Sparse signal processing and compressed sensing recovery

Sujit Kumar Sahoo

2013

Sujit Kumar Sahoo. (2013). Sparse signal processing and compressed sensing recovery.
Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/61795>

<https://doi.org/10.32657/10356/61795>

Sparse Signal Processing and Compressed Sensing Recovery

Sujit Kumar Sahoo

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

2013

Acknowledgments

It is my pleasure to thank all the people whom I am grateful to, for all their help during the course of this journey.

First and foremost, I would like to express my most sincere gratitude to my advisor, Prof. Anamitra Makur, for his continuous support, guidance and encouragement. It is his encouragement and timely help that led to the completion of this thesis.

I am also grateful to the School of EEE for their generous financial support and for providing excellent laboratory facilities. The invaluable administrative help by Ms. Leow of Media Technology Laboratory, which made life so easy, is greatly acknowledged. I would also like to extend this acknowledgment to Mr. Mui and Ms. Hoay for their administrative help during my stay in the Information Systems Research Laboratory. I would also like to acknowledge my ex-supervisors, the ex-faculties of NTU, Prof. Bogdan J. Falkowski and Prof. Lu Wenmiao. It was purely pragmatic to start my research journey with their guidance.

I would like to acknowledge M. Aharon and M. Elad for making the code available online, which made it easier for us to reproduce the results of Chapter 3 and 4. I would also like to acknowledge Morphological Component Analysis group (J. Fadili, J. L. Starck, M. Elad, and D. Donoho) for reproducible research, their inpainting results were illustrated in Chapter 5. I would also like to thank P. Chatterjee and P. Milanfar for making their code available, their denoising results were illustrated in Chapter 5.

I am very much thankful to my team-mates and friends, Jayachandra, Anil, Vinod, Sathya, Huang Honglin, Divya,....the list goes on, who helped me in one way or the other during the course of my studies. My special thanks to Arun, Dileep, Hateesh and Prince who made my stay in Singapore joyous and a most memorable one.

I am very lucky to have a wonderful parents, sister and brother-in-law, who always provide me with loads of encouragement and support. The arrival of my super charged,

ever smiling niece has brought lot of happiness and lifted all our spirits to a totally different level. A few minutes of just listening to her various sounds over the phone is enough to be delighted. It is extremely difficult to even imagine this work without all their support. I am truly grateful to them. My loving grandparents are my mentors. It is very difficult to put in words my gratitude to them. I dedicate this thesis to the memories of my loving grandparents, and the Almighty.

Contents

Acknowledgments	i
Summary	vii
List of Figures	ix
List of Tables	xi
List of Notations	xii
1 Introduction	1
1.1 Sparsity	2
1.2 Dictionary	3
1.3 Application of Sparsity	3
1.4 Compressed Sensing	4
1.5 Contributions of the Thesis	6
1.6 Organization of the Thesis	7
2 Literature Review	9
2.1 Dictionary	9
2.1.1 Method of Optimal Directions (MOD)	10
2.1.2 Union of Orthonormal Bases (UOB)	10
2.1.3 Generalized Principal Component Analysis (GPCA)	11
2.1.4 K -SVD	12
2.2 Sparse Coding	12
2.2.1 Orthogonal Matching Pursuit (OMP)	13
2.2.2 Basis Pursuit (BP)	14
2.2.3 FOCUSS	14
2.3 Image Recovery Problems	15

2.3.1	Inpainting	15
2.3.2	Denoising	16
2.4	Compressed Sensing Recovery	17
2.5	Summary	19
3	Dictionary Training	21
3.1	K -means Clustering for VQ	22
3.2	K -SVD	24
3.2.1	K -means and K -SVD	25
3.3	MOD	26
3.3.1	K -means and MOD	27
3.4	A Sequential Generalization of K -means	27
3.4.1	K -means and SGK	28
3.5	Complexity Analysis	29
3.5.1	K -SVD	29
3.5.2	Approximate K -SVD	30
3.5.3	MOD	30
3.5.4	SGK	31
3.5.5	Comparison	31
3.6	Synthetic Experiment	31
3.6.1	Training Signal Generation	32
3.6.2	Dictionary Design	32
3.6.3	Results	32
3.7	Discussions	36
3.8	Summary	36
4	Applications of Trained Dictionary	37
4.1	Image Compression	38
4.1.1	Compression Experiments	39
4.2	Image Inpainting	44
4.2.1	Inpainting Experiments	44
4.3	Image Denoising	47

4.3.1	Dictionary Training on Noisy Images	49
4.3.2	Denoising Experiments	50
4.4	Discussions	52
4.5	Summary	57
5	Improving Image Recovery by Local Block Size Selection	58
5.1	Local Block Size Selection	59
5.2	Inpainting using Local Sparse Representation	60
5.2.1	Block Size Selection for Inpainting	61
5.2.2	Implementation Details	62
5.3	Denoising Using Local Sparse Representation	64
5.3.1	Local Block Size Selection for Denoising	65
5.3.2	Implementation Details	66
5.4	Experimental Results	69
5.4.1	Inpainting	69
5.4.2	Denoising	71
5.5	Discussions	75
5.6	Summary	75
6	Extended Orthogonal Matching Pursuit	77
6.1	OMP for CS Recovery	78
6.2	Extended OMP for CS Recovery	80
6.3	Analysis of OMP_α	82
6.3.1	Admissible Measurement Matrix	83
6.3.2	Probability of Success	84
6.3.3	Main Result	85
6.3.4	OMP as a Special Case	88
6.4	Practical OMP_α	89
6.4.1	OMP_α without Prior Knowledge of Sparsity (OMP_∞)	91
6.5	Experiments	94
6.6	Discussions	96
6.7	Summary	97

7 Summary and Future Work	98
7.1 Summary	98
7.2 Future Work	100
Appendix	102
Author's Publications	104
References	106

Summary

The works presented in this thesis focus on sparsity in the real world signals, its applications in image processing, and recovery of sparse signal from Compressed Sensing (CS) measurements. In the field of signal processing, there exist various measures to analyze and represent the signal to get a meaningful outcome. Sparse representation of the signal is a relatively new measure, and the applications based on it are intuitive and promising. Overcomplete and signal dependant representations are modern trends in signal processing, which helps sparsifying the redundant information in the representation domain (dictionary). Hence, the goal of signal dependant representation is to train a dictionary from sample signals. Interestingly, recent dictionary training algorithms such as K -SVD, MOD, and their variations are reminiscent of the well know K -means clustering. The first part of the work analyses such algorithms from the viewpoint of K -means. The analysis shows that though K -SVD is sequential like K -means, it fails to simplify to K -means by destroying the structure in the sparse coefficients. In contrast, MOD can be viewed as a parallel generalization of K -means, which simplifies to K -means without affecting the sparse coefficients. Keeping stability and memory usage in mind, an alternative to MOD is proposed: a Sequential Generalization of K -means (SGK). Through the synthetic data experiment, the performance of SGK is demonstrated to be comparable with K -SVD and MOD. Using complexity analysis, SGK is shown to be much faster compared to K -SVD, which is also validated from the experiment. The next part of the work illustrates the applications of trained dictionary in image processing, where it compares the usability of SGK and K -SVD through image compression and image recovery (inpainting, denoising). The obtained results suggest that K -SVD can be successfully replaced with SGK, due to its quicker execution and comparable outcomes. Similarly, it is possible to extend the use of SGK to other applications of sparse representation. The subsequent part of

the work proposes a framework to improve the image recovery performance using sparse representation of local image blocks. An adaptive blocksize selection procedure for local sparse representation is proposed, which improves the global recovery of underlying image. Ideally, the adaptive blocksize selection should minimize the Mean Square Error (MSE) in a recovered image. The results obtained using the proposed framework are comparable to the recently proposed image recovery techniques. The succeeding part of the work addresses the recovery of sparse signals from CS measurements. The objective is to recover the large dimension sparse signals from small number of random measurements. Orthogonal Matching Pursuit (OMP) and Basis Pursuit (BP) are two well known sparse signal recovery algorithms. To recover a d -dimensional m -sparse signal, BP only needs the number of measurements $N = O\left(m \ln \frac{d}{m}\right)$, which is similar to theoretical ℓ_0 norm recovery. On the contrary, the best known theoretical guarantee for a successful signal recovery in probability shows OMP is needing $N = O(m \ln d)$, which is more than BP. However, OMP is known for its swift execution speed, and it's considered to be the mother of all greedy pursuit techniques. In this piece of the work, an improved theoretical recovery guarantee for OMP is obtained. A new scheme called OMP_α is introduced for CS recovery, which runs OMP for $m + \lfloor \alpha m \rfloor$ iterations, where $\alpha \in [0, 1]$. It is analytically shown that OMP_α recovers a d -dimensional m -sparse signal with high probability when $N = O\left(m \ln \frac{d}{\lfloor \alpha m \rfloor + 1}\right)$, which is a similar trend as that of BP.

List of Figures

2.1	OMP for CS Recovery	18
3.1	Dictionary training algorithm for sparse representation, the superscript $(.)^{(t)}$ denotes the matrices and the vectors at iteration number t	21
3.2	Average number of atoms retrieved after each iteration for different values of m at SNR = ∞ dB	33
3.3	Average number of atoms retrieved after each iteration for different values of m at SNR = 30 dB	34
3.4	Average number of atoms retrieved after each iteration for different values of m at SNR = 20 dB	34
3.5	Average number of atoms retrieved after each iteration for different values of m at SNR = 10 dB	35
4.1	The dictionaries of atom size 8×8 trained on the 19 sample images, starting with overcomplete DCT as initial dictionary.	40
4.2	Visual comparison of compression results of sample images.	42
4.3	Compression results: rate-distortion plot.	43
4.4	The corrupted image (where the missing pixels are blackened), and the reconstruction results using overcomplete DCT dictionary, K -SVD trained dictionary, and SGK trained dictionary, respectively. The first row is for 50% missing pixels, and the second row is for 70% missing pixels.	46
4.5	Image denoising using a dictionary trained on the noisy image blocks. The experimental results are obtained with $J = 10$, $\lambda = 30/\sigma$, $\epsilon^2 = n(1.15\sigma)^2$, and OMP.	51

4.6	The dictionaries trained on Barbara image at $\sigma = 20$ —initial dictionary, K -SVD trained dictionary, and SGK trained dictionary.	53
4.7	The denoising results for the Barbara image at $\sigma = 20$ —the original, the noisy, and restoration results using the two trained dictionaries.	54
5.1	Block schematic diagram of the proposed image inpainting framework. . .	62
5.2	Illustration of the block size selection for inpainting.	63
5.3	Flowchart of the proposed image denoising framework.	65
5.4	Illustration of clustering based on window selection for AWGN of various σ . . .	67
5.5	Visual comparison of inpainting performance across the methods.	70
5.6	Visual comparison of the denoising performances for AWGN ($\sigma = 25$). . .	73
5.7	Visual inspection at irregularities	74
6.1	The percentage of signal recovered in 1000 trials with increasing α , for various m -sparse signals in dimension $K = 1024$, from their $d = 256$ random measurements.	82
6.2	(A) The percentage of input signals of dimension $K = 256$ exactly recovered as a function of numbers of measurements (d) for different sparsity level (m). (B) The minimum number of measurements d required to recover any m -sparse signal of dimension $K = 256$ at least 95% of the time.	93
6.3	The minimum number of measurements (d) required to recover an m -sparse signal of dimension $K = 1024$ at least 95% of the time.	96

List of Tables

3.1	comparison of execution time (in millisecond)	32
3.2	Average no. of atoms retrieved by dictionary training	33
4.1	Comparison of execution time in seconds for one iteration of dictionary update (Compression). Boldface is used for the better result.	41
4.2	Comparison of execution time in seconds for one iteration of dictionary update (Inpainting). Boldface is used for the better result.	45
4.3	Comparison of average PSNR of the reconstructed test images in dB, at various percentage of missing pixel. Boldface is used for the better result.	45
4.4	Comparison of the denoising PSNR results in dB. In each cell two denoising results are reported. Left: using K -SVD trained dictionary. Right: using SGK trained dictionary. All numbers are an average over five trials. The last two columns present the average result and their standard deviation over all images. Boldface is used for the better result.	55
4.5	Comparison of execution time in seconds. Left: K -SVD training time. Right: SGK training time. Boldface is used for the better result.	56
5.1	Image inpainting performance comparison in PSNR	71
5.2	Image denoising performance comparison in PSNR	74
6.1	Linear Fitting of Fig. 6.2(B)	94
6.2	Linear Fitting of $C_0 m \ln \frac{K}{\alpha m + 1} + C_6$ in Fig. 6.3	95

List of Notations

Common Notations

$\langle, \rangle$	Inner product of two vectors of equal dimension
$ \cdot $	Cardinality of a set or the number of elements in a set, or absolute value of a scalar
$\ \cdot\ _0$	Number of non zero entries in a vector
$(\cdot)^T$	Transpose of a matrix
$O(\cdot)$	Order of a variable
σ	AWGN standard deviation
$\mathbb{R}$	Set of Real numbers
B	Binary mask of image size
$C_1, C_2, C_3, \dots$	Positive constants
$c_1, c_2, c_3, \dots$	Positive constants
$\mathbf{D} \in \mathbb{R}^{n \times K}$	Dictionary consisting of prototype signal atoms
$\mathbf{d} \in \mathbb{R}^n$	Signal atoms, or column vectors of $\mathbf{D}$
K	Number of atoms in a dictionary, or length of $\mathbf{s}$
k	Atom index
m	Sparsity or the number of nonzero entries in $\mathbf{s}$
n	Length of $\mathbf{x}$
$\mathbf{s} \in \mathbb{R}^K$	The sparse signal, or the sparse representation vector
$\hat{\mathbf{s}}$	Estimated sparse representation
t	Iteration / time instance

V	Additive noise of image size
X	Original image, or a non-corrupted image
$\hat{X}$	Recovered image
$\mathbf{x} \in \mathbb{R}^n$	Signal vector
$\hat{\mathbf{x}} = \mathbf{D}\hat{\mathbf{s}}$	Recovered local signal
Y	Corrupted image

Chapter 3: Dictionary Training

$(.)^{(t)}$	Time instance
$Q(.)$	Additional <i>structure/constraint</i> for sparse coding
$\mathcal{T}_{(.)}$	Computational complexity
a, b	Power of K for order comparison
$\mathbf{d}_k$	k^{th} dictionary atom
$\mathbf{E} \in \mathbb{R}^{n \times N}$	Representation error matrix
$\mathbf{E}_k \in \mathbb{R}^{n \times N}$	Representation error without the support of $\mathbf{d}_k$
e_k	Trivial basis having all 0 entries except 1 in the k^{th} position
i	Index of the signals $\mathbf{x}$ and sparse vectors $\mathbf{s}$
N	Number of training samples
R_k	The set of signal index using $\mathbf{d}_k$ for representation. It also denote the clusters in K -means
$\mathbf{S} \in \mathbb{R}^{K \times N}$	Matrix consisting of sparse representation vectors $\mathbf{s}_i$
$\mathbf{S}_k \in \mathbb{R}^N$	k^{th} row of $\mathbf{S}$
$\mathbf{s}_i$	Sparse representation vector of $\mathbf{x}_i$
$\mathbf{X} \in \mathbb{R}^{n \times N}$	Matrix consisting of signal vectors
$\mathbf{X}_k \in \mathbb{R}^{n \times R_k }$	Sub Matrix of signals indexed from R_k
$\mathbf{x}_i$	i^{th} training signal vector

Chapter 4: Applications of Trained Dictionary

λ	Lagrange multiplier for Global optimization
μ	Lagrange multiplier for local optimization
μ_{ij}	Lagrange multiplier for corresponding location (i, j)
C	Noise gain for sparse coding
$\hat{\mathbf{D}}$	Updated dictionary
J	Number of dictionary update iterations
(i, j)	2-D coordinates
$\mathbf{R}_{ij}$	The operator to extract a $\sqrt{n} \times \sqrt{n}$ size local patch from coordinate (i, j) of $\mathbf{X}$ and store as a $1 \times n$ column vector
$\mathbf{s}_{ij}$	Sparse vector to represent a patch extracted from coordinate (i, j)
$\hat{\mathbf{s}}_{ij}$	Recovered sparse vector to represent a patch extracted from coordinate (i, j)

