increasing m . The MSE curves for both the OMP and OMPFast coincide. However, as seen from Section 5.5.2 the OMPFast and OMPRe are more robust in the presence of noise or perturbations in the measurement matrix. The computational times for all the methods is shown in Figure 5.8. OMPFast is the most computationally efficient, while OMPRe with the additional orthogonalizations is still superior to the OMP, the gap in the computational times widening as sparsity, m increases.

Figure 5.6: Error Convergence for Various Measurements in a m -sparse Signal with $N = 256$ averaged over 100 trials

5.5.4 Summary

In this chapter we propose a computationally less intensive methodology for detection of targets on airborne platform such as drone. The detection is achieved using the compressed sensed features captured from the frames. The evaluation of the proposed methodology is done on a cubieboard. The result shows that the accuracy is at least 98.5% and is almost real-time. The computational complexity for onboard detection depends largely only on the encoding time of the ROI. The implementation of the detection on-board the drone not only reduces the encoding time required for the CS-based detection, but also reduces the transmission bandwidth that would otherwise be required to transmit every frame to the ground station for a decision to be

Figure 5.7: Error Convergence for Various Measurements in a m -sparse Signal with $N = 256$ averaged over 100 trials

taken. This kind of on-board detection facilitates quick decision that can be taken on-board and for further course of actions like tracking. Compressed-sensed domain analytics can be viewed as a potential edge computing technique. The more efficient implementations of OMP are discussed and it is shown that orthogonalization technique can improve the computational times a great deal, especially if the sparsity and dimension of the signal increases.

Figure 5.8: Computational Times for Various Measurements in a m -sparse Signal with $N = 256$ averaged over 100 trials

Chapter 6

Conclusions and Future Directions

The conclusions drawn from the works discussed in the thesis and the proposed future directions are outlined here.

6.1 Conclusions

Sparsity forms the core of the works discussed in the thesis. When signals are sparse in a different space/dimension or in a dictionary, solutions to signal processing are feasible by known sparsity. Sparse representation of the signal reveals that signal can be compressed. Recent trends of research focus on acquiring the sparse signal through compressed sensing. The thesis discussed various applications of sparse representation-based signal processing in airborne platforms, namely in accelerometer data collected from helicopters, SAR and on-board surveillance drones. The works discussed in the thesis are summarized below.

- Multiple approaches to dictionary training are proposed, namely the (i) l_∞ -norm minimization (ii) baseline retraining and (iii) a modified and enhanced improvement to the SGK algorithm, the Weighted SGK, which modifies the dictionary update step in the dictionary learning algorithm to better represent signals that do not fall in the more “common” set of the training data. Anomaly Detection in artificially generated data in

NAB and data obtained from accelerometers positioned on helicopters was achieved using the retraining methods. A scheme for anomaly detection with the features extracted from the dictionary-based sparse representation, namely the representation error and the sparsity of the data, is also proposed. It is shown that Weighted SGK based retraining enabled a better representation-basis for the signals thereby resulting in better classification with the proposed anomaly detection scheme. Comparison with other classification algorithms such as isolation Forest and GMM also show that Weighted SGK-based retraining gives a better performance for the accelerometer data.

- SAR as an application for the use of CS is considered, since data compression and related tasks in SAR always require more efficient and effective methods. Though sparsity-based reconstruction for raw SAR data has been studied in the last decade, there is still a need to formulate and represent sparsity in more effective ways. A CS-based LRSD method for compressing and reconstructing real raw SAR data is proposed. A measurement matrix is introduced into the variational approach to RPCA to enable the measurements to be compressed-sensed. CS-based LRSD achieves simultaneous decompression as well as decomposition. The quality of images of the decomposed components with subsequent SAR processing are analysed. It is shown that the metrics used to indicate visual aspects of the images gave reasonable results for the CS-based LRSD scheme, as compared to the uncompressed and the traditional CS-compressed data. The results show that for this image quality, CS-based LRSD scheme could achieve a 5% compression, which is significant savings for SAR downlink bandwidth. The CS-based LRSD method is also extended for speckle processing and results are presented for the real SAR data.
- CS-based video processing for intrusion-detection in surveillance camera was implemented entirely in the compressive-sensed domain or the measurement domain. The power consumption, computational complexity of the scheme and the detection accuracy are presented. The CS encoding times are shown to outperform the time used for a classical