Chapter 5: Improving Image Recovery By Local Block Size Selection

$\mathbf{b}_{ij}^n$	The binary mask in the occluded patch $\mathbf{y}_{ij}^n$
$\mathbf{D}_n$	Dictionary of signal prototypes, and dimension n is a variable
(i, j)	2-D coordinates
N	Total number of pixels in the image
$\mathbf{R}_{ij}^n$	The operator to extract a $\sqrt{n} \times \sqrt{n}$ size local patch from coordinate (i, j) of $\mathbf{X}$ and store as a $1 \times n$ column vector, where signal size n is a variable
$\mathbf{s}_{ij}^n$	Sparse representation of $\mathbf{x}_{ij}^n$ in $\mathbf{D}_n$
$\hat{\mathbf{s}}_{ij}^n$	Estimated sparse representation
$\mathbf{v}_{ij}^n$	The additive noise in the noisy patch $\mathbf{y}_{ij}^n$
$\mathbf{x}_{ij}^n = \mathbf{R}_{ij}^n X$	Columnized form of a patch extracted by a moving window of size $\sqrt{n} \times \sqrt{n}$ from X at the coordinate (i, j)

$\hat{\mathbf{x}}_{ij}^n = \mathbf{D}_n \hat{\mathbf{s}}_{ij}^n$	Estimation of $\mathbf{x}_{ij}^n$
$\mathbf{y}_{ij}^n$	Columnized form of corrupted version of the patch extracted from Y

Chapter 6: Extended Orthogonal Matching Pursuit

$\ \cdot\ _\infty$	Infinite norm of a vector or the maximum absolute entry in the vector
$\sigma(\cdot)$	The singular values of a matrix
$\mathbb{P}(\cdot)$	Probability of an event
$\mathbf{R}(\cdot)$	The range space or the column space of a matrix
α	OMP over run factor
δ	Restricted Isometry Property (RIP) constant
$\Phi \in \mathbb{R}^{d \times K}$	Measurement matrix
$\Phi_I \in \mathbb{R}^{d \times I }$	The matrix consists of the columns of Φ with indices $i \in I$
d	Number of linear projections
E_{fail}	Event consists of all possible instances of failure
E_{succ}	Event consists of all possible instances of success
I	Indices subset $I \subset \{1, 2, \dots, K\}$
I^c	Complement set of indices I from the universal set $\{1, 2, \dots, K\}$
i	Index subscripts
J_C	Selected indices from I
J_W	Selected indices from I^c
j	Index subscripts
$\mathbf{s}_I$	Vector in $\mathbb{R}^{ I }$ consists of the components of $\mathbf{s}$ indexed by $i \in I$
t_{max}	The maximum number of iterations or the halting iteration number
$\mathbf{z} \in \mathbb{R}^d$	Measurement vector $\Phi \mathbf{s}$

Chapter 1

Introduction

The abundance of redundancy in natural signals (information content) led the researchers to think of compact representation of signals or to store the signals in a compact form. The evolving digital world and rising of computational capacity make it possible. Prior arts can be seen from the well-known LZ77 and LZW algorithms, which are the practical realizations of the correlation in neighboring data units [1, 2]. Many contributions are there in the field of Data Compression [3]. Along with this development researchers explored the phenomenon of signal approximation; this gave rise to the world of lossy compression. The idea was to make the signal more compact and portable without compromising the interest. Lossy compression was well adored in the growing field of Communication. A remarkable contribution to this growing field of interest is JPEG. This is still in use for basic mode of transmission for still images, and even some video codecs follow JPEG standards.

As the representation space became a subject of interest for researchers, it gave birth to numerous transforms or domains to analyze/visualize the signals. It starts from Fourier Transform to Wavelets and all kind of *lets*. The detailed history can be found in the text [4]. Scalability of the signal and sparseness in transform domain (notably wavelet) gave a new compression standard called JPEG2000 to the world of Information Engineering [5]. It has both the features of scalability and compactness, which made the successive

approximation or progressive transmission effective. This arouses the interest in the field of sparse representation and signal approximation. However, it amused the researchers that while we are contended with its approximation, we are unnecessarily acquiring the whole signal. This observation gives birth to the concept of compressed sensing, that is, to acquire a sparse signal in a simple manner by taking fewer samples/measurements.

1.1 Sparsity

In the field of sparse representation and compressed sensing, we assume that the signal is sparse (having few nonzero entries). Preferentially, we suppose that any natural signal $\mathbf{x} \in \mathbb{R}^n$ can be represented using an overcomplete dictionary $\mathbf{D} \in \mathbb{R}^{n \times K}$, which contains K atoms (prototype signals $\{\mathbf{d}_j\}_{j=1}^K$). The signal $\mathbf{x}$ can be written as a linear combination of these atoms in exact form $\mathbf{x} = \mathbf{D}\mathbf{s}$ or in approximate form $\mathbf{x} \approx \mathbf{D}\mathbf{s}$, satisfying $\|\mathbf{s}\|_0 \ll n$ ($\|\cdot\|_0$ is ℓ_0 norm, counting the number of nonzero entries in a vector). The vector $\mathbf{s} \in \mathbb{R}^K$ contains the representation coefficients of signal $\mathbf{x}$.

As we mentioned earlier, $\mathbf{D}$ is an overcomplete dictionary. It means $n < K$ and $\mathbf{D}$ is a full rank matrix. This implies for any signal $\mathbf{x}$ there are infinite number of solutions to $\mathbf{x} \approx \mathbf{D}\mathbf{s}$. However, we are only interested in the solution $\mathbf{s}$ that contains least number of nonzero entries, that makes the sparse representation as the solution of either

$$\arg \min_{\mathbf{s}} \|\mathbf{s}\|_0 \text{ such that } \mathbf{x} \approx \mathbf{D}\mathbf{s}, \quad (P_0)$$

or

$$\arg \min_{\mathbf{s}} \|\mathbf{s}\|_0 \text{ such that } \|\mathbf{x} - \mathbf{D}\mathbf{s}\|_2 \leq \epsilon, \quad (P_{0,\epsilon})$$

where ϵ is the allowed representation error. These problems are combinatorial in nature, and very difficult to solve in general. Hence, algorithms which find solutions to the above problems are called pursuits. Finding a quick and surely converging pursuit is an active field of research.

1.2 Dictionary

An overcomplete set of prototype signal atoms forms a dictionary, which we can determine in two ways: either by fixing it as one of the predefined dictionaries, or by building a dictionary from a set of sample signals. Anyone will prefer to choose a predefined dictionary due to its simplicity and availability in literature. Examples of such dictionaries are overcomplete discrete cosine transform, short-time-Fourier transforms, wavelets, curvelets, contourlets, steerable wavelet filters and many more. Success of this method depends on how suitably the dictionaries can sparsify the signal in its representation domain. As mentioned above, multiscale and oriented bases and shift invariance are the guidelines of these traditional bases constructions.

However, the predefined bases are limited compared to the varieties of data sets we have. The signals we sense from any natural phenomenon are random in nature. The randomness in the signal is due to the lack of knowledge of its basis which it can best fit. Modern adaptation theory gives us a chance to get close to the basis where we can claim the signal is optimally sparse. Designing a dictionary that can adapt to the input signal to support and enhance sparsity has always been a subject of interest among the researchers. There exist many works in this direction [6, 7, 8, 9, 10], and part of this thesis contributes towards it.

1.3 Application of Sparsity

Sparsity is a relatively new measure for a signal in the world of signal processing. However, applications using sparse representation are very intuitive. Let's take the most basic inverse problem of removing noise from a signal $\mathbf{y} = \mathbf{x} + \mathbf{v}$, where $\mathbf{v}$ is the additive noise. As we know, additive noise is not a well defined signal, so it should not have any sparse representation using some well defined prototype signals. By taking sparsity

as a prior knowledge for the expected signal, we can put it in a Bayesian framework as $\hat{\mathbf{s}} = \arg \min_{\mathbf{s}} \|\mathbf{y} - \mathbf{D}\mathbf{s}\|_2^2 + \mu \|\mathbf{s}\|_0$, where the prior probability is $e^{-\|\mathbf{s}\|_0}$. If our knowledge of $\mathbf{s}$ being sparse in $\mathbf{D}$ is true, we can successfully obtain the noise free estimation $\hat{\mathbf{x}} = \mathbf{D}\hat{\mathbf{s}}$ from the noisy signal $\mathbf{y}$. The equation $P_{0,\epsilon}$ is another manifestation of this Bayesian framework, where ϵ depends on μ .

Another appealing inverse problem is signal inpainting, which can be well treated in the framework of sparsity. We know a priori the signal $\mathbf{x}$ is sparse in dictionary $\mathbf{D}$ satisfying equation P_0 . If some samples are removed from $\mathbf{x}$ at some locations, we can still assume that the sparse vector $\mathbf{s}$ will remain unchanged, on the new dictionary $\bar{\mathbf{D}}$ formed by removing the samples at the same locations of the atoms. We need to obtain $\hat{\mathbf{s}} = \arg \min_{\mathbf{s}} \|\mathbf{s}\|_0$ such that $\bar{\mathbf{D}}\mathbf{s} = \bar{\mathbf{x}}$, where $\bar{\mathbf{x}}$ is the signal with missing samples. Thus the recovered signal will be $\hat{\mathbf{x}} = \mathbf{D}\hat{\mathbf{s}}$. Some of the recently explored frameworks using sparsity prior can be found in [11, 12, 13], and part of this thesis contributes towards these intriguing applications.

1.4 Compressed Sensing

The knowledge of signal sparsity not only helps solving inverse problems, but also helps acquire it compressively. Compressed sensing (CS) is about measuring the sparse signals from a limited number of linear projections at a subNyquist rate. It is a growing field of interest for researchers [14]. Through d linear projections $\mathbf{z} \in \mathbb{R}^d$, CS measures a K -dimensional real valued sparse signal $\mathbf{s} \in \mathbb{R}^K$, where $K \gg d$. In CS, we stack N projection vectors to form a measurement matrix $\Phi \in \mathbb{R}^{d \times K}$, and that makes $\mathbf{z} = \Phi\mathbf{s}$. The core idea of CS relies on the fact that measured signal $\mathbf{s}$ is sparse, i.e. $\|\mathbf{s}\|_0 \ll K$. CS also extends to the signals which are compressible in some basis or frame.

The first problem in CS is to find a measurement matrix that ensures every m -sparse

signal (i.e. $\|\mathbf{s}\|_0 = m$) has unique measurements. The following theorem gives an example of a desirable measurement matrix.

Theorem 1.1 (Theorem 1 of [15]) *Let $d \geq C_1 m \ln \frac{K}{m}$, and Φ has $d \times K$ Gaussian i.i.d entries. The following statement is true with probability exceeding $1 - e^{-c_1 d}$. It is possible to reconstruct every m -sparse signals $\mathbf{s} \in \mathbb{R}^K$ from the data $\mathbf{z} = \Phi \mathbf{s}$.*¹

In order to bring generality, we usually quantify Φ using the Restricted Isometry Property (RIP). Any matrix Φ satisfies RIP of order m , if there exists a constant $0 \leq \delta_m < 1$ for which the following statement holds $\forall \|\mathbf{s}\|_0 \leq m$.

$$(1 + \delta_m) \|\mathbf{s}\|_2^2 \geq \|\Phi \mathbf{s}\|_2^2 \geq (1 - \delta_m) \|\mathbf{s}\|_2^2 \quad (\text{RIP})$$

In other words, any combination of m or less columns from Φ will form a well conditioned submatrix. Hence, if Φ has a RIP of order $2m$, it guarantees unique measurements for any m -sparse signal. Thus theorem 1.1 means, the Gaussian measurement matrix with $d = O\left(m \ln \frac{K}{m}\right)$ satisfies RIP of order $2m$.

The second problem in CS is to find a suitable algorithm, which can recover any sparse signal exactly from its unique measurements,

$$\hat{\mathbf{s}} = \arg \min_{\mathbf{s}} \|\mathbf{s}\|_0 \text{ such that } \mathbf{z} \approx \Phi \mathbf{s}. \quad (L_0)$$

Part of my thesis focuses on this problem, where typically two major questions are addressed:

- 1) Knowing that the measured signal $\mathbf{s}$ is sparse, i.e. $\|\mathbf{s}\|_0 \ll n$, can an algorithm reconstruct it exactly?
- 2) How many measurements are necessary for the algorithm to work?

¹Throughout the text, we have indicated positive universal constants as C_n, c_n , etc.

1.5 Contributions of the Thesis

- The thesis contributes a new dictionary training algorithm called Sequential Generalization of K -means (SGK). SGK is sequential like K -SVD [9], and it does not modify the sparse representation coefficients like MOD [6]. Hence, it overcomes the limitations of both K -SVD and MOD. The computational complexities for all the three algorithms K -SVD, MOD and SGK are analyzed and compared. It is shown that MOD is least complex followed by SGK. Since, MOD is a resource grasping parallel update procedure, SGK should be chosen as the sequential alternative.
- The thesis demonstrates three image processing frameworks using trained dictionaries, that is image compression, image inpainting, and image denoising. In the image compression framework, the sparse representation coefficients of the non-overlapping image blocks are coded like JPEG. In image inpainting framework, the missing pixels of the non-overlapping image blocks are recovered by estimating their sparse representation from the available pixels. In image denoising framework, the image is recovered by estimating the sparse representation of the overlapping image blocks and averaging them. Extensive comparisons are made between K -SVD and SGK using the above frameworks, which shows SGK to be an efficient alternative to K -SVD in practice.
- The thesis contributes an adaptive local block size based sparse representation framework to have a better recovery (inpainting and denoising) of the underlying image details. Simple local block size selection criteria are introduced for image recovery. A maximum a posteriori probability (MAP) based aggregation formula is derived to inpaint the global image from the overlapping local inpainted blocks. A block size based representation error threshold is derived to perform equiprobable

denoising of the image blocks of various size. The proposed inpainting framework produces a better inpainting result compared to the state of the art techniques. In the case of heavy noise, the proposed local block size selection based denoising framework produces a relatively better denosing compared to the recently proposed image denoising techniques based on sparse representation.

- The thesis contributes two new schemes of OMP for sparse signal recovery from CS measurements. Theoretical guarantees on required number of measurements for exact signal recovery are derived. OMP for CS recovery of the sparse signals is analyzed, where a proposition is stated to highlight the behavior of OMP. As a result of this analysis, two new scheme of OMP called OMP_α and OMP_∞ are proposed. A proposition is stated to describe the events of success and failure for OMP_α , which leads to the analysis of its recovery performance. OMP_∞ is proposed as a further extension to OMP_α , which does not need any prior knowledge of sparsity like BP. The required number of measurements for OMP_α and OMP_∞ is derived, which in same order as that of BP.

1.6 Organization of the Thesis

The thesis consists of seven chapters. The first chapter introduces the works presented in the thesis. The second chapter briefs on the prior and related works. The third chapter takes the reader through the details of generalization of K -means for dictionary training, where a Sequential Generalization of K -means (SGK) is proposed for dictionary training. The fourth chapter illustrates the applications of trained dictionaries in image compression and image recovery, where the usability of SGK is demonstrated in practice. The fifth chapter proposes a framework to improve the image recovery performance using sparse representation, where the local block sizes are adaptively chosen from the

corrupt image. The sixth chapter investigates the recovery of sparse signals from CS measurements. It analyzes the orthogonal matching pursuit (OMP) algorithm for better signal recovery in the case of random measurements, and two new schemes of OMP are proposed. The seventh chapter concludes and speculates on some future work extensions.

Chapter 2

Literature Review

2.1 Dictionary

In the recent years, sparse representation has emerged as a new tool for signal processing. Given a dictionary $\mathbf{D} \in \mathbb{R}^{n \times K}$ containing prototype signal 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 $\hat{\mathbf{x}} = \mathbf{D}\mathbf{s}$, 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 prespecified set of linear transforms (e.g. Fourier, Cosine, Wavelet, etc.) or can be trained on a set of training signals.

Given a set of training signals, a trained $\mathbf{D}$ will always produce a better sparse representation in comparison to traditional parametric bases. This is because, for a set of training signals $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]$, $\mathbf{D}$ is trained to minimize the representation error,

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

with a constraint that $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_N]$ are the sparse representations of $\{\mathbf{x}_i\}$. $\|\mathbf{E}\|_F = \sqrt{\sum_{ij} E_{ij}^2}$ is the Frobenius norm of matrix $\mathbf{E} = \mathbf{X} - \mathbf{D}\mathbf{S}$. Noting that the error minimization depends both on $\mathbf{S}$ and $\mathbf{D}$, the solution is obtained iteratively by alternating between *sparse coding* (for $\mathbf{X}$) and *dictionary update* (for $\mathbf{D}$). Some known contributions

in this field are Method of Optimal Directions (MOD) [6], Union of Ortho-normal Bases [7], Generalized PCA [8], and K-SVD [9].

2.1.1 Method of Optimal Directions (MOD)

Given a set of training signals $\mathbf{X}$, and an initial dictionary $\mathbf{D}$, the aim of MOD is to find the sparse representation coefficient matrix $\mathbf{S}$ and an updated dictionary $\mathbf{D}$ as the solution to (Eq. 2.1) [6]. The resulting optimization problem is highly non-convex in nature, thus we hope to obtain a local minimum at best. Therefore, it alternates between two steps. In the first step, it performs the sparse coding of the training signals using a pursuit algorithm on the initial dictionary. Then in the second step, it updates the dictionary by analytically solving the quadratic problem (Eq. 2.1) for $\mathbf{D}$. It is given by $\mathbf{D} = \mathbf{X}\mathbf{S}^\dagger$, where $\mathbf{S}^\dagger$ denotes the generalized matrix inverse of $\mathbf{S}$ (the sparse representation coefficient matrix obtained in the first step).

The MOD is overall a very effective method, and it requires some number of iterations to converge. The only drawback of the method is that it requires a matrix inversion.

2.1.2 Union of Orthonormal Bases (UOB)

Training a dictionary as a union of orthonormal bases, is a very recent idea. It uses SVD in dictionary update, rather than generalized matrix inverse like MOD. It is one of the first attempts to train a structured overcomplete dictionary. The suggested model is to train a concatenation of L orthonormal bases, that is $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, \dots, \mathbf{D}_L]$, where $\mathbf{D}_i \in \mathbb{R}^{n \times n}$ is an orthonormal basis. It shares the same idea of alternate sparse coding of the given set of training signals $\mathbf{X}$ followed by the dictionary update step. It uses BCR (Block Coordinate Relaxation) algorithm to compute the representation coefficients $\mathbf{S}_i$ for each orthonormal basis $\mathbf{D}_i$ [16]. The detailed algorithm steps are as follows.

- (i) Choose an initial dictionary $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, \dots, \mathbf{D}_L]$;

- (ii) Update the coefficients $\mathbf{S}^T = [\mathbf{S}_1^T, \mathbf{S}_2^T, \dots, \mathbf{S}_L^T]$ using the current $\mathbf{D}$;
- (iii) Repeat the following steps for all the basis $\mathbf{D}_k$:
 - (a) Compute $\mathbf{E}_k = \mathbf{X} - \sum_{i \neq k} \mathbf{D}_i \mathbf{S}_i^T$
 - (b) Compute the singular value decomposition: $\mathbf{E}_k \mathbf{S}_k = \mathbf{U} \Delta \mathbf{V}^T$
 - (c) Update $\mathbf{D}_k = \mathbf{U} \mathbf{V}^T$
- (iv) If the stopping criterion is not reached, go to step 2.

Interestingly, one after another sequential update of UOB reminds us of K -means clustering. However, a drawback of this algorithm is its restrictive form of the union of orthogonal base, which constrains the number of atoms to integer multiple of signal dimensions. Generalized PCA is going to be discussed in the next subsection, where some similarities with UOB can be found.

2.1.3 Generalized Principal Component Analysis (GPCA)

GPCA offers a very different approach to overcomplete dictionary design, which is an extension of Principal Component Analysis (PCA) formula. PCA approximates a higher dimensional signal set into some lower dimensional subspace, whereas GPCA approximates a given set of training signals into a union of several low dimensional subspaces of unknown dimensionality. In [7], an algebraic geometric approach is illustrated to determine the number subspaces, and orthogonal bases for them.

One good thing about GPCA is that it determines the number of atoms in the dictionary by itself. In GPCA, each training signal is mapped using a set of atoms to its associated subspace. Combination of atoms cannot span across subspaces, which is different from the classical sparsity model viewpoint. If we want to look at GPCA from

classical sparse modeling viewpoint, it appears that several distinct dictionaries are allowed to coexist, and each training signal is assumed to be exactly sparse on one of these distinct dictionaries.

2.1.4 K -SVD

At present, the sequential dictionary training algorithm K -SVD has become a benchmark in dictionary training [9]. In the dictionary update procedure, instead of using an unstable generalized matrix inversion like MOD, K -SVD uses stable Singular Value Decomposition (SVD) operations like UOB. One variation in K -SVD is that it does not update the dictionary as a whole. It uses a far simpler sparse coding followed by K times atom by atom update using SVD. Hence, it acquires the name K -SVD. It is claimed that K -SVD is advantageous over MOD in terms of speed and accuracy [9]. However, both MOD and K -SVD are reminiscent of long-known K -means clustering for codebook design in Vector Quantization (VQ) [17]. The next chapter analyzes both the algorithms from the viewpoint of K -means, and it proposes a sequential generalization of K -means for dictionary training.

2.2 Sparse Coding

Sparse coding is the procedure to compute the sparse representation coefficient $\mathbf{s}$, for a given signal $\mathbf{x}$ on a dictionary $\mathbf{D}$. This procedure is also referred as atomic decomposition in literatures. Basically, we have to find the solution to the following problems,

$$\begin{aligned} (P_0) \quad & \arg \min_{\mathbf{s}} \|\mathbf{s}\|_0 \text{ such that } \mathbf{x} \approx \mathbf{D}\mathbf{s}, \\ (P_{0,\epsilon}) \quad & \arg \min_{\mathbf{s}} \|\mathbf{s}\|_0 \text{ such that } \|\mathbf{x} - \mathbf{D}\mathbf{s}\|_2 \leq \epsilon, \end{aligned}$$

where (P_0) means exact solution and $(P_{0,\epsilon})$ means approximate solution with an error tolerance of ϵ . It is very difficult to solve a constrained minimization problem with ℓ_0 -norm as the requirement function, because it is combinatorial in nature. Therefore, these

NP-hard problems are solved using pursuit algorithm, which is an alternative approach to the solution. Several promising sparse coders can be found in the literatures, which includes Method of Frames (MOF) [18], Best Orthogonal Basis (BOB) for special dictionaries [19], Matching Pursuit (MP) [20], Orthogonal Matching Pursuit (OMP) [21], Focal Under-determined System Solver (FOCUSS) [22], and Basis Pursuit (BP) [23]. Since sparse coding is a very basic requirement for any problem in the world of sparsity, some of these methods are briefed in the following subsections.

2.2.1 Orthogonal Matching Pursuit (OMP)

Orthogonal matching pursuit is a greedy stepwise converging algorithm. At each step of the algorithm it selects the dictionary element having the maximum projection on to the residue or error signal space. In this sense, it tries to approximate signal $\mathbf{x}$ in each step by adding details. The approximation error is called the residue. In this algorithm, it is assumed that the columns of the dictionary are ℓ_2 normalized. It starts with an initial setup as residue $\mathbf{r}_t = \mathbf{x}$ at iteration $t = 0$.

(i) Select the index of the next dictionary element $\lambda_t = \arg \max_{j=1,\dots,K} |\langle \mathbf{d}_j, \mathbf{r}_{t-1} \rangle|$.

(ii) Update the current approximation

$$\mathbf{x}_t = \arg \min_{\mathbf{x}_t} \|\mathbf{x} - \mathbf{x}_t\|_2^2, \text{ such that } \mathbf{x}_t \in \mathbf{R}\{\mathbf{d}_{\lambda_1}, \mathbf{d}_{\lambda_2}, \dots, \mathbf{d}_{\lambda_t}\}.$$

(iii) Update the residual $\mathbf{r}_t = \mathbf{x} - \mathbf{x}_t$.

The algorithm can be stopped after a predetermined number of steps or after reaching the maximum residual norm. This algorithm is very effective, simple and easily programmable. It is extensively used in all the experiments of the thesis.

2.2.2 Basis Pursuit (BP)

The basis pursuit algorithm proposes that if we replace the ℓ_0 -norm with ℓ_1 -norm in problem (P_0) and $(P_{0,\epsilon})$, the solutions will be indifferent. Therefore, it solves

$$(P_1) \quad \arg \min_{\mathbf{s}} \|\mathbf{s}\|_1 \text{ such that } \mathbf{x} \approx \mathbf{D}\mathbf{s},$$

for exact representation of the signal, and

$$(P_{1,\epsilon}) \quad \arg \min_{\mathbf{s}} \|\mathbf{s}\|_1 \text{ such that } \|\mathbf{x} - \mathbf{D}\mathbf{s}\|_2 \leq \epsilon,$$

for approximate sparse representation. The advantage of using ℓ_1 norm is that the exact solution (P_1) can be solved by linear programming structure, and the approximate solution $(P_{1,\epsilon})$ can be solved through quadratic programming structure. Thus, any available optimization toolbox can do the sparse coding for us. However, its computational complexity can be more than that of OMP.

2.2.3 FOCUSS

Focal Undetermined System Solver is an approximation algorithm to find the solution to (P_0) or $(P_{0,\epsilon})$ by replacing ℓ_0 -norm with ℓ_p -norm, where $p \leq 1$. Therefore in this method (P_0) become

$$(P_p) \quad \arg \min_{\mathbf{s}} \|\mathbf{s}\|_p^p \text{ such that } \mathbf{x} \approx \mathbf{D}\mathbf{s},$$

where $\|\mathbf{s}\|_p^p = \sum_{i=1}^K |\mathbf{s}(i)|^p$. The use of Lagrange multiplication vector $\lambda \in \mathbb{R}^n$ produces the Lagrange function

$$L(\mathbf{s}, \lambda) = \|\mathbf{s}\|_p^p + \lambda^T (\mathbf{x} - \mathbf{D}\mathbf{s})$$

Hence, in order to solve problem (P_p) , we have to minimize L . This implies the conditions for pair $(\mathbf{s}, \lambda)$ are

$$\Delta_{\mathbf{s}} L(\mathbf{s}, \lambda) = p\mathbf{I}(\mathbf{s})\mathbf{s} - \mathbf{D}^T \lambda = 0,$$

$$\Delta_\lambda L(\mathbf{s}, \lambda) = \mathbf{x} - \mathbf{D}\mathbf{s} = 0,$$

where $\mathbf{I}(\mathbf{s})$ is defined as a diagonal matrix of dimension $n \times n$ having diagonal entries as $|\mathbf{s}(i)|^{p-2}$ for $i = 1, 2, \dots, K$. The separation of $\Delta_\lambda L(\mathbf{s}, \lambda)$ into multiplication of $\mathbf{I}(\mathbf{s})$ weight matrix and vector $\mathbf{s}$ is the main idea of FOCUSS. Several simple steps of algebra lead to the solution

$$\mathbf{s} = \mathbf{I}(\mathbf{s})^{-1} \mathbf{D}^T (\mathbf{D} \mathbf{I}(\mathbf{s})^{-1} \mathbf{D}^T)^{-1} \mathbf{x}.$$

However, this type of closed form solution is impossible to achieve. Hence it is reformulated to an iterative form

$$\mathbf{s}_t = \mathbf{I}(\mathbf{s}_{t-1})^{-1} \mathbf{D}^T (\mathbf{D} \mathbf{I}(\mathbf{s}_{t-1})^{-1} \mathbf{D}^T)^{-1} \mathbf{x}.$$

Parallel expressions can be derived quite similarly for the treatment of $(P_{0,\epsilon})$,

$$(P_{p,\epsilon}) \quad \arg \min_{\mathbf{s}} \|\mathbf{s}\|_p^p \text{ such that } \|\mathbf{x} - \mathbf{D}\mathbf{s}\|_2 \leq \epsilon.$$

However, in this case the determination of the Lagrange multiplier is more difficult, and must be searched within the algorithm [24].

2.3 Image Recovery Problems

The natural images are generally sparse in some transform domain, which makes sparse representation an emerging tool to solve image processing problems.

2.3.1 Inpainting

Inpainting is a problem of filling up the missing pixels in an image by taking help of the existing pixels. In literatures, inpainting is often referred as disocclusion, which means to remove an obstruction or unmask a masked image. The success of inpainting lies on how well it infers the missing pixels from the observed pixels. It is a simple form of inverse

problem, where the task is to estimate an image $X \in \mathbb{R}^{\sqrt{N} \times \sqrt{N}}$ from its measurement $Y \in \mathbb{R}^{\sqrt{N} \times \sqrt{N}}$ which is obstructed by a binary mask $B \in \{0, 1\}^{\sqrt{N} \times \sqrt{N}}$.

$$Y = X \circ B : B(i, j) = \begin{cases} 1 & \text{if } (i, j) \text{ is observed} \\ 0 & \text{if } (i, j) \text{ is obstructed} \end{cases} \quad (\text{Eq. 2.2})$$

In literature, the problem of image inpainting has been addressed from different points of view, such as Partial Differential Equation (PDE), variational principle and exemplar region filling. An overview of these methods can be found in these recent articles [25, 26]. Apart from these approaches, use of explicit sparse representation has produced very promising inpainting results [12, 13]. Natural images are generally sparse in some transform domain, which makes sparse representation as an emerging tool for solving image processing problems. Inpainting is a fundamental problem in sparse representation which supports the arguments from compressed sensing [14], where random sampling is one of the techniques.

2.3.2 Denoising

Growth of semiconductor technologies has made the sensor arrays overwhelmingly dense, which makes the sensors more prone to noise. Hence denoising still remains an important research problem in image processing. Denoising is a form of challenging inverse problem, where the task is to estimate the signal X from its measurement Y which is corrupted by additive noise V ,

$$Y = X + V. \quad (\text{Eq. 2.3})$$

Note that the noise V is commonly modelled as Additive White Gaussian Noise (AWGN).

In literature, the problem of image denoising has been addressed from different points of view such as statistical modeling, spatial adaptive filtering, and transfer domain thresholding [27]. In recent years image denoising using sparse representation has been proposed. The well known shrinkage algorithm by D. L. Donoho and L. M. Johnstone [28]

is one example of such approach. In [11], M. Elad and M. Aharon has explicitly used sparsity as a prior for image denoising. In [29], P. Chatterjee and P. Milanfar have clustered an image into K clusters to enhance the sparse representation via locally learned dictionaries.

2.4 Compressed Sensing Recovery

Recovering a sparse signal from its CS measurements is one of the intriguing fields of research. Basically, the techniques are the same as finding sparse solution to an undetermined linear system of equations that we have discussed earlier. However, the dictionary $\mathbf{D}$ is replaced by the measurement matrix Φ in (P_0) , (P_1) and (P_p) . The two broad classes of such techniques are convex relaxation [23, 30], and iterative greedy pursuit [20, 21, 31]. The convex relaxation technique is well known as Basis Pursuit (BP), which changes the objective from ℓ_0 -norm minimization to ℓ_1 -norm minimization,

$$\hat{\mathbf{s}} = \arg \min_{\mathbf{s}} \|\mathbf{s}\|_1 \text{ such that } \mathbf{z} \approx \Phi \mathbf{s}. \quad (L_1)$$

In contrast, the greedy pursuits iteratively identify the nonzero indices of $\mathbf{s}$. Due to its theoretically provable recovery performance, the convex relaxation technique has gained more importance in comparison to greedy pursuit.

BP can exactly reconstruct an m -sparse signal with high probability, when Φ satisfies Restricted Isometry Property (RIP) of order $2m$ with $\delta_{2m} < \sqrt{2} - 1$ [32]. As a result, it only requires $N = O\left(m \ln \frac{d}{m}\right)$ for the case of Gaussian measurement matrices. However, BP is computationally more demanding, which requires $O\left(N^2 d^{3/2}\right)$ number of operations [33]. In contrast, the greedy pursuits are faster, and can be useful for large scale CS problems. One of the fundamental greedy pursuit techniques is OMP [34], which requires only $O(mNd)$ number of operations [35]. It minimizes the ℓ_2 norm of the residue by

selecting one atom in each iteration, where atoms refer to $\varphi_j \in \mathbb{R}^N$, the columns of the measurement matrix Φ . Some of the theoretical guarantees for OMP have been established in [34, 36, 37]. The best result shows, OMP can recover m -sparse signals exactly with high probability, when $N = O(m \ln d)$ [15]. For the sake of completeness the OMP algorithm is detailed next.