JPEG encoding. A quicker and more efficient alternative to the implementation of the OMP algorithm in the CS-reconstruction is also discussed, which uses sequential orthogonalization of atoms to speed up the computation of the pseudoinverse. A comparison of MSE as a function of perturbation in the measurement matrix shows that the accumulated error in the pseudo-inverse is much more with the method OMP as compared to OMPFast and OMPRe.

6.2 Future Directions

The following are presented as promising future directions to the topics discussed in the thesis:

- The application of Weighted SGK-based retraining can be extended to other real data sets for one class classification. A normalized RMSE weights matrix is used for the relative weighting of the errors. Other weighting functions could be explored which might give a better performance. Adaptation of the sample-wise l_∞ -norm minimization to anomaly detection could prove to be useful and is a topic to explore further.
- CS-based LRSD could be incorporated in to the SAR processing so that the decomposition would directly yield the SAR images, and this can be applied to more types of SAR data. The nature of phase in the SAR signal and the potential of sparse sensing along the azimuth, to image the target scene, for point targets is promising.
- A future development in the task of intrusion detection could be potentially dynamically learning the ROI and the dynamic learning of the threshold to enable the detection. This would enable the airborne system to be completely autonomous. In the event of an intrusion-detection, on-board implementation of CS-based tracking would exploit the mobility of the drone and facilitate data-driven decisions. With the feasibility of on-board processing demonstrated, the algorithm could be extended to accommodate the motion effects of the drone. The ability of compressed-sensed domain processing in edge analytics and its integration on to the edge computing platform is promising.

Author's Publications

Journal Papers

[1]. A. Srinivasan, A. Makur and A. Vijayakumar, “Sparse Representation-based Dictionary Learning for One Class Classification Problems”, *In Preparation* for submission to Mechanical Systems and Signal Processing (MSSP) Journal, Elsevier.

Conference Papers

[1] A. Srinivasan, A. Vijayakumar and A.Makur, “Compressive Sensing-based On-board Intrusion Detection for a Surveillance Drone”, *IEEE VTS Asia Pacific Wireless Communications Symposium*, APWCS Aug 2019.

[2] A. Srinivasan, A. Makur and A. Vijayakumar, “Compressive Sensing (CS)-based Low Rank and Sparse Decomposition for Synthetic Aperture Radar (SAR)”, *In Preparation for Submission*

References

- [1] R. G. Baraniuk, “More Is Less: Signal Processing and the Data Deluge,” *Science*, vol. 331, no. 6018, pp. 717–719, Feb 2011.
- [2] S. Mallat, *A Wavelet Tour of Signal Processing*, 3rd ed. Elsevier, 2009.
- [3] C. Shannon, “A Mathematical Theory of Communication,” *The Bell System Technical Journal*, vol. 27, pp. 623–656, Oct 1948.
- [4] T. Strohmer, “Measure what should be measured: Progress and Challenges in Compressive Sensing,” *IEEE Signal Processing Letters*, vol. 19, no. 12, pp. 887–893, 2012.
- [5] S. Qaisar, R. M. Bilal, W. Iqbal, M. Naureen, and S. Lee, “Compressive Sensing: From Theory to Applications,” *IEEE Journal of Communications and Networks*, vol. 15, no. 5, pp. 443–456, Nov 2013.
- [6] Y. Chan and V. Koo, “An Introduction to Synthetic Aperture Radar (SAR),” *Progress in Electromagnetics Research B*, vol. 2, pp. 27–60, 2008.
- [7] S. K. Sahoo and A. Makur, “Dictionary Training for Sparse Representation as Generalization of K-means Clustering,” *IEEE Signal Processing Letters*, vol. 20, no. 6, Jun 2013.
- [8] 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, May 2015.