Algorithm 2.1 (OMP for CS Recovery)

Input:

- *measurement matrix* $\Phi \in \mathbb{R}^{d \times K}$
- *measurement* $\mathbf{z} \in \mathbb{R}^d$
- *maximum iterations* $t_{\max}$

Output:

- *signal estimation* $\hat{\mathbf{s}}$
- *index set* Λ_t *containing elements from* $\{1, \dots, K\}$
- *residual* $\mathbf{r}_t \in \mathbb{R}^d$

Procedure:

- (i) Initialize: *residual* $\mathbf{r}_0 = \mathbf{z}$, *index set* $\Lambda_0 = \emptyset$ *and iteration counter* $t = 0$;
- (ii) Increment $t = t + 1$;
- (iii) choose the atom $\lambda_t = \arg \max_{j=1, \dots, K} |\langle \varphi_j, \mathbf{r}_{t-1} \rangle|$
- (iv) Update $\Lambda_t = \Lambda_{t-1} \cup \{\lambda_t\}$;
- (v) Update $\mathbf{a}_t = \Phi_{\Lambda_t}^\dagger \mathbf{z}$;
- (vi) Update $\mathbf{r}_t = \mathbf{z} - \Phi_{\Lambda_t} \mathbf{a}_t$;
- (vii) Go to Step.2 if $t < t_{\max}$, else terminate;
- (viii) The estimation $\hat{\mathbf{s}}$ for the signal $\mathbf{s}$ has nonzero elements at Λ_t and rest zeros, i.e. $\hat{\mathbf{s}}_{\Lambda_t} = \mathbf{a}_t$.

Figure 2.1: OMP for CS Recovery

OMP begins by initializing the residual to the input measurement vector $\mathbf{r}_0 = \mathbf{z}$, and the selected index set to empty set $\Lambda_0 = \emptyset$. At iteration t , OMP chooses a new index λ_t by finding the best atom matching with the residual,

$$\lambda_t = \arg \max_{j=1,\dots,K} |\langle \varphi_j, \mathbf{r}_{t-1} \rangle|,$$

and updates the selected index set $\Lambda_t = \Lambda_{t-1} \cup \{\lambda_t\}$. Here $|\langle \varphi_j, \mathbf{r}_{t-1} \rangle|$ stands for the absolute dot product of the residue vector $\mathbf{r}_t$ with the atoms φ_j . Then, OMP obtains the best t -term approximation by a Least-Squares (LS) minimization. That is,

$$\mathbf{a}_t = \arg \min_{\mathbf{a}} \|\mathbf{z} - \Phi_{\Lambda_t} \mathbf{a}\|,$$

which has a close form solution $\mathbf{a}_t = \Phi_{\Lambda_t}^\dagger \mathbf{z}$, where $\Phi_{\Lambda_t}^\dagger = (\Phi_{\Lambda_t}^T \Phi_{\Lambda_t})^{-1} \Phi_{\Lambda_t}^T$. LS procedure in OMP [21] brings a significant improvement in comparison to its parent algorithm, the Matching Pursuit (MP) [20].

2.5 Summary

The motivation for dictionary training is introduced. Some recent dictionary training algorithms, MOD, UOB, GPCA, and K -SVD are briefly reviewed. The aforementioned algorithms bear some resemblance to K -means, especially K -SVD. K -SVD is popular among the researchers due to its convergence and sequential update structure. However, the use of SVD makes it computationally demanding, and limits its usage to unit norm atoms. Alongside, it is difficult for SVD to cater to a dictionary training for all kind of sparse representation, such as constrained representation like Vector Quantization (VQ). Thus, the motivation is to overcome the limitations of K -SVD and propose an alternative dictionary training algorithm.

One of the well known application of sparse representations is image recovery (in-painting, denoising), which is briefly reviewed. Since sparsity leads to these applications,

it is important to set up a common platform that can verify the usefulness of the sparsifying dictionary. Therefore, the motivation is to illustrate image processing applications like inpainting and denoising, which can evaluate the proposed dictionary.

Global recovery through the aggregation of local recovery, as presented, is the main framework of image recovery using sparse representation, where a predefined local block size is assigned. The objective of local recovery is to simplify the problem, because it is easier to enforce sparsity in smaller image blocks. Since the signal characteristics inside a local block vary from location to location, it motivates proposing an adaptive block size selection based image recovery framework.

The key element of sparse signal processing is the sparse coder, or the pursuit that gives the sparse representation. Three important sparse coders, OMP, BP, and FOCUSS are reviewed. Among them, OMP is popular due to its simplicity and swift execution speed. Therefore, it has been extensively used as the sparse coder for all the experiments carried out in the thesis.

Compressive sensing (CS) has become an intuitive quest once a signal is known to be sparse, which is briefly reviewed. The recovery of sparse signal from CS measurements needs a sparse coder as well, where the present implementation of OMP has an inferior recovery guarantee compared to BP. This motivates proposing a new scheme of signal recovery using OMP to improve its recovery guarantee.

Chapter 3

Dictionary Training

The celebrated algorithms such as K -SVD [9] and MOD [6] are reminiscent of long-known K -means clustering used for codebook design (dictionary training) in Vector Quantization (VQ) [17]. Similar to K -means, they train the dictionary iteratively, by alternating between *sparse coding* (for $\mathbf{S}$) and *dictionary update* (for $\mathbf{D}$) as described in figure 3.1.

Algorithm 3.2 (Dictionary Training) *Input : Trsamples $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]$, where $\mathbf{x}_i \in \mathbb{R}^n$; initial dictionary $\mathbf{D}^{(0)} \in \mathbb{R}^{n \times K}$.*

Procedure : Initialize $t = 0$, and repeat until convergence:

1) Sparse coding stage: Obtain $\mathbf{S}^{(t)} = [\mathbf{s}_1^{(t)}, \mathbf{s}_2^{(t)}, \dots, \mathbf{s}_N^{(t)}]$ for $\mathbf{X}$ as

$$\forall_i \mathbf{s}_i^{(t)} = \arg \min_{\mathbf{s}_i} \|\mathbf{x}_i - \mathbf{D}^{(t)} \mathbf{s}_i\|_2^2 : \|\mathbf{s}_i\|_0 \leq m_{max}, \quad (\text{Eq. 3.1})$$

where m_{max} is the admissible number of coefficients.

2) Dictionary update stage: For the obtained $\mathbf{S}^{(t)}$, update $\mathbf{D}^{(t)}$ such that

$$\mathbf{D}^{(t+1)} = \arg \min_{\mathbf{D}} \|\mathbf{X} - \mathbf{D} \mathbf{S}^{(t)}\|_F^2, \quad (\text{Eq. 3.2})$$

and increment $t = t + 1$.

Figure 3.1: Dictionary training algorithm for sparse representation, the superscript $(.)^{(t)}$ denotes the matrices and the vectors at iteration number t .

This chapter investigates how K -means clustering may be generalized to sparse representation. It starts with a brief analysis of K -means. In the next sections, K -SVD and MOD are elaborated, and their analogy to K -means is discussed. It is shown that K -SVD in its present form fails to retain any *structured/constrained* sparsity such as VQ, as a result of which, it does not simplify to K -means. Use of SVD interferes with the sparse coding, and also restricts the signal-atoms to unit norm. In contrast, it is shown that MOD retains any *structured/constrained* sparsity such as VQ, and simplifies to K -means, hence it may be claimed as a parallel generalization of K -means clustering.

However, in many practical scenarios sequential algorithms are desirable to operate with minimum computational resources. Thus a sequential alternative to MOD is proposed, which is referred as SGK. In the subsequent sections the computational complexity is analyzed, and the training performances are examined experimentally. The results suggest a very much comparable training performance across the algorithms, and MOD takes the least execution time followed by SGK.

3.1 K -means Clustering for VQ

Vector quantization is an extreme form of sparse representation, where dictionary $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_K]$ is termed as codebook. This extreme sparse representation is restricted to trivial basis in $\mathbb{R}^K$, that is, $\mathbf{s} = e_k$ has all 0s except 1 in the k^{th} position. Hence, a signal $\mathbf{x}_i$ represented by some e_k will have the approximation as $\hat{\mathbf{x}}_i = \mathbf{d}_k$. To minimize the representation error, VQ codebook typically is trained using K -means clustering algorithm. It is an iterative process similar to dictionary training which alternates between finding sparse representation $\mathbf{S}$ and updating dictionary $\mathbf{D}$. The detailed steps are as follows.

- 1) *Sparse coding (encoding) stage*: This stage involves finding a trivial basis in $\mathbb{R}^K$ for each signal $\mathbf{x}_i$, so (Eq. 3.1) becomes

$$\forall_i \mathbf{s}_i^{(t)} = \arg \min_{\mathbf{s}_i} \|\mathbf{x}_i - \mathbf{D}^{(t)} \mathbf{s}_i\|_2^2 : \mathbf{s}_i \in \{e_1, e_2, \dots, e_K\}. \quad (\text{Eq. 3.3})$$

As a result, $\mathbf{X}$ is partitioned into K disjoint clusters,

$$\{1 : N\} = \left\{ R_1^{(t)} \cup R_2^{(t)} \dots \cup R_K^{(t)} \right\},$$

where each cluster $R_k^{(t)} = \left\{ i : 1 \leq i \leq N, \mathbf{s}_i^{(t)}(k) = 1 \right\} = \left\{ i : 1 \leq i \leq N, \mathbf{s}_i^{(t)} = e_k \right\} = \left\{ i : 1 \leq i \leq N, \hat{\mathbf{x}}_i^{(t)} = \mathbf{d}_k^{(t)} \right\}.$

- 2) *Dictionary update (codebook design) stage*: The codebook is updated using the nearest neighbor rule. In order to minimize its representation error, each signal-atom (codeword) $\mathbf{d}_k$ is updated individually as

$$\mathbf{d}_k^{(t+1)} = \arg \min_{\mathbf{d}_k} \sum_{i \in R_k^{(t)}} \|\mathbf{x}_i - \mathbf{d}_k\|_2^2 = \frac{1}{|R_k^{(t)}|} \sum_{i \in R_k^{(t)}} \mathbf{x}_i. \quad (\text{Eq. 3.4})$$

Hence, (Eq. 3.2) reduces to

$$\mathbf{D}^{(t+1)} = \left[\frac{1}{|R_1^{(t)}|} \sum_{i \in R_1^{(t)}} \mathbf{x}_i, \frac{1}{|R_2^{(t)}|} \sum_{i \in R_2^{(t)}} \mathbf{x}_i, \dots, \frac{1}{|R_K^{(t)}|} \sum_{i \in R_K^{(t)}} \mathbf{x}_i \right].$$

This algorithm acquired the name K -means because it updates the signal-atoms as K distinct means of the training signals. Note that K -means clustering should not be misinterpreted as a sequential update process for K atoms. As VQ represents each training signal via only one distinct atom, it produces disjoint clusters, i.e. $\forall_{i \neq j} \{R_i \cap R_j\} = \emptyset$. Thus the global minimization of (Eq. 3.2) becomes equivalent to the sequential minimization of each cluster in (Eq. 3.4).

3.2 K -SVD

In the dictionary update stage, K -SVD breaks the global minimization problem (Eq. 3.2) into K sequential minimization problems [9]. It considers each column $\mathbf{d}_k$ in $\mathbf{D}$ and its corresponding row $\mathbf{S}_k$ of the coefficient matrix $\mathbf{S}$, where $\mathbf{S}^T = [\mathbf{S}_1^T, \mathbf{S}_2^T, \dots, \mathbf{S}_K^T]$. Thus the representation error term, $\|\mathbf{E}^{(t)}\|_F^2 = \|\mathbf{X} - \mathbf{D}^{(t)}\mathbf{S}^{(t)}\|_F^2$ may be written as

$$\begin{aligned}\|\mathbf{E}^{(t)}\|_F^2 &= \left\| \mathbf{X} - \sum_{j=1}^K \mathbf{d}_j^{(t)} \mathbf{S}_j^{(t)} \right\|_F^2 \\ &= \left\| \left(\mathbf{X} - \sum_{j \neq k} \mathbf{d}_j^{(t)} \mathbf{S}_j^{(t)} \right) - \mathbf{d}_k^{(t)} \mathbf{S}_k^{(t)} \right\|_F^2.\end{aligned}$$

The quest is for the $\mathbf{d}_k \mathbf{S}_k$ which is closest to $\mathbf{E}_k^{(t)} = \mathbf{X} - \sum_{j \neq k} \mathbf{d}_j^{(t)} \mathbf{S}_j^{(t)}$,

$$\{\mathbf{d}_k^{(t+1)}, \hat{\mathbf{S}}_k^{(t)}\} = \arg \min_{\mathbf{d}_k, \mathbf{S}_k} \|\mathbf{E}_k^{(t)} - \mathbf{d}_k \mathbf{S}_k\|_F^2. \quad (\text{Eq. 3.5})$$

In [9] SVD is used to find the closest rank-1 matrix (in Frobenius norm) that approximates $\mathbf{E}_k^{(t)}$ subject to $\|\mathbf{d}_k^{(t+1)}\|_2 = 1$. SVD decomposition is done on $\mathbf{E}_k^{(t)} = \mathbf{U} \Delta \mathbf{V}^T$. $\mathbf{d}_k^{(t+1)}$ is taken as the first column of $\mathbf{U}$, and $\hat{\mathbf{S}}_k^{(t)}$ is taken as the first column of $\mathbf{V}$ multiplied by the first diagonal element of Δ .

Note that different from (Eq. 3.2), both $\mathbf{d}_k$ and $\mathbf{S}_k$ are updated in K -SVD dictionary update stages (apart from updating $\mathbf{S}_k$ in the sparse coding stage). Unlike K -means, if each signal-atom is updated independently, the resulting $\mathbf{D}^{(t+1)}$ may diverge. This is due to the considerable amount of overlap among the clusters $\{R_1, R_2, \dots, R_K\}$, where $R_k^{(t)} = \{i : 1 \leq i \leq N, \mathbf{S}_k^{(t)}(i) \neq 0\}$. Hence, modifying an atom affects other atoms. In order to take care of these overlaps, before updating the next atom, both $\{\mathbf{d}_k^{(t)}, \mathbf{S}_k^{(t)}\}$ are replaced with $\{\mathbf{d}_k^{(t+1)}, \hat{\mathbf{S}}_k^{(t)}\}$. This process is repeated for all K atoms. We should note that K -SVD is an interdependent sequential update procedure, not an independent update procedure like K -means.

However, there are few matters of concern over the simultaneous update of $\{\mathbf{d}_k, \mathbf{S}_k\}$ in (Eq. 3.5) using SVD.

- 1) *Loss of sparsity*: As there is no sparsity control term $\|\mathbf{S}_k^{(t)}\|_0$ in SVD, the least square solution $\hat{\mathbf{S}}_k^{(t)}$ may contain all nonzero entries, which will result in a nonsparse updated representation $\hat{\mathbf{S}}^{(t)}$.
- 2) *Loss of structure/constraint*: Similarly, if any *structured/constrained* sparsity is used in the sparse coding stage of the dictionary training, this *structure* may also not be retained by SVD.
- 3) *Normalized dictionary*: The use of SVD limits the usability of this dictionary training algorithm only to the settings of unit norm atoms, $\|\mathbf{d}_k^{(t+1)}\|_2 = 1$.

To address the *Loss of sparsity* issue, K -SVD restricts the minimization problem of (Eq. 3.5) to only the set of training signals $\mathbf{X}_k^{(t)} = \{\mathbf{x}_i : \mathbf{S}_k^{(t)}(i) \neq 0\} = \{\mathbf{x}_i : i \in R_k^{(t)}\}$. Hence, SVD decomposition is done on only a part of $\mathbf{E}_k^{(t)}$ that keeps the columns from index set $R_k^{(t)}$. However, the *Loss of structure/constraint* issue still remains unaddressed. Let's take an example of a sparse coder with additional *structure/constraint* $Q(\mathbf{s}_i)$,

$$\mathbf{s}_i^{(t)} = \arg \min_{\mathbf{s}_i} \left\{ \|\mathbf{x}_i - \mathbf{D}^{(t)}\mathbf{s}_i\|_2^2 + Q(\mathbf{s}_i) \right\} : \|\mathbf{s}_i\|_0 \leq m_{\max} \quad (\text{Eq. 3.6})$$

K -SVD in its present form updates both $\{\mathbf{d}_k, \mathbf{S}_k\}$ using SVD, which cannot take care of the additional *structure/constraint* $Q(\mathbf{S}_k)$. Similarly, it fails to simplify to K -means for the VQ as elaborated in the next paragraph. Alongside, the issue of *Normalized dictionary* brings further complication to the usability of K -SVD in VQ.

3.2.1 K -means and K -SVD

In order to verify K -SVD as a generalization of K -means clustering, K -SVD is used to update the codebook for VQ, where $\{\mathbf{d}_k^{(t+1)}, \hat{\mathbf{S}}_k^{(t)}\}$ is obtained using SVD decomposition.

First thing to note that, use of SVD will result in $\|\mathbf{d}_k^{(t+1)}\|_2 = 1$ which is not same as the K -means. Secondly, VQ is a binary *structured/constrained sparsity* with only 0 and 1 entries. Hence, even if we obtain $\hat{\mathbf{S}}_k^{(t)}$ by doing SVD only on the selected columns of $\mathbf{E}_k^{(t)}$ from the index set $R_k^{(t)} = \{i : 1 \leq i \leq N, \mathbf{S}_k^{(t)}(i) = 1\}$, all its entries can not be guaranteed to be 1 irrespective of any scaling factor. This is a classical example of discussed *Loss of structure/constraint* issue of K -SVD, which destroys the binary *structure* imposed by VQ. Thus, it can be concluded that K -SVD as presented in [9] is not a generalization of K -means.

3.3 MOD

In the dictionary update stage, MOD analytically solves the minimization problem (Eq. 3.2) [6]. The quest is for a $\mathbf{D}$ that minimizes the error $\|\mathbf{E}^{(t)}\|_F^2 = \|\mathbf{X} - \mathbf{D}\mathbf{S}^{(t)}\|_F^2$ for the obtained $\mathbf{S}^{(t)}$. Thus taking the derivative of $\|\mathbf{E}^{(t)}\|_F^2$ with respect to $\mathbf{D}$, and equating with 0 gives the relationship: $\frac{\partial}{\partial \mathbf{D}} \|\mathbf{E}^{(t)}\|_F^2 = -2(\mathbf{X} - \mathbf{D}\mathbf{S}^{(t)})\mathbf{S}^{(t)T} = 0$, leading to

$$\mathbf{D}^{(t+1)} = \mathbf{X}\mathbf{S}^{(t)T} \left(\mathbf{S}^{(t)}\mathbf{S}^{(t)T} \right)^{-1}. \quad (\text{Eq. 3.7})$$

In each iteration, MOD obtains $\mathbf{S}^{(t)}$ for a given $\mathbf{D}^{(t)}$, and updates $\mathbf{D}^{(t+1)}$ using (Eq. 3.7). MOD doesn't require the atoms of the dictionary to be unit norm. However, if it is required by the sparse coder, the atoms of $\mathbf{D}^{(t+1)}$ may be normalized to unit norm.

It is interesting to note that MOD is a coder independent dictionary training algorithm, which can be used for all sparse representation applications. Let's take the example of sparse coder with additional *structure/constraint* $Q(\mathbf{s}_i)$ as in (Eq. 3.6). As MOD updates $\mathbf{D}$ independent of $\mathbf{S}$, the presence of $Q(\mathbf{S}^{(t)})$ will not affect the minimization in (Eq. 3.7). Hence codebook update for VQ using MOD simplifies to K -means as elaborated in the next paragraph.

3.3.1 K -means and MOD

In order to verify MOD as a generalization of K -means clustering, MOD is used to update the codebook for VQ. In the case of VQ, $\mathbf{S}_k^{(t)}$ has all 0 entries except 1s at the positions $i \in R_k^{(t)}$, that is when $\hat{\mathbf{x}}_i = \mathbf{D}^{(t)} e_k = \mathbf{d}_k^{(t)}$. As it produces disjoint clusters ($\forall_{i \neq j} \{R_i^{(t)} \cap R_j^{(t)}\} = \emptyset$), rows of $\mathbf{S}^{(t)}$ will be orthogonal to each other ($\forall_{j \neq k} \mathbf{S}_j^{(t)} \mathbf{S}_k^{(t)T} = 0$). This gives us

$$\mathbf{S}^{(t)} \mathbf{S}^{(t)T} = \text{diag} \left\{ |R_1^{(t)}|, |R_2^{(t)}|, \dots, |R_K^{(t)}| \right\},$$

where $|R_k^{(t)}| = \mathbf{S}_k^{(t)} \mathbf{S}_k^{(t)T}$, is the number of training signals associated with signal-atom $\mathbf{d}_k^{(t)}$. Similarly, it can be written that

$$\mathbf{X} \mathbf{S}^{(t)T} = \left[\sum_{i \in R_1^{(t)}} \mathbf{x}_i, \sum_{i \in R_2^{(t)}} \mathbf{x}_i, \dots, \sum_{i \in R_K^{(t)}} \mathbf{x}_i \right],$$

because $\mathbf{X} \mathbf{S}_k^{(t)T} = \sum_{i \in R_k^{(t)}} \mathbf{x}_i$. Thus the dictionary update of MOD as in (Eq. 3.7) simplifies to the dictionary update of K -means clustering.

In other words, minimization of the representation error of K -means clustering generalizes to MOD when the trivial basis of VQ is extended to arbitrary sparse representation with an admissible number of coefficients $m_{\max}$. However, it is a parallel update algorithm in contrast to K -means, which may require more resources (e.g. memory, cache and higher bit processors) to execute for large K and N .

3.4 A Sequential Generalization of K -means

Though MOD is suitable for all kind of sparse representation applications, irrespective of *constraints* on sparse coefficient and dictionary, it may demand more computational resource to operate. In contrary, sequential algorithms like K -SVD and K -means can

manage with lesser resources. This leads naturally to the possibility to generalize K -means sequentially for general purpose sparse representation application. Thus, a modification to the problem formulation in (Eq. 3.5) is proposed. If we keep $\mathbf{S}_k^{(t)}$ unchanged, both concerns of loss of sparsity and loss of *structure* of $\hat{\mathbf{S}}^{(t)}$ will no longer be there. Thus the sequential update problem is posed as

$$\mathbf{d}_k^{(t+1)} = \arg \min_{\mathbf{d}_k} \left\| \mathbf{E}_k^{(t)} - \mathbf{d}_k \mathbf{S}_k^{(t)} \right\|_F^2. \quad (\text{Eq. 3.8})$$

The solution to (Eq. 3.8) can be obtained in the same manner as (Eq. 3.7)

$$\mathbf{d}_k^{(t+1)} = \mathbf{E}_k^{(t)} \mathbf{S}_k^{(t)T} \left(\mathbf{S}_k^{(t)} \mathbf{S}_k^{(t)T} \right)^{-1}. \quad (\text{Eq. 3.9})$$

The overlap among $\mathbf{S}_k^{(t)}$'s (clusters R_k) is taken care of by replacing $\mathbf{d}_k^{(t)}$ with $\mathbf{d}_k^{(t+1)}$ before updating the next atom in the sequence. Similar to K -means, this process is repeated for all K atoms sequentially, hence its is called sequential generalization of K -means (SGK).

Similar to MOD, SGK does not constrain the signal-atoms to be unit norm. If required by the sparse coder, all the atoms can be normalized after updating the entire dictionary. Like MOD, the update equation of SGK (Eq. 3.9) is independent of the sparse coder, which remains unaffected by the presence of any additional *structure/constraint* $Q(\mathbf{S}_k^{(t)})$ as per the exemplar coder (Eq. 3.6). Thus, codebook update for VQ using SGK simplifies to K -means as follows.

3.4.1 K -means and SGK

Let's now verify whether SGK is a true generalization of K -means clustering or not. Hence, SGK is used to update the codebook for VQ. In the case of VQ, the sparse coefficients become trivial bases. Similar to the case of MOD, it can be shown that

$$\mathbf{E}_k^{(t)} \mathbf{S}_k^{(t)T} = \left(\mathbf{X} - \sum_{j \neq k} \mathbf{d}_j^{(t)} \mathbf{S}_j^{(t)} \right) \mathbf{S}_k^{(t)T} = \mathbf{X} \mathbf{S}_k^{(t)T} - \sum_{j \neq k} \mathbf{d}_j^{(t)} \mathbf{S}_j^{(t)} \mathbf{S}_k^{(t)T} = \sum_{i \in R_k^{(t)}} \mathbf{x}_i,$$

because $\mathbf{X}\mathbf{S}_k^{(t)T} = \sum_{i \in R_k^{(t)}} \mathbf{x}_i$ and $\forall_{j \neq k} \mathbf{S}_j^{(t)T} \mathbf{S}_k^{(t)} = 0$. Thus by using the fact that $\mathbf{S}_k^{(t)} \mathbf{S}_k^{(t)T} = |R_k^{(t)}|$, the update equation (Eq. 3.9) gives

$$\mathbf{d}_k^{(t+1)} = \frac{1}{|R_k^{(t)}|} \sum_{i \in R_k^{(t)}} \mathbf{x}_i,$$

which is same as K -means. However, the proposed generalization is a sequential update routine unlike MOD.

3.5 Complexity Analysis

Apart from the above analyses of the dictionary training algorithms, complexity of an algorithm plays a key role is its practical usability. Hence we are interested in the complexity analysis of the dictionary update stage. In order to compute the complexity, let's assume that each training signal of length n has a sparse representation with m nonzero entries, and $\mathbf{X}$ contains N such training signals.

3.5.1 K -SVD

In the process of updating $\mathbf{d}_k$ using K -SVD, we need $2n(m-1)|R_k^{(t)}|$ floating point operations (flop) to compute $\mathbf{E}_k^{(t)} = \mathbf{X} - \sum_{j \neq k} \mathbf{d}_j^{(t)} \mathbf{S}_j^{(t)}$ in the restricted index set $R_k^{(t)}$, because the columns of the sparse representation matrix $\{\mathbf{s}_i : i \in R_k^{(t)}\}$ have only $(m-1)$ nonzero entries to be multiplied with remaining $d_{j \neq k}^{(t)}$. Then performing SVD on $n \times |R_k^{(t)}|$ matrix $\mathbf{E}_k^{(t)}$ requires $2|R_k^{(t)}|n^2 + 11n^3$ flops [38], and $|R_k^{(t)}|$ flops to compute $\hat{\mathbf{S}}_k^{(t)}$ by multiplying the first column of $\mathbf{V}$ with the first diagonal element of Δ . This gives a total of $2n(m-1)|R_k^{(t)}| + 2n^2|R_k^{(t)}| + 11n^3 + |R_k^{(t)}|$ flops to update one atom in $\mathbf{D}^{(t)}$. Thus the flops needed for K -SVD will be the sum over all K atoms,

$$\mathcal{T}_{K\text{-SVD}} = 2nm^2N + 2mn^2N + 11n^3K + mN - 2nmN, \quad (\text{Eq. 3.10})$$

because $\mathbf{S}^{(t)}$ contains $\sum_k |R_k^{(t)}| = Nm$ nonzero elements.

3.5.2 Approximate K -SVD

Though SVD gives the closest rank-1 approximation, this step makes K -SVD very slow. Thus in [39] an inexact SVD step was proposed, which makes it faster. In approximate K -SVD, the solution to (Eq. 3.5) is estimated in two steps: 1) $\mathbf{d}_k^{(t+1)} = \mathbf{E}_k^{(t)} \mathbf{S}_k^{(t)T} / \|\mathbf{E}_k^{(t)} \mathbf{S}_k^{(t)T}\|_2$; 2) $\hat{\mathbf{S}}_k^{(t)} = \mathbf{d}_k^{(t+1)T} \mathbf{E}_k^{(t)}$. Thus we need $n(2|R_k^{(t)}| - 1)$ operations to compute $\mathbf{E}_k^{(t)} \mathbf{S}_k^{(t)T}$, approximately $3n$ operations to normalize the atom, and $|R_k^{(t)}|(2n - 1)$ operations to compute $\mathbf{E}_k^{(t)T} \mathbf{d}_k^{(t+1)}$. Including $2n(m - 1)|R_k^{(t)}|$ operations to compute $\mathbf{E}_k^{(t)}$, it needs a total of $2n(m + 1)|R_k^{(t)}| + 2n - |R_k^{(t)}|$ flops to update one atom in $\mathbf{D}^{(t)}$. Thus the flops needed for approximate K -SVD will be the sum over all K atoms,

$$\mathcal{T}_{K\text{-SVDa}} = 2nm^2N + 2nmN + 2nK - mN \quad (\text{Eq. 3.11})$$

3.5.3 MOD

In the case of MOD, we need to derive the number of operations required to compute (Eq. 3.7). It is known that $\mathbf{S}^{(t)}$ is sparse and contains only $N.m$ nonzero entries. Thus, the total number of operations required to perform the multiplication $\mathbf{XS}^{(t)T}$ will sum up to $2nmN - nK$. Likewise, $\mathbf{S}^{(t)}\mathbf{S}^{(t)T}$ will need $2m^2N - K^2$ operations. $\mathbf{S}^{(t)}\mathbf{S}^{(t)T}$ is a symmetric positive definite matrix¹, thus Cholesky factorization can be used to solve the linear inverse problem (Eq. 3.7). Cholesky factorization expresses $\mathbf{A} \in \mathbb{R}^{K \times K}$ as $\mathbf{A} = \mathbf{LL}^T$ in $\frac{K^3}{3}$ operations, and to solve the linear inverse problem for n vectors it needs $2nK^2$ operations, which sum up to $2nK^2 + \frac{1}{3}K^3$ operations [38]. Thus the total flop count for MOD will be

$$\mathcal{T}_{\text{MOD}} = 2nmN + 2m^2N + 2nK^2 + \frac{K^3}{3} - nK - K^2. \quad (\text{Eq. 3.12})$$

¹ $\mathbf{S}^{(t)}\mathbf{S}^{(t)T}$ can be positive semi definite if any atom from $\mathbf{D}^{(t)}$ is completely unused. In that case, we can remove those atoms from $\mathbf{D}^{(t)}$ and the corresponding row from the sparse representation matrix.

3.5.4 SGK

Similarly, for SGK we need $2n(m-1)|R_k^{(t)}|$ operations to compute $\mathbf{E}_k^{(t)}$, $n(2|R_k^{(t)}| - 1)$ operations are needed to compute $\mathbf{E}_k^{(t)} \mathbf{S}_k^{(t)T}$, approximately $2|R_k^{(t)}| - 1$ operations are needed to compute $\mathbf{S}_k^{(t)} \mathbf{S}_k^{(t)T}$, and n operations are needed for the division. This gives a total of $2nm|R_k^{(t)}| + 2|R_k^{(t)}| - 1$ operations needed to update one atom in $\mathbf{D}^{(t)}$. Thus the total flops required for SGK will be the sum over all K atoms,

$$\mathcal{T}_{\text{SGK}} = 2nm^2N + 2mN - K. \quad (\text{Eq. 3.13})$$

3.5.5 Comparison

The complexity expressions give a sense that MOD is the least complex, which contains only 3rd order terms. However for a fair comparison, let's express all the variables in terms of K . In general, the signal dimension $n = O(K)$, and the number of training samples $N = O(K^{1+a})$, where $a \geq 0$. Therefore, a condition for minimum complexity may be derived by taking sparsity $m = O(K^b)$. It can be found that $\arg \min_{a,b} \mathcal{T}_{K\text{-SVD}} = O(K^4)$, and $\arg \min_{a,b} \mathcal{T}_{\text{MOD}} = O(K^3)$, whereas $\forall_{b \geq 0} \mathcal{T}_{K\text{-SVDa}} = \mathcal{T}_{\text{SGK}} = O(K^{2+2b+a})$. Thus MOD remains least complex as long as $b \geq 0.5(1-a)$, and this dimensionality condition is very likely in practical situations. Therefore it can safely be stated, $\mathcal{T}_{\text{MOD}} \leq \mathcal{T}_{\text{SGK}} < \mathcal{T}_{K\text{-SVDa}} \ll \mathcal{T}_{K\text{-SVD}}$. Alongside, the execution time of all algorithms in Matlab environment² is compared in Table 3.1, for $n = 20, K = 50, N = 1500$, and various m , which agrees with the above analysis. It also reflects that being a parallel update procedure, MOD's execution time reduces by a factor of $O(K)$.

3.6 Synthetic Experiment

Similar to [9], K -SVD, approximate K -SVD, MOD and the sequential generalization are applied on synthetic signals. The purpose is to test how well these algorithms recover

²Matlab was running on a 64 bit OS with 8GB memory and 3.1GHz CPU.

Table 3.1: comparison of execution time (in millisecond)

m	$\mathcal{T}_{K\text{-SVD}}$	$\mathcal{T}_{K\text{-SVDa}}$	$\mathcal{T}_{\text{MOD}}$	$\mathcal{T}_{\text{SGK}}$
3	148.86	12.35	0.52	4.31
4	158.76	13.77	0.66	5.21
5	166.33	15.26	0.76	6.32

the original dictionary that generated the signal.

3.6.1 Training Signal Generation

A matrix $\mathbf{D}$ (later referred as generating dictionary) of size 20×50 is generated, whose entries are uniform i.i.d. random variables. As K -SVD can only operate on a normalized dictionary, each column is normalized to unit ℓ_2 norm. Then, 1500 training signals $\{\mathbf{x}_i\}_{i=1}^{1500}$ of dimension 20 are generated by a linear combination of m atoms at random locations with i.i.d. coefficients. In order to check the robustness of the algorithms, additive white Gaussian noises are added to the resulting training signals. The additive noises are scaled accordingly to obtain equal signal to noise (SNR) ratio across the training signals.