- [9] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. R. Bach, “Supervised Dictionary Learning,” in *Advances in Neural Information Processing Systems 21 (NIPS 2008)*. Vancouver, Canada, 2008.
- [10] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T.-W. Lee, and T. J. Sejnow, “Dictionary Learning Algorithms for Sparse Representation,” *Neural Computation*, vol. 15, no. 2, pp. 349–396, Feb 2003.
- [11] M. Aharon, M. Elad, and A. Bruckstein, “K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation,” *IEEE Trans. on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, Nov 2006.
- [12] Q. Zhang and B. Li, “Discriminative K-SVD for Dictionary Learning in Face Recognition,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*. CA, USA, 2010.
- [13] K. Engan, S. O. Aase, and J. H. Husy, “Multi-frame Compression: Theory and Design,” *Signal Process.*, vol. 80, no. 10, pp. 2121–2140, 2000.
- [14] S. Sahoo and A. Makur, “Image denoising via sparse representations over sequential generalization of k-means (sgk),” in *International Conference on Information Communication Signal Processing*. Taiwan, 2013.
- [15] D.-S. Pham and S. Venkatesh, “Joint Learning and Dictionary Construction for Pattern Recognition,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*. Anchorage, USA, 2008.
- [16] Z. Jiang, Z. Lin, and L. S. Davis, “Learning a Discriminative Dictionary for Sparse Coding via Label Consistent K-SVD,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 1697–1704.

- [17] T. Andrysiak and L. Saganowski, “Incoherent Dictionary Learning for Sparse Representation in Network Anomaly Detection,” *Schedae Informaticae*, vol. 24, 2015.
- [18] D. Donoho, “Compressed Sensing,” *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.
- [19] D. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, “Sparse Solution of Underdetermined Systems of Linear Equations by Stagewise Orthogonal Matching Pursuit,” *IEEE Transactions on Information Theory*, Feb 2012.
- [20] E. Candes and M. Wakin, “An Introduction To Compressive Sampling,” *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, Mar 2008.
- [21] M. Duarte and Y. Eldar, “Structured Compressed Sensing: From Theory to Applications,” *IEEE Trans. on Signal Processing*, vol. 59, no. 9, Sep 2011.
- [22] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, “A Simple Proof of the Restricted Isometry Property for Random Matrices,” *Constructive Approximation*, vol. 28, 2008.
- [23] R. A. DeVore, “Deterministic Constructions of Compressed Sensing Matrices,” *Journal of Complexity*, vol. 23, no. 4-6, pp. 918–925, Aug 2007.
- [24] E. Candes, T. Tao, and J. Romberg, “Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Frequency Information,” *IEEE Trans. Inform. Theory*, vol. 52, pp. 489–509, Aug 2006.
- [25] R. M. Gray, “Toeplitz and Circulant Matrices: A Review,” *Foundations and Trends in Communications and Information Theory*, vol. 2, no. 3, pp. 155–239, 2006.
- [26] A. C. Sankaranarayanan, M. A. Herman, P. Turaga, and K. F. Kelly, “Enhanced Compressive Imaging Using Model-Based Acquisition,” *IEEE Signal Processing Magazine*, vol. 33, no. 5, Sep 2016.