3.6.2 Dictionary Design

In all the algorithms, the dictionaries are initialized with the same set of K training signals selected at random. As per the suitability of K -SVD, an *unconstrained* sparse coding is done using orthogonal matching pursuit (OMP), which produces best m -term approximation for each signal [15]. All dictionary training algorithms are iterated $9m^2$ times for sparsity level m .

3.6.3 Results

The trained dictionaries are compared against the known generating dictionary in the same way as in [9]. The mean number of atoms retrieved over 50 trials are computed

Table 3.2: Average no. of atoms retrieved by dictionary training

	10 dB	20 dB	30 dB	No Noise	
<i>K</i> -SVD	36.88	46.48	46.94	47.06	$m = 3$
<i>K</i> -SVD _a	36.86	46.28	46.68	46.90	
MOD	36.60	46.00	45.86	46.52	
SGK	36.24	45.66	46.08	46.92	
<i>K</i> -SVD	17.46	47.18	47.10	47.04	$m = 4$
<i>K</i> -SVD _a	16.88	46.34	46.63	46.98	
MOD	18.20	45.88	46.24	46.36	
SGK	18.44	46.76	46.82	47.20	
<i>K</i> -SVD	00.88	45.72	47.04	46.90	$m = 5$
<i>K</i> -SVD _a	00.68	45.98	47.20	47.18	
MOD	00.76	45.86	46.38	46.88	
SGK	00.98	46.52	46.50	46.76

 Figure 3.2: Average number of atoms retrieved after each iteration for different values of m at $\text{SNR} = \infty$ dB

Figure 3.3: Average number of atoms retrieved after each iteration for different values of m at SNR = 30 dB

Figure 3.4: Average number of atoms retrieved after each iteration for different values of m at SNR = 20 dB

Figure 3.5: Average number of atoms retrieved after each iteration for different values of m at SNR = 10 dB

for each algorithm at different sparsity levels $m = 3, 4, 5$; with additive noise SNR = 10, 20, 30, ∞ dB. The results are tabulated in Table 3.2, which shows marginal difference among all the algorithms. In order to show convergence of the algorithms, the average number of atoms retrieved after each iteration is shown in Fig. 3.2-3.5.

Given their comparable performance but differing complexity, it may be concluded that MOD is the better choice for dictionary training. However, sequential update become essential to deal with high storage memory demanding larger data sets, which makes SGK the algorithm of choice for dictionary training. Moreover, SGK's update procedure only includes weighted averaging of vectors, which is a much stable procedure compared to MOD's generalized matrix inversion. The advantage of both MOD and SGK is that they can be used in sparse representation applications, irrespective of *constraints* on dictionary and sparse coder.

3.7 Discussions

Existing dictionary training algorithms MOD and K -SVD are presented in line with K -means clustering for VQ. It is shown that MOD simplifies to K -means, while K -SVD fails to simplify due to its principle of updating. As MOD does not need to update the sparse representation vector during dictionary update stage, it is compatible to any *structured/constrained* sparsity model such as K -means. However, MOD is not sequential and it involves an unstable generalized matrix inversion step. Hence, a sequential generalization to K -means is proposed that avoids the difficulties of K -SVD and MOD. Computational complexity for all algorithms are derived, and MOD is shown to be the least complex followed by SGK. Experimental results show that all the algorithms are performing equally well with marginal differences. Thus, MOD being the fastest among all, it remains the dictionary training algorithm of choice for any kind of sparse representation. However, if sequential update becomes essential, SGK should be chosen.

3.8 Summary

Two important dictionary training algorithms, MOD and K -SVD are analyzed in a common platform. It is demonstrated that K -SVD does not preserve any additional *structure/constraint* imposed into the sparse coefficient. As a result of which, it does not simplify to K -means in the case of VQ. It is also shown that MOD can preserve additional *structure/constraint* imposed into the sparse coefficient. As a result of which, it simplifies to K -means in the case of VQ. A new dictionary training algorithm called SGK is proposed as a sequential alternative to MOD. The computational complexities for all the three algorithms, K -SVD, MOD and SGK, are analyzed and compared. It is shown that MOD is least complex followed by SGK. Since MOD is a resource hungry parallel update procedure, SGK should be chosen as the sequential alternative.

Chapter 4

Applications of Trained Dictionary

This chapter intends to illustrate some interesting applications of trained dictionary for image processing, in particular image compression, inpainting and denoising. Dictionary training produces a set of signal prototype which can well describe the training signals. Therefore, to make an effective use of dictionary training, it is better to have the training samples from the same class as the test signals. A dictionary trained on a narrower class of signal will perform better, which can also be observed from the image denoising experiments of [11]. The dictionary trained on the image blocks extracted from a global class of images performs inferior denoising compared to the dictionary trained on the image blocks extracted from the noisy image itself. Thus, the applications are evaluated on single class databases such as face or car. In this chapter, an extensive comparison is made between SGK and K -SVD through the problems of image processing. In the previous chapter, through synthetic data experiments, it has been shown that the dictionary adaptation performances of K -SVD and SGK are comparable. Analytically it has also been shown that SGK has a superior execution speed in comparison to K -SVD, and it is advantageous to use SGK. Through this chapter, these claims are also verified in practical circumstances.

4.1 Image Compression

Similar to JPEG image compression, the goal is to compress an image X in its transform domain. Here transform domain means explicit sparse representation on a overcomplete dictionary. In order to simplify the transform coding, the image is divided into smaller blocks of size $\sqrt{n} \times \sqrt{n}$ (similar to JPEG, where 8×8 blocks are used). Then the obtained sparse representation is encoded for each block. Hence, sparser representation in transform domain results in better compression. The trained dictionaries are expected to compress better than the traditional dictionaries, because the goal of dictionary training is to minimize the sparser representation error by adapting to the training signals. Here, the objective is to show that with its swift execution speed, SGK can perform energy compaction as effectively as K -SVD.

For simplification, all the sparse representations of columnized image blocks $\mathbf{x} \in \mathbb{R}^n$ are obtained on dictionary $\mathbf{D}$ containing columnized two dimensional (2-D) atoms. However, we can rearrange them into 2-D shapes for visualization. The sparse representation is obtained as follows,

$$\hat{\mathbf{s}} = \arg \min_{\mathbf{s}} \|\mathbf{s}\|_0 \text{ such that } \|\mathbf{x} - \mathbf{D}\mathbf{s}\|_2^2 \leq \epsilon^2, \quad (\text{Eq. 4.1})$$

where ϵ is the error control parameter. In order to control the compression ratio or the bits per pixel (BPP), a fixed bits per coefficient q is allocated, and the coefficients are quantized uniformly as $Q(\hat{\mathbf{s}})$. It is clear from equation (Eq. 4.1) that higher value of ϵ leads to lesser number of nonzero coefficients $\|\hat{\mathbf{s}}\|_0$. Hence, a desired BPP can be obtained by controlling the representation root mean square error ϵ .

BPP of any compression scheme depends on the amount of information required to be stored, so that we can recover back the compressed image. In this scheme of compression, the following necessary informations are needed to be coded[9].

- The number of coefficients in each block (a bits are allocated to store it)
- The corresponding index of the coefficients (b bits are allocated to store each index)
- The coefficients (q bits are allocated to store each coefficient)

The value of a and b can be chosen based on the maximum values of the corresponding informations, and a suitable uniform quantization step size for Q can be obtained by checking the extreme values of the coefficients. The BPP is computed as follows,

$$\text{BPP} = \frac{a \cdot \# \text{blocks} + (b + q) \cdot \# \text{coefs}}{\# \text{pixels}}, \quad (\text{Eq. 4.2})$$

where $\# \text{blocks}$ is the number of blocks in a image, $\# \text{coefs}$ is the total number of coefficients used to represent the image, and $\# \text{pixels}$ stands for total number of pixels in the image.

4.1.1 Compression Experiments

The image compression experiment is performed on Yale face database and MIT car database. 39 face images of size 192×168 , and 39 car images of size 128×128 are taken. For each data base images are divided in to two sets, one is training set that contains 19 images, and another is test set that contains 20 images. The images in training set are used for dictionary training, and the images in the test set are used to evaluate the performance dictionaries. Blockwise transform coding is performed on the test images for blocks of size 8×8 . Including a sign bit, 7 bits per coefficient ($q = 7$) are allocated to quantize the coefficients uniformly. The quantization step size depends on the range of the coefficients for each instance of image compression. Similarly, a and b of equation (Eq. 4.2) are obtained on each instance of image compression. BPP of the compressed images are computed as describe in (Eq. 4.2). The image $\hat{X}$ is restored by restoring each

Figure 4.1: The dictionaries of atom size 8×8 trained on the 19 sample images, starting with overcomplete DCT as initial dictionary.

image block $\hat{\mathbf{x}} = \mathbf{D}\mathbf{Q}(\hat{\mathbf{s}})$, and the compressed image quality is verified using peak signal to noise ratio,

$$\text{PSNR} = 20 \log_{10} \left(\frac{255}{\|\mathbf{X} - \hat{\mathbf{X}}\|_2} \right).$$

All the sparse coding in this experiments are done using orthogonal matching pursuit (OMP). Note that a better performance can be obtained by switching to a better pursuit algorithm to find a sparse solution, e.g. FOCUSS. However, OMP is emphasized due to its simplicity and fast execution.

A set of 8×8 training blocks are extracted from the first 19 face images. Two separate dictionaries are trained as described in the previous chapter, one using K -SVD update step and another using SGK. 32 iterations are used for the dictionary training algorithms to converge. Similar to [9], the first dictionary element is denoted as the DC, which contains a constant value in all of its entries and never updated afterwards. Since, the DC takes part in all representations, all other dictionary elements remain with zero mean after all iterations. In the sparse coding stage of the dictionary training, the sparse representation is obtained for each training signal as

$$\hat{\mathbf{s}} = \arg \min_{\mathbf{s}} \|\mathbf{x} - \mathbf{D}\mathbf{s}\|_2^2 \text{ such that } \|\mathbf{s}\|_0 = m_0, \quad (\text{Eq. 4.3})$$

where $m_0 = 10$ [9]. For this scenario of dictionary training, the execution time is compared in Table 4.1, which is in accordance with the complexity analysis of the previous chapter. The trained dictionaries are displayed in Figure 4.1.

Table 4.1: Comparison of execution time in seconds for one iteration of dictionary update (Compression). Boldface is used for the better result.

	K -SVD	SGK
Face database	1.674	0.166
Car database	2.160	0.267

Figure 4.2: Visual comparison of compression results of sample images.

The image compression results are obtained for all three dictionaries: overcomplete DCT, *K*-SVD and SGK. Similar to the experimental set up of [9], the dictionaries carry 441 number of atoms. Various BPP can be obtained by varying the value of ϵ in (Eq. 4.1). Hence, using the obtained dictionaries, an average rate-distortion (R-D) plot is generated over the remaining 20 images, and presented in Figure 4.3. In order to have a visual comparison, one compressed image from each database is shown in Figure 4.2. The compression results confirms the competency of SGK over *K*-SVD, by showing its superior execution speed with at par energy compaction.

Figure 4.3: Compression results: rate-distortion plot.

4.2 Image Inpainting

In the problem of image inpainting, the missing pixels of an image are needed to be filled up. The corrupted images with missing pixels can be modeled as

$$Y = B \circ X,$$

where an image X is element-wise multiplied with a binary mask B . This problem is handled in the same manner as it is done for image compression, that is by dividing the image into small blocks of size $\sqrt{n} \times \sqrt{n}$. Thus, the missing pixels of these small $\sqrt{n} \times \sqrt{n}$ images are needed to be filled up individually.

Let's denote $\mathbf{x} \in \mathbb{R}^n$ as a columnized image block, and $\mathbf{b} \in \{0, 1\}^n$ be the corresponding binary mask, then the individual corrupt image blocks can be presented as $\mathbf{y} = \mathbf{b} \circ \mathbf{x}$. It is known that it is possible to represent $\mathbf{x} = \mathbf{D}\mathbf{s}$ in a suitable dictionary $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_K]$ as per the standard notations, where $\mathbf{s} \in \mathbb{R}^K$ is sparse (i.e. $\|\mathbf{s}\|_0 \ll n$). Hence, it is assumed that $\mathbf{y}$ also has the same sparse representation $\mathbf{s}$ in

$$[(\mathbf{b}1_K^T) \circ \mathbf{D}] = [\mathbf{b} \circ \mathbf{d}_1, \mathbf{b} \circ \mathbf{d}_2, \dots, \mathbf{b} \circ \mathbf{d}_K],$$

where 1_K is a vector containing K ones. Therefore, a dictionary $\mathbf{D}$ is taken, and estimate the sparse representation $\mathbf{s}$ for each corrupt image block as follows.

$$\hat{\mathbf{s}} = \arg \min_{\mathbf{s}} \|\mathbf{s}\|_0 \text{ such that } \|\mathbf{y} - [(\mathbf{b}1_K^T) \circ \mathbf{D}]\mathbf{s}\|_2^2 \leq \epsilon^2, \quad (\text{Eq. 4.4})$$

where ϵ is the allowed representation error. After obtaining $\hat{\mathbf{s}}$, the image block is restored as $\hat{\mathbf{x}} = \mathbf{D}\hat{\mathbf{s}}$.

4.2.1 Inpainting Experiments

Using the above framework, the performances of the trained dictionaries are compared. Similar to the previous section, taking the same training set, dictionaries are trained.

Table 4.2: Comparison of execution time in seconds for one iteration of dictionary update (Inpainting). Boldface is used for the better result.

	<i>K</i> -SVD	SGK
Face database	2.042	0.253
Car database	1.732	0.164

Table 4.3: Comparison of average PSNR of the reconstructed test images in dB, at various percentage of missing pixel. Boldface is used for the better result.

	30%	40%	50%	60%	70%	80%	90%	
Face database	33.84	32.45	30.90	29.07	26.79	23.33	15.46	DCT
	35.39	34.41	33.11	31.51	29.04	25.60	16.18	<i>K</i> -SVD
	35.42	34.37	33.01	31.38	29.27	25.55	16.23	SGK
Car database	29.96	27.66	25.82	23.85	21.73	19.27	13.79	DCT
	33.36	31.26	29.06	26.98	24.33	20.89	14.14	<i>K</i> -SVD
	33.30	31.17	29.23	26.86	24.57	20.76	14.20	SGK

However, in the sparse coding stage, the sparse coder (Eq. 4.3) is used with $m_0 = 5$. Similar to [9], the problem of pixels missing at random locations is only considered. Thus, two test images are taken from the images that are not used for dictionary training. 50% of pixels at random locations are set to 0 for first image, and 70% of pixels are set to 0 for the second image. Each image is divided using 8×8 blocks, which makes the signal length $n = 64$. For each image block, OMP is used to solve equation (Eq. 4.4) by setting $\epsilon = 3\sqrt{n}$, which means a maximum error of ± 3 gray levels is allowed in the reconstruction. Similar to the previous section, three set of results are obtained for overcomplete DCT, *K*-SVD, and SGK for all 20 test images. To have a visual comparison of the inpainting performance of the dictionaries, one inpainted image from each database is shown in Figure 4.4. To have an extensive comparison, the average PSNR over the test images for various percentage of missing pixel is presented in Table 4.3. These results prove that SGK is as promising as *K*-SVD also in the case of image inpainting. In addition to this, SGK has a superior execution speed, which can be verified from Table 4.2.

A sample face image.

A sample car image.

Figure 4.4: The corrupted image (where the missing pixels are blackened), and the reconstruction results using overcomplete DCT dictionary, *K*-SVD trained dictionary, and SGK trained dictionary, respectively. The first row is for 50% missing pixels, and the second row is for 70% missing pixels.

4.3 Image Denoising

Image denoising is a classical problem. Over the past 50 years, it has been addressed from numerous points of view. In this inverse problem, an unknown image X of dimension $\sqrt{N} \times \sqrt{N}$ is contaminated with Additive White Gaussian Noise (AWGN) $V \in \mathbb{R}^{\sqrt{N} \times \sqrt{N}}$ resulting in the measured image

$$Y = X + V.$$

The aim is to obtain $\hat{X}$ - a close estimation of X , in the sense of Euclidean distance. In this piece of work, the image denoising problem is addressed from the sparse representation point of view.