- [27] M.A.T.Figueiredo, R.D.Nowak, and S.J.Wright, “Gradient Projection for Sparse Reconstruction: Application to Compressed Sensing and Other Inverse Problems,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 586–597, Dec 2007.
- [28] I. Daubechies, M. Defrise, and C. D. Mol, “An Iterative Thresholding Algorithm for Linear Inverse Problems with a Sparsity Constraint,” *Comm. Pure and Appl. Math.*, vol. 57, pp. 1413–1457,, 2004.
- [29] S. J. Wright and R. Nowak, “Sparse Reconstruction by Separable Approximation ,” *IEEE Trans. on Signal Processing*, vol. 57, no. 7, Jul. 2009.
- [30] Y. Pati, R. Rezaifar, and P. Krishnaprasad, “Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition,” in *Proc. of 27th Asilomar Conference on Signals, Systems and Computers*. Pacific Grove, CA, USA, Nov, 1993, pp. 40–44.
- [31] D. Needell and R. Vershynin, “Signal Recovery from Incomplete and Inaccurate Measurements via Regularized Orthogonal Matching Pursuit,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 310–316, Feb 2010.
- [32] R. Chalapathy and S. Chawla, “Deep learning for Anomaly Detection: A Survey,” arXiv preprint arXiv:1901.03407.
- [33] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly Detection: A Survey,” *ACM Computing Surveys*, vol. 41, no. 3, Jul 2009.
- [34] S. S. Khan and M. G. Madden, “A Survey of Recent Trends in One Class Classification,” in *AICS 2009: Artificial Intelligence and Cognitive Science*. Dublin, Ireland, 2009.
- [35] B. Krawczyk and B. Cyganek, “Selecting locally specialised classifiers for one-class classification ensembles,” *Pattern Analysis and Applications*, vol. 20, pp. 427–439, 2017.

- [36] A. Adler, M. E. and Yacov Hel-Or, and E. Rivlin, “Sparse Coding with Anomaly Detection,” *Journal of Signal Processing Systems*, vol. 79, pp. 179–188, 2015.
- [37] D. Carrera, G. Boracchi, A. Foi, and B. Wohlberg, “Detecting Anomalous Structures by Convolutional Sparse Models,” in *In Proc. of 2015 International Joint Conference on Neural Networks (IJCNN)*. Killarney, Ireland, 2015.
- [38] B. Pilastre, L. Boussouf, S. d. Escrivan, and J.-Y. Tourneret, “Anomaly detection in mixed telemetry data using a sparse representation and dictionary learning,” *Signal Processing*, vol. 168, 2020.
- [39] A. C. Earle, “Minimum l_∞ Norm Solutions to Finite Dimensional Algebraic Underdetermined Linear Systems,” <http://hdl.handle.net/10539/16858>, Feb 2015, A Dissertation.
- [40] G. A. Watson, “Approximation in normed linear spaces,” *Journal of Computational and Applied Mathematics*, vol. 121, no. 1, pp. 1–36, 2000.
- [41] E. W. Cheney, *Introduction to Approximation Theory*. AMS Chelsea Publication Series, 1982.
- [42] A. Lavin and S. Ahmad, “Evaluating real-time anomaly detection algorithms – the numenta anomaly benchmark,” *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, Dec 2015. [Online]. Available: <http://dx.doi.org/10.1109/ICMLA.2015.141>
- [43] “Airbus data challenge,” <https://www.airbus.com/innovation/tech-challenges-and-competitions/aigym.html>, 2019.
- [44] M. Braei and S. Wagner, “Anomaly detection in univariate time-series: A survey on the state-of-the-art,” arXiv preprint arXiv:2004.00433, 2020.
- [45] L. Li, R. J. Hansman, R. Palacios, and R. Welsch, “Anomaly detection via a gaussian mixture model for flight operation and safety monitoring,” *Transportation*

- Research Part C: Emerging Technologies*, vol. 64, pp. 45–57, 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0968090X16000188>
- [46] W. Zhao, L. Li, S. Alam, and Y. Wang, “An incremental clustering method for anomaly detection in flight data,” arXiv preprint arXiv:2005.09874, 2020.
 - [47] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
 - [48] “Sandia national laboratories,” http://www.sandia.gov/radar/what_is_/index.html.
 - [49] A. Moreira, P. Prats-Iraola, M. Younis, G. Krieger, I. Hajnsek, and K. P. Papathan, “A Tutorial on Synthetic Aperture Radar,” *IEEE Geoscience and Remote Sensing Magazine*, vol. 2, Mar 2013.
 - [50] M. Soumekh, *Synthetic Aperture Radar Signal Processing*. John Wiley and Sons, Inc, 1999.
 - [51] A. E. Boustani, K. Brunham, and W. Kinsner, “A Review of Current Raw SAR Data Compression Techniques,” in *In Proc. Canadian Conference on Electrical and Computer Engineering 2001*. Toronto, Canada, 2001.
 - [52] M. Herman and T. Strohmer, “High-Resolution Radar via Compressed Sensing,” *IEEE Transactions on Signal Processing*, vol. 57, no. 6, Jun 2009.
 - [53] D. H. V.M. Patel, G.R. Easley and R. Chellappa, “Compressed Synthetic Aperture Radar,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, Apr 2010.
 - [54] J. S.J. Wei, X.L. Zhang, “Sparse Reconstruction for SAR Imaging based on Compressed Sensing,” *Progress in Electromagnetics Research*, vol. 109, no. 2, pp. 63–81, 2010.

- [55] W. Min, “High Resolution radar Imaging based on Compressed Sensing and Adaptive l_p -norm Algorithm,” in *Proc. of 2011 IEEE CIE International Conference on Radar*. USA, 2001.
- [56] G. S.I. Kelly, C.Du and M. Davies, “Advanced Image Formation and Processing of Partial Synthetic Aperture Radar Data,” *IET Signal Processing*, 2011.
- [57] M. C. S. Samadi and M. Masnadi-Shirazi, “Sparse representation-based SAR imaging,” *IET Radar, Sonar and Navigation*, 2011.
- [58] S. Ugur and O. Arikan, “SAR Image Reconstruction and Autofocus by Compressed Sensing,” *Elsevier Digital Signal Processing*, vol. 22, no. 6, pp. 923–932, Dec 2012.
- [59] W. Wang, B. Zhang, J. Mu, and X. Wu, “Sparsity and Compressive Sensing for SAR Signal,” in *2012 IEEE Globecom Workshops*. Anaheim, CA, USA, 2012.
- [60] M. Cetin, I. Stojanovic, N. O. Onhon, K. Varshney, S. Samadi, W. C. Karl, and A. S. Willsky, “Sparsity-driven SAR Imaging,” *IEEE Signal Processing Magazine*, Jul 2014.
- [61] A. Sognali and M. Cetin, “Low-rank Sparse Matrix Decomposition for Sparsity-driven SAR Image Reconstruction,” in *2015 3rd International Workshop on Compressed Sensing Theory and its Applications to Radar, Sonar and Remote Sensing (CoSeRa)*. Pisa, Italy, 2015.
- [62] F. Bondi, “Low Rank Plus Sparse Decomposition of SAR data for Maritime Surveillance,” in *2016 4th International Workshop on Compressed Sensing Theory and its Applications to Radar, Sonar and Remote Sensing (CoSeRa)*. Aachen, Germany, 2016.
- [63] G. Li, Y. Ma, and J. Dong, “Total Variation Regularization-based Compressed Sensing Synthetic Aperture Radar Imaging,” *Journal of Applied Remote Sensing*, Dec 2018.
- [64] “Earthdata: Nasa,” <https://asf.alaska.edu/data-sets/sar-data-sets/radarsat-1/>.