With explicit use of sparse representation, a framework for image denoising was first illustrated in [11]. The key idea is to obtain a global denoising of the image by denoising overlapped local image blocks. Let's define $\mathbf{R}_{ij}$ as an $n \times N$ matrix that extracts a $\sqrt{n} \times \sqrt{n}$ block $\mathbf{x}_{ij}$ from the columnized image X starting from its 2D coordinate (i, j) ¹. By sweeping across the coordinates (i, j) of X , overlapping local patches can be extracted as $\{\forall_{ij} \mathbf{x}_{ij} = \mathbf{R}_{ij}X\}$. It is assumed that there exists a sparse representation for any columnized image block $\mathbf{x} \in \mathbb{R}^n$ on a suitable dictionary $\mathbf{D} \in \mathbb{R}^{n \times K}$. That is,

$$\begin{aligned} \hat{\mathbf{s}} &= \arg \min_{\mathbf{s}} \|\mathbf{s}\|_0 \quad \text{such that} \quad \|\mathbf{x} - \mathbf{D}\mathbf{s}\|_2^2 \leq \epsilon^2 \\ &= \arg \min_{\mathbf{s}} \left\{ \mu \|\mathbf{s}\|_0 + \|\mathbf{x} - \mathbf{D}\mathbf{s}\|_2^2 \right\} \end{aligned}$$

where ϵ is the representation error tolerance, and μ is the local Lagrangian multiplier based on the value of ϵ , for which these two minimization problems become the same. Similarly, it can be extended to all the image blocks,

$$\forall_{ij} \quad \hat{\mathbf{s}}_{ij} = \arg \min_{\mathbf{s}_{ij}} \left\{ \mu_{ij} \|\mathbf{s}_{ij}\|_0 + \|\mathbf{R}_{ij}X - \mathbf{D}\mathbf{s}_{ij}\|_2^2 \right\}, \quad (\text{Eq. 4.5})$$

¹Basically, $\mathbf{R}_{ij}$ can be viewed as a matrix, which contains n selected rows of an $N \times N$ identity matrix $\mathbf{I}_N$. Hence it picks n elements from an N dimensional vector.

where μ_{ij} is location dependent.

The global recovery of the image from these local representations is formulated using a maximum a posteriori probability (MAP) estimate in [11],

$$\{\hat{X}, \forall_{ij} \hat{\mathbf{s}}_{ij}\} = \arg \min_{X, \forall_{ij} \mathbf{s}_{ij}} \left\{ \lambda \|Y - X\|_2 + \sum_{ij} \mu_{ij} \|\mathbf{s}_{ij}\|_0 + \sum_{ij} \|\mathbf{R}_{ij}X - \mathbf{D}\mathbf{s}_{ij}\|_2 \right\}. \quad (\text{Eq. 4.6})$$

The first term in (Eq. 4.6) is the log-likelihood that demands the closeness between the measured image Y and its estimated (and unknown) version X . This shows the direct relationship between λ and $E[V^2(i, j)] = \sigma^2$. In this denoising framework, the noise variance σ^2 is known *a priori*.

The solution to estimate (Eq. 4.6) is obtained in two steps. First, all the local sparse representations are obtained as per equation (Eq. 4.5). Since X is unknown, the sparse representations are estimated by treating Y as X ,

$$\forall_{ij} \quad \hat{\mathbf{s}}_{ij} = \arg \min_{\mathbf{s}_{ij}} \|\mathbf{s}_{ij}\|_0 \text{ s.t. } \|\mathbf{R}_{ij}Y - \mathbf{D}\mathbf{s}_{ij}\|_2^2 \leq \epsilon_{ij}^2.$$

Assuming the uniformity of the noise, the values of ϵ_{ij} can be set equal, to an appropriate value based on noise variance σ^2 ². Note that a better sparse solution will lead to a better denoising performance. In the experiments, Orthogonal Matching Pursuit (OMP) is used, due to its simple implementation and sure convergence quality [15].

After estimating $\{\forall_{ij} \hat{\mathbf{s}}_{ij}\}$, the denoised image blocks are obtained as $\{\forall_{ij} \hat{\mathbf{x}}_{ij} = \mathbf{D}\hat{\mathbf{s}}_{ij}\}$. Then the final denoised image $\hat{X}$ is derived from the reduced MAP estimator, i.e.

$$\begin{aligned} \hat{X} &= \arg \min_X \left\{ \lambda \|Y - X\|_2 + \sum_{ij} \|\mathbf{R}_{ij}X - \mathbf{D}\hat{\mathbf{s}}_{ij}\|_2 \right\} \\ &= \arg \min_X \left\{ \lambda \|Y - X\|_2 + \sum_{ij} \|\mathbf{R}_{ij}X - \hat{\mathbf{x}}_{ij}\|_2 \right\}. \end{aligned} \quad (\text{Eq. 4.7})$$

² $\forall_{ij} \epsilon_{ij}^2 = \epsilon^2 = n(1.15 \times \sigma)^2$ is used in [11].

There exists a closed form solution to the above minimization problem, i.e.

$$\hat{X} = \left(\lambda \mathbf{I}_N + \sum_{ij} \mathbf{R}_{ij}^T \mathbf{R}_{ij} \right)^{-1} \left(\lambda Y + \sum_{ij} \mathbf{R}_{ij}^T \hat{\mathbf{x}}_{ij} \right), \quad (\text{Eq. 4.8})$$

where $\mathbf{R}_{ij}^T$ is the transpose of the matrix $\mathbf{R}_{ij}$ that places back the image block into the coordinate (i, j) of a blank image, which is in columnized form of $N \times 1$. This cumbersome expression means that averaging of the denoised image blocks is to be done, with some relaxation obtained from the noisy image. Hence $\lambda \propto \frac{1}{\sigma}$, which decides to what extent the noisy image can be trusted. The matrix to invert in (Eq. 4.8) is a diagonal matrix, hence the calculation of the above expression can be done on a pixel-by-pixel basis, after $\{\forall_{ij} \hat{\mathbf{x}}_{ij}\}$ is obtained.

Apart from this formulation, the main ingredient of [11] was the use of trained dictionary $\mathbf{D}$. It has shown that K -SVD dictionary trained on the noisy image blocks gives outstanding denoising performance compared to traditional dictionaries (e.g. over-complete DCT). Hence, it has motivated many extensions and enhancements; e.g. color image restoration [40], video denoising [41], multi-scale dictionary [42], and adaptive local window selection for sparse representation [Chapter 5 of the thesis].

4.3.1 Dictionary Training on Noisy Images

It is known from the previous chapter that K -SVD is a computationally demanding algorithm, and a faster dictionary training algorithm, SGK, is proposed. In this piece of work, it is shown that K -SVD can be substituted with SGK in the denoising framework of [11] because its outcomes are indifferent to K -SVD with noticeable gain in speed. Similarly, SGK can also be substituted in the extensions and enhancements of this denoising framework including [40], [41] and [42].

The MAP estimation equation (Eq. 4.6) assumes that $\mathbf{D}$ is known *a priori*. Thus, the solution is obtained in two steps: first compute $\{\forall_{ij} \hat{\mathbf{s}}_{ij}\}$ by taking $X = Y$, and then

compute $\hat{X}$ using (Eq. 4.8). However, a quest for a better dictionary $\hat{\mathbf{D}}$ can also be incorporated into the MAP expression,

$$\begin{aligned} \{\hat{X}, \hat{\mathbf{D}}, \mathbf{s}_{ij}\} = \arg \min_{X, \mathbf{D}, \mathbf{s}_{ij}} & \left\{ \lambda \|Y - X\|_2 + \sum_{ij} \mu_{ij} \|\mathbf{s}_{ij}\|_0 \right. \\ & \left. + \sum_{ij} \|\mathbf{R}_{ij}X - \mathbf{D}\mathbf{s}_{ij}\|_2 \right\}. \end{aligned} \quad (\text{Eq. 4.9})$$

Like in [11], it is going to be a two stage iterative process; a *sparse coding* stage followed by a *dictionary update* stage. Hence, $X = Y$ is taken and an initial dictionary $\mathbf{D}$. A set of training signals $\mathbf{X}$ is obtained by sweeping $\mathbf{R}_{ij}$ across the coordinates of X . Though K -SVD was explicitly used for dictionary training in [11], here it is compared with SGK.

4.3.2 Denoising Experiments

This subsection demonstrates the results obtained by applying the discussed framework on several test images, in the case of both K -SVD and SGK trained dictionaries. For a fair comparison, the test images, as well as the tested noise levels, are kept the same as those used in the experiments reported in [11].

Table 4.4 summarizes the denoising results for both K -SVD and SGK trained dictionaries. Table 4.5 shows the time taken to obtain the trained dictionaries. In this set of experiments, the dictionaries used were of size 64×256 (that is $n = 64, K = 256$), and extracted image blocks are of size 8×8 pixels. All the tabulated figures are an average over 5 experiments of different noise realizations. The overcomplete DCT dictionary that was used as the initialization for both the training algorithms, is shown on the extreme left of Figure 4.6, and each of the atoms occupy a cell of 8×8 pixel image.

All the experiments include a sparse coding of each image block of size 8×8 pixels from the noisy image, where OMP is used to accumulate the atoms till the average error

Task: Denoise a given image Y contaminated with additive white Gaussian noise of variance σ^2 . In other words, to solve

$$\{\hat{X}, \hat{\mathbf{D}}, \forall_{ij} \hat{\mathbf{s}}_{ij}\} = \arg \min_{\mathbf{X}, \mathbf{D}, \forall_{ij} \mathbf{s}_{ij}} \left\{ \lambda \|Y - X\|_2 + \sum_{ij} \mu_{ij} \|\mathbf{s}_{ij}\|_0 + \sum_{ij} \|\mathbf{R}_{ij}X - \mathbf{D}\mathbf{s}_{ij}\|_2 \right\}.$$

Input Parameters: block size- n , number of atoms- K , number of dictionary training iterations- J , Lagrangian multiplier- λ , and error threshold- ϵ .

Output: denoised image- $\hat{X}$, trained dictionary- $\hat{\mathbf{D}}$

Procedure:

(i) *Initialization:* Set $X = Y$, $\mathbf{D}$ = overcomplete DCT dictionary.

(ii) *Dictionary Training:* Repeat J times

- *Sparse Coding Stage:* Using any sparse pursuit algorithm, compute the representation vector $\mathbf{s}_{ij}$ for each extracted image block $\mathbf{R}_{ij}X$, which estimates the solution of

$$\hat{\mathbf{s}}_{ij} = \arg \min_{\mathbf{s}_{ij}} \|\mathbf{s}_{ij}\|_0 \text{ s.t. } \|\mathbf{R}_{ij}X - \mathbf{D}\mathbf{s}_{ij}\|_2^2 \leq \epsilon^2.$$

- *Dictionary Update Stage:* By sweeping $\mathbf{R}_{ij}$ across the coordinates of X , obtain the set of training signals $\mathbf{X}$ and corresponding sparse representations $\mathbf{S}$. Update $\mathbf{D}$ either by SVD or by SGK formulation [Chapter 3 of the thesis].

(iii) *Final Denoising:*

- Using the obtained K -SVD or SGK trained dictionary $\hat{\mathbf{D}}$, estimate the final sparse representation vector $\hat{\mathbf{s}}_{ij}$ for each extracted image block $\mathbf{R}_{ij}X$.

$$\hat{\mathbf{s}}_{ij} = \arg \min_{\mathbf{s}_{ij}} \|\mathbf{s}_{ij}\|_0 \text{ s.t. } \|\mathbf{R}_{ij}X - \hat{\mathbf{D}}\mathbf{s}_{ij}\|_2^2 \leq \epsilon^2.$$

- Estimate

$$\hat{X} = \left(\lambda \mathbf{I}_N + \sum_{ij} \mathbf{R}_{ij}^T \mathbf{R}_{ij} \right)^{-1} \left(\lambda Y + \sum_{ij} \mathbf{R}_{ij}^T \hat{\mathbf{D}} \hat{\mathbf{s}}_{ij} \right)$$

Figure 4.5: Image denoising using a dictionary trained on the noisy image blocks. The experimental results are obtained with $J = 10$, $\lambda = 30/\sigma$, $\epsilon^2 = n(1.15\sigma)^2$, and OMP.

passed the threshold $(1.15 \times \sigma)^2$ ³ [21]. The denoised blocks were averaged, as described in (Eq. 4.8), using $\lambda = 30/\sigma$ as in [11]. The dictionary is trained on overlapping image blocks extracted from the noisy image itself. In each such experiment, all available image blocks are included for dictionary training in the case of 256×256 images, and every second image block from every second row in the case of 512×512 images. The algorithm described in Figure 4.5 was applied on the test images once using K -SVD dictionary update step, and again using SGK dictionary update step.

It can be seen from Table 4.4 that the results of all methods are indifferent to each other in general. Table 4.5 shows the faster execution of SGK, which is approximately 4 times faster than K -SVD. It can also be noticed that the computation time for all the images are reducing with the noise level, because at higher noise level image blocks are represented with lesser number of coefficients, to avoid the noise getting into estimation. Hence, the required number of computations reduces, which depends on the number of coefficients m [Chapter 3 of the thesis].

Figure 4.7 shows the denoised images using both the dictionaries for the image Barbara at $\sigma = 20$. The final trained dictionaries that lead to those results are presented in Figure 4.6.

4.4 Discussions

The previous chapter's synthetic data experiment only validates that SGK converges as good as K -SVD to a unique dictionary. Hence, through the described framework of image compression, the advantage of SGK over K -SVD is highlighted. Though, the intention is not to propose any new image compression framework, certain things can be optimized for a better compression. For simplicity, a uniform quantization of the

³This value was empirically chosen in [11].

Figure 4.6: The dictionaries trained on Barbara image at $\sigma = 20$ —initial dictionary, K -SVD trained dictionary, and SGK trained dictionary.

Figure 4.7: The denoising results for the Barbara image at $\sigma = 20$ —the original, the noisy, and restoration results using the two trained dictionaries.

Table 4.4: Comparison of the denoising PSNR results in dB. In each cell two denoising results are reported. Left: using K -SVD trained dictionary. Right: using SGK trained dictionary. All numbers are an average over five trials. The last two columns present the average result and their standard deviation over all images. Boldface is used for the better result.

σ	Lena		Barb		Boats		Fgrpt		House		Peppers		Average		σ_{PSNR}	
	K -SVD	SGK	K -SVD	SGK	K -SVD	SGK	K -SVD	SGK	K -SVD	SGK	K -SVD	SGK	K -SVD	SGK	K -SVD	SGK
2	43.35	43.35	43.34	43.34	42.96	42.96	42.87	42.86	44.50	44.49	43.33	43.33	43.39	43.39	0.02	0.02
5	38.21	38.21	37.65	37.65	37.00	37.00	36.51	36.51	39.43	39.43	37.89	37.88	37.78	37.78	0.02	0.02
10	35.06	35.04	33.94	33.93	33.39	33.39	32.21	32.21	35.96	35.94	34.25	34.25	34.13	34.12	0.02	0.02
15	33.25	33.23	31.96	31.93	31.47	31.45	29.83	29.83	34.29	34.26	32.19	32.17	32.16	32.15	0.02	0.02
20	31.92	31.89	30.44	30.42	30.10	30.09	28.21	28.20	33.17	33.13	30.77	30.75	30.77	30.75	0.04	0.04
25	30.87	30.85	29.28	29.26	29.03	29.01	27.01	27.00	32.08	32.05	29.73	29.69	29.67	29.64	0.03	0.03
50	27.35	27.35	25.23	25.22	25.65	25.63	23.02	23.01	28.08	28.07	26.17	26.15	25.92	25.90	0.06	0.06
75	25.29	25.29	22.79	22.79	23.71	23.70	19.86	19.85	25.24	25.25	23.59	23.60	23.41	23.41	0.09	0.09
100	23.91	23.93	21.65	21.66	22.45	22.46	18.25	18.24	23.63	23.65	21.87	21.88	21.96	21.97	0.04	0.04

Table 4.5: Comparison of execution time in seconds. Left: K -SVD training time. Right: SGK training time. Boldface is used for the better result.

Lena		Barb		Boats		Fgrpt		House		Peppers		Average	
σ /PSNR	K -SVD	SGK	K -SVD	SGK	K -SVD	SGK	K -SVD	SGK	K -SVD	SGK	K -SVD	SGK	SGK
2/42.11	12.384	2.952	17.038	3.873	17.699	4.155	23.171	5.214	10.176	2.405	16.439	3.825	16.151
5/34.15	5.225	1.324	8.548	1.975	8.128	1.949	12.750	2.738	4.518	1.110	7.636	1.728	7.801
10/28.13	3.065	0.851	4.750	1.191	4.085	1.154	7.232	1.682	2.603	0.760	4.259	1.038	4.332
15/24.61	1.977	0.578	2.900	0.817	2.600	0.772	4.562	1.173	1.947	0.521	2.771	0.692	2.793
20/22.11	1.697	0.501	2.312	0.712	2.116	0.648	3.444	0.896	1.708	0.438	2.104	0.546	2.230
25/20.17	1.555	0.433	1.915	0.584	1.792	0.537	2.688	0.768	1.516	0.382	1.752	0.512	1.870
50/14.14	1.577	0.355	1.482	0.402	1.556	0.442	1.926	0.496	1.395	0.326	1.621	0.399	1.593
75/10.63	1.311	0.303	1.435	0.396	1.546	0.353	1.499	0.438	1.423	0.324	1.489	0.325	1.450
100/8.13	1.364	0.308	1.424	0.339	1.422	0.314	1.528	0.390	1.411	0.282	1.389	0.315	1.423

coefficients is used; and a simple coding is used to store the number of coefficients, the indices, and the coefficients. However, a better quantization strategy with entropy coding can further improve the compression ratio/BPP. Alongside, the described framework for image inpainting also validates the effectiveness of SGK.