- [65] D. Yang, G. Liao, S. Zhu, X. Yang, and X. Zhang, “SAR Imaging with Undersampled Data via Matrix Completion,” *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 9, Sep 2014.
- [66] K. Dai, W. Lyu, S. Luo, and Q. Shi, “Bayesian Sparse Representation Model for SAR Image Classification,” *Journal of Applied Remote Sensing*, vol. 13, no. 4, Dec 2019.
- [67] Z. He, H. Xiao, and R. Liu, “Synthetic Aperture Radar Target Recognition Based on Multidimensional Sparse Model,” *Jou. of Physics: Conference Series*, vol. 1169, 2019.
- [68] M. Liu and S. Chen, “Label-dependent Sparse Representation for Synthetic Aperture Radar Target Configuration Recognitions,” *International Journal of Remote Sensing*, vol. 38, no. 17, pp. 4868–4887, May 2017.
- [69] L. Jin, J. Chen, and X. Peng, “Synthetic Aperture Radar Target Classification via Joint Sparse Representation of Multi-level Dominant Scattering Images,” *Optik, Elsevier*, vol. 186, no. 4, pp. 110–119, Jun 2019.
- [70] W. Li, J. Yang, and Y. Ma, “Target Recognition of Synthetic Aperture Radar Images Based on Two-Phase Sparse Representation,” *Journal of Sensors*, Feb 2020.
- [71] “The sensor data management system,” <https://www.sdms.afrl.af.mil/index.php?collection=mstar>.
- [72] E. Candes, X. Li, Y. Ma, and J. Wright, “Robust Principal Component Analysis,” *Journal of the ACM*, vol. 58, no. 3, May 2011.
- [73] A. E. Waters, A. C. Sankaranarayanan, and R. G. Baraniuk, “SpaRCS: Recovering Low-Rank and Sparse Matrices from Compressive Measurements,” in *Advances in Neural Information Processing Systems 24 (NIPS 2011)*. Honolulu, Hawaii, 2010, pp. 1089–1097.

- [74] A. Aravkin, S. Becker, V. Cevher, and P. Olsen, “A Variational Approach to Stable Principal Component Pursuit,” in *UAI’14: Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*. Quebec, Canada, 2014, pp. 32–41.
- [75] B. Recht, M. Fazel, and P. A. Parrilo, “Guaranteed Minimum Rank Solutions of Linear Matrix Equations via Nuclear Norm Minimization,” *SIAM Rev.*, vol. 52, no. 3, pp. 471–501, 2010.
- [76] T. T. Do, L. Gan, N. H. Nguyen, and T. D. Tran, “Fast and Efficient Compressive Sensing using Structurally Random Matrices,” *IEEE Transactions on Signal Processing*, vol. 60, no. 1, pp. 139–154, Jan 2012.
- [77] D. Long, M. E. E. Zaugg, and J. Maslanik, “The MicroASAR Experiment on CASIE-09,” in *Proc. of the International Geoscience and Remote Sensing Symposium*. Honolulu, Hawaii, 2010, pp. 3466–3469.
- [78] D. Long and C. Stringham, “The sample byu casie-09 microsar dataset,” http://www.mers.byu.edu/microASAR/CASIE_sample/.
- [79] C. Pieterse, “Metrics to evaluate Compression Algorithms in Raw SAR Data,” https://repository.up.ac.za/bitstream/handle/2263/72668/Pieterse_Metrics_2019.pdf, Apr 2019, Report.
- [80] T. D. Penrod and G. G. Kuperman, “Image Quality Analysis of Compressed Synthetic Aperture Radar Imagery,” U.S. Air force, Tech. Rep., 1993.
- [81] K. Matkovic, L. Neumann, A. Neumann, T. Psik, and W. Purgathofer, “Global Contrast Factor - a New Approach to Image Contrast,” in *in Proceedings of the First Euro Graphics Conference on Computational Aesthetics in Graphics, Visualization and Imaging*. Girona, Spain, 2005.

- [82] J.-S. Lee, "Speckle Suppression And Analysis For Synthetic Aperture Radar Images," *Optical Engineering*, vol. 25, no. 5, pp. 636 – 643, 1986. [Online]. Available: <https://doi.org/10.1117/12.7973877>
- [83] J. S. Lee, L. Jurkevich, P. Dewaele, P. Wambacq, and A. Oosterlinck, "Speckle filtering of synthetic aperture radar images: A review," *Remote Sensing Reviews*, vol. 8, no. 4, pp. 313–340, 1994. [Online]. Available: <https://doi.org/10.1080/02757259409532206>
- [84] Y. Zhang, K. Ji, Z. Deng, S. Zhou, and H. Zou, "Clustering-based sar image denoising by sparse representation with ksvd," in *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2016, pp. 5003–5006.
- [85] L. Shuaiqi, Q. Hu, P. Li, J. Zhao, C. Wang, and Z. Zhu, "Speckle suppression based on sparse representation with non-local priors," *Remote Sensing*, vol. 10, p. 439, 03 2018.
- [86] H. Zhou, L. Chen, B. Fu, and H. Shi, "Sar image denoising method based on sparse representation," *The Journal of Engineering*, vol. 2019, no. 20, pp. 7153–7156, 2019.
- [87] J.-X. Wang, "Research and Implementation of Intrusion Detection Algorithm in Video Surveillance," in *International Conference on Audio, Language and Image Processing (ICALIP)*, 2016.
- [88] R. A. Shahad, L. G. Bein, M. H. M. Saad, and A. Hussain, "Complex Event Detection in an Intelligent Surveillance System using CAISER Platform," in *2016 International Conference on Advances in Electrical, Electronic and Systems Engineering (ICAEES)*. Putrajaya, Malaysia, 2016.
- [89] D.Arjun, P. Indhukala, and K. U. Menon, "Border Surveillance and Intruder Detection using Wireless Sensor Networks: A Brief Survey," in *International Conference on Communication and Signal Processing (ICCSP)*, 2017.

- [90] “Autonomous Intrusion Detection Drone,” <https://i-hls.com/archives/67503>, Israel Homeland Security (iHLS).
- [91] V. Cevher, A. Sankaranarayanan, M. F. Duarte, D. Reddy, R. G. Baraniuk, and R. Chellappa, “Compressive Sensing for Background Subtraction,” in *ECCV '08 Proceedings of the 10th European Conference on Computer Vision: Part II*. Girona, Spain, 2008, pp. 155–168.
- [92] M. Davenport, P. Boufounos, M. Wakin, and R. Baraniuk, “Signal Processing with Compressive Measurements,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 445–460, Apr 2010.
- [93] P. Bristeau, F. Callou, D. Vissiere, and N. Petit, “The Navigation and Control technology inside the AR. Drone micro UAV,” in *18th IFAC World Congress*, vol. 18, 2011, pp. 1477–1484.
- [94] K. Shetti and A. Vijayakumar, “Evaluation of Compressive Sensing encoding on AR Drone,” in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, 2015.
- [95] M. Davenport, P. Boufounos, M. Wakin, and R. Baraniuk, “Signal Processing with Compressive Measurements,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 445–460, Apr 2010.
- [96] T. L. N. Nguyen and Y. Shin, “Deterministic Sensing Matrices in Compressive Sensing,” *The Scientific World Journal*, 2013.
- [97] J. Ribera, “Jpeg encoder,” <https://github.com/javiribera/jpeg-encoder>, Github Repository.
- [98] W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, “Edge Computing: A Survey,” *Future Generation Computer Systems*, vol. 97, no. 4, pp. 219–235, Aug 2019.

- [99] M. Patel, Y. Hu, P. Hede, J. Joubert, C. Thornton, B. Naughton, J. R. Ramos, C. Chan, V. Young, S. J. Tan, and D. Lynch, *Mobile-Edge Computing Introductory Technical White Paper*. Huawei, IBM, Intel, Nokia Networks, NTT DOCOMO, Vodafone, September 2014.
- [100] H. Djelouat, A. Amira, and F. Bensaali, “Compressive Sensing-Based IoT Applications: A Review,” *Journal of Sensor and Actuator Networks*, vol. 7, no. 4, 2018.
- [101] D. S. Watkins, *Fundamentals of Matrix Computations*, 3rd ed. Wiley, Jul 2010.
- [102] T. Blumensath and M. Davies, “On the Difference Between Orthogonal Matching Pursuit and Orthogonal Least Squares,” 2007, unpublished.
- [103] G. Stewart, “On The Perturbation OF Pseudo-inverses, Projections And Linear Least Squares Problems,” *SIAM Review*, vol. 19, no. 4, Oct 1977.
- [104] C. R. Rao and S. K. Mitra, *Generalized Inverse of Matrices and Its Applications*. Wiley, 1971.