To further validate the effectiveness of SGK in practice, it is incorporated into the framework of image denoising via sparse representation. SGK can be seen as a simpler and intuitive implementation compared to the use of K -SVD. The experimental results suggest that SGK performs as effectively as K -SVD, and needs lesser computations. Hence, K -SVD can be replaced with SGK in the image denoising framework, and all its extensions. Similarly, it is also possible to extend the use of SGK to other applications of sparse representation.

4.5 Summary

An image compression framework is illustrated, which codes the sparse representation coefficients of the non-overlapping image blocks like JPEG. An image inpainting framework is illustrated, which recovers the missing pixels of the non-overlapping image blocks by estimating their sparse representation from the available pixels. An image denoising framework is illustrated, which recovers the image by estimating the sparse representation of the overlapping image blocks. The estimated overlapping pixels are averaged to recover the image. Extensive comparisons are made between K -SVD and SGK using the above frameworks. It is shown that SGK is as effective as K -SVD in practice, where as SGK has the advantage of speed.

Chapter 5

Improving Image Recovery by Local Block Size Selection

In the previous chapter, the notion of image inpainting and denoising using sparse representation has been introduced, where the global image recovery is carried out through recovery of local image blocks. The two main reasons behind the use of local image blocks are the following - (i) the smaller blocks take lesser computation time and storage space; (ii) the smaller image blocks contain lesser diversity, hence it is easier to obtain a sparse representation with fewer coefficients. Though, how much smaller the block size will be is left to the user, it has an impact on the recovery performance. This impact is due to a change in image content inside a local block with a change in block size. Thus, it will be better, if we can find a suitable block size at each location that performs the optimal recovery of an image. Nevertheless, the task is challenging, because we don't have the original image to verify the recovery performance. The possibilities of numerous block sizes makes it even more complicated. In this chapter, a framework of block size selection is proposed, which bypasses these challenges. Essentially, possible window sizes are prefixed to a limited number, instead of dwelling around infinite possibilities. Next, a block size selection criterion is formulated that uses the corrupt image alone. Some background of block size selection is introduced in the next section, and in the subsequent

sections both the recovery frameworks (inpainting, denoising) is restated in conjunction with block size selection.

5.1 Local Block Size Selection

In order to simplify the global recovery problem, local recoveries are undertaken as small steps. In general, local block size selection plays an important role in the setup of local to global recovery. In the language of signal processing, this phenomenon of block size selection is often termed as bandwidth selection for local filtering. A natural question arises, that whether an optimal block size should be selected globally or locally. It is relatively easier to find a block size globally which yields the Minimum Mean Square Error (MMSE). Ideally, the optimal block size for local operation should be selected at each location of the image. This is because the global mean square error ($MSE = \frac{1}{N} \sum_{ij} [X(i, j) - \hat{X}(i, j)]^2$) is a collective contribution of the local mean square errors $\{\forall_{ij} MSE_{ij} = [X(i, j) - \hat{X}(i, j)]^2\}$, where X is the original image of size $\sqrt{N} \times \sqrt{N}$ and $\hat{X}$ is the recovered image. Thus, the optimal block size for a pixel location (i, j) is the one that gives minimum MSE_{ij} . In the absence of the original image X , this task becomes very challenging.

An earlier attempt towards adaptive block size selection can be found in [43], where each pixel is estimated pointwise using Local Polynomial Approximation (LPA). Increasing odd sized square blocks $n = n_1 < n_2 < n_3 < \dots$ were taken centering over each pixel (i, j) , and the best estimate is obtained as $\hat{X}^{\hat{n}}(i, j)$. The task is to find $\hat{n} = \arg \min_n MSE_{ij}^n = \arg \min_n [X(i, j) - \hat{X}^n(i, j)]^2$, where $\hat{X}^n(i, j)$ is the obtained polynomial approximation of the pixel $X(i, j)$ with block size $\sqrt{n} \times \sqrt{n}$. At each pixel (i, j) , a confidence interval $\mathcal{D}(n) = [L^n, U^n]$ is obtained for all the block sizes

$$n = n_1 < n_2 < n_3 < \dots,$$

$$\begin{aligned} L_n &= \hat{X}^n(i, j) - \gamma \cdot \text{std} \left(\hat{X}^n(i, j) \right), \\ U_n &= \hat{X}^n(i, j) + \gamma \cdot \text{std} \left(\hat{X}^n(i, j) \right), \end{aligned}$$

where γ is a fixed constant and $\text{std} \left(\hat{X}^n(i, j) \right)$ is the standard deviation of $\hat{X}^n(i, j)$ over different n . In order to find the Intersection of Confidence Intervals (ICI), the intervals $\forall_n \mathcal{D}(n)$ are arranged in the increasing order of local block size n . The first block size at which all the intervals intersect is decided as the optimal block $\hat{n}$. It is theoretically proven that ICI will often select the block size with minimum MSE_{ij}^n . However, the success of ICI is dependent on the accurate estimation of $\hat{X}^n(i, j)$ and its standard deviation $\text{std} \left(\hat{X}^n(i, j) \right)$. In addition, ICI has a drawback that it can only be applied to single pixel recovery framework. Since, more than one pixel of the estimated local blocks are used in the recovery frameworks, ICI will not help us selecting block size.

5.2 Inpainting using Local Sparse Representation

In this problem, an image $X \in \mathbb{R}^{\sqrt{N} \times \sqrt{N}}$ is being occluded by a mask $B \in \{0, 1\}^{\sqrt{N} \times \sqrt{N}}$, resulting in $Y = B \circ X$, where “ $\circ$ ” multiplies two matrices element wise.. The goal is to find $\hat{X}$ - the closest possible estimation of X . In the previous chapter, $\hat{X}$ has been obtained in a simple manner by estimating each non-overlapping local block, where the motive was only to show the competitiveness of SGK dictionary over K -SVD. However, a better inpainting result can obtained by considering overlapping local blocks. Thus, a block extraction mechanism is adapted based on the denoising framework of the previous chapter.

Here, blocks of size $\sqrt{n} \times \sqrt{n}$ having a center pixel are explicitly considered, which means $\sqrt{n}$ is an odd number. A $n \times N$ matrix, $\mathbf{R}_{ij}^n$ is defined, which extracts a $\sqrt{n} \times \sqrt{n}$

block $\mathbf{y}_{ij}^n$ from a $\sqrt{N} \times \sqrt{N}$ image Y as $\mathbf{y}_{ij}^n = \mathbf{R}_{ij}^n Y$, where the block is centered over the pixel (i, j) . Let's recall that Y , X and B are columnized to $N \times 1$ vector for this block extraction operation. Hence, sweeping across the 2D coordinates (i, j) of Y , overlapping image blocks can be extracted, i.e. $\forall_{ij} \{\mathbf{y}_{ij}^n = \mathbf{R}_{ij}^n Y\} \in \mathbb{R}^n$. The original image block is denoted as $\mathbf{x}_{ij}^n$, and the corresponding local mask as $\mathbf{b}_{ij}^n \in \{0, 1\}^n$, which makes the corrupt image block $\mathbf{y}_{ij}^n = \mathbf{x}_{ij}^n \circ \mathbf{b}_{ij}^n$.

Let $\mathbf{D}^n \in \mathbb{R}^{n \times K}$ be a known dictionary, where $\mathbf{x}_{ij}^n$ has a representation $\mathbf{x}_{ij}^n = \mathbf{D}^n \mathbf{s}_{ij}^n$, such that $\|\mathbf{s}_{ij}^n\|_0 \ll n$. Similar to the previous chapter, $\mathbf{s}_{ij}^n$ can be estimated as follows,

$$\hat{\mathbf{s}}_{ij}^n = \arg \min_{\mathbf{s}} \|\mathbf{s}\|_0 \quad \text{such that} \quad \|\mathbf{y}_{ij}^n - [(\mathbf{b}_{ij}^n 1_K^T) \circ \mathbf{D}^n] \mathbf{s}\|_2^2 \leq \epsilon^2(n),$$

where $\epsilon(n)$ is the representation error tolerance. To have equal error tolerance per pixel irrespective of the block size, $\epsilon(n) = 3\sqrt{n}$ is set for the experiment, which gives an error tolerance of 3 gray levels per pixel. Using the estimated sparse representations, the inpainted local image blocks are obtained as $\{\forall_{ij} \hat{\mathbf{x}}_{ij}^n = \mathbf{D}^n \hat{\mathbf{s}}_{ij}^n\}$. In spite of equal error tolerance per pixel, the estimation mean square error $(\frac{1}{n} \|\mathbf{x}_{ij}^n - \hat{\mathbf{x}}_{ij}^n\|_2^2)$ varies with block size n . It is because at some location, dictionary of some block size may fit better with the available pixels than another block size, which basically depends on the image content in that locality. Hence a MMSE based block size selection becomes essential.

5.2.1 Block Size Selection for Inpainting

The effect of block size is very perceptive in inpainting using local sparse representation. As bigger block sizes capture more details from the image, smaller block sizes are preferred for local sparse representation. However, bigger block sizes are suitable for inpainting as it is hard to follow the trends of the geometrical structures in small block sizes, even in visual perspective. So, there exists a trade-off between the block size and accuracy of

Figure 5.1: Block schematic diagram of the proposed image inpainting framework.

fitting. In the absence of the original image, some measure need to be derived to reach

$$\min_n MSE_{ij}^n = \min_n \frac{1}{n} \|\mathbf{x}_{ij}^n - \hat{\mathbf{x}}_{ij}^n\|_2^2. \quad (\text{Eq. 5.1})$$

In order to solve the aforementioned problem an approximation for MSE_{ij}^n is carried out. It is done by computing the MSE_{ij}^n for the observed pixels only. Thus, it can be written as

$$\widehat{MSE}_{ij}^n = \frac{1}{\mathbf{b}_{ij}^{nT} \mathbf{b}_{ij}^n} \|\mathbf{b}_{ij}^n \circ (\mathbf{x}_{ij}^n - \hat{\mathbf{x}}_{ij}^n)\|_2^2 = \frac{1}{\mathbf{b}_{ij}^{nT} \mathbf{b}_{ij}^n} \|\mathbf{y}_{ij}^n - \mathbf{b}_{ij}^n \circ \hat{\mathbf{x}}_{ij}^n\|_2^2.$$

$\widehat{MSE}_{ij}^n$ are computed at each pixel (i, j) for different n , and the block size $\hat{n} = \arg \min_n \widehat{MSE}_{ij}^n$ empirically obtained. Then in a separate image space $W(i, j) = \hat{n}$ is marked, which gives us a clustered image based on the selected block size.

5.2.2 Implementation Details

The framework is implemented according to the flowchart presented in Figure 5.1. In practice, the comparison of the sample mean square error will be unfair among the blocks

Figure 5.2: Illustration of the block size selection for inpainting.

of different size $n = n_1 < n_2 < n_3 < \dots$, because the number of samples are different for each block size. In order to stay unbiased, MSE_{ij}^n for each block is computed only over the region covered with the smallest block size n_1 . The comparison is done in terms of $\widehat{MSE}_{ij}^n = \frac{1}{\mathbf{b}_{ij}^{n_1 T} \mathbf{b}_{ij}^{n_1}} \|\mathbf{R}_{ij}^{n_1} \mathbf{R}_{ij}^{n_1 T} (\mathbf{y}_{ij}^n - \mathbf{b}_{ij}^n \circ \hat{\mathbf{x}}_{ij}^n)\|_2^2$, where $\mathbf{R}_{ij}^{n_1} \mathbf{R}_{ij}^{n_1 T}$ extracts the common pixels that are covered with block size n_1 .

Since $\widehat{MSE}_{ij}^n$ only compares the region covered with n_1 for any center pixel (i, j) , only those recovered pixels are used, which are covered with n_1 , that is $\hat{\mathbf{x}}_{ij}^{n_1} = \mathbf{R}_{ij}^{n_1} \mathbf{R}_{ij}^{n_1 T} \hat{\mathbf{x}}_{ij}^n$. Then the global inpainted image is recovered from these local inpainted image blocks $\{\forall_{ij} \hat{\mathbf{x}}_{ij}^{n_1}\}$. Thus, a MAP estimator is formulated similar to the denoising framework of the previous chapter,

$$\hat{X} = \arg \min_X \left\{ \lambda \|Y - B \circ X\|_2^2 + \sum_{ij} \|\mathbf{R}_{ij}^{n_1} X - \hat{\mathbf{x}}_{ij}^{n_1}\|_2^2 \right\}.$$

Differentiating the right hand side quadratic expression with respect to X , the following solution can be obtained.

$$\begin{aligned} -\lambda B \circ [Y - B \circ \hat{X}] + \sum_{ij} \mathbf{R}_{ij}^{n_1 T} [\mathbf{R}_{ij}^{n_1} \hat{X} - \hat{\mathbf{x}}_{ij}^{n_1}] &= 0 \\ \hat{X} &= \left[\lambda \text{diag}(B) + \sum_{ij} \mathbf{R}_{ij}^{n_1 T} \mathbf{R}_{ij}^{n_1} \right]^{-1} \left[\lambda Y + \sum_{ij} \mathbf{R}_{ij}^{n_1 T} \hat{\mathbf{x}}_{ij}^{n_1} \right] \end{aligned} \quad (\text{Eq. 5.2})$$

This expression means that averaging of the inpainted image blocks is to be done, with some relaxation obtained from the corrupt image. Hence $\lambda \propto 1/r$, where r is the fraction of pixels to be inpainted¹. The matrix to invert in the above expression is a diagonal one, hence the calculation of (Eq. 5.2) can be done on a pixel-by-pixel basis after $\{\forall_{ij} \hat{\mathbf{x}}_{ij}^{n_1}\}$ is obtained.

5.3 Denoising Using Local Sparse Representation

Similar to the earlier stated inpainting framework, square blocks of size $\sqrt{n} \times \sqrt{n}$ with a center pixel are considered, which means n is an odd number. Sweeping across the coordinate (i, j) of Y , the overlapping local patches are extracted, that is $\forall_{ij} \{\mathbf{y}_{ij}^n = \mathbf{R}_{ij}^n Y\} \in \mathbb{R}^n$. The original image patch is denoted as $\mathbf{x}_{ij}^n$, and the noise as $\mathbf{v}_{ij}^n \in \mathbf{N}^n(0, \sigma^2)$, making the noisy image patch $\mathbf{y}_{ij}^n = \mathbf{x}_{ij}^n + \mathbf{v}_{ij}^n$.

Let $\mathbf{D}^n$ be a known dictionary, where $\mathbf{x}_{ij}^n$ has a representation $\mathbf{x}_{ij}^n = \mathbf{D}^n \mathbf{s}_{ij}^n$, and $\mathbf{s}_{ij}^n$ is sparse. Since the additive random noise will not be sparse in any dictionary, $\mathbf{s}_{ij}^n$ is estimated as

$$\hat{\mathbf{s}}_{ij}^n = \arg \min_{\mathbf{s}} \|\mathbf{s}\|_0 \text{ i.e. } \|\mathbf{y}_{ij}^n - \mathbf{D}^n \mathbf{s}\|_2^2 \leq \epsilon^2(n), \quad (\text{Eq. 5.3})$$

where $\epsilon(n) \geq \|\mathbf{v}_{ij}^n\|_2$. According to multidimensional Gaussian distribution, if $\mathbf{v}_{ij}^n$ is an n dimensional Gaussian vector, $\|\mathbf{v}_{ij}^n\|_2^2$ is distributed by generalized Rayleigh law,

$$\mathbf{Pr} \left(\|\mathbf{v}_{ij}^n\|_2^2 \leq n(1 + \epsilon)\sigma^2 \right) = \frac{1}{\Gamma(\frac{n}{2})} \int_{z=0}^{\frac{n(1+\epsilon)}{2}} z^{\frac{n}{2}-1} e^{-z} dz. \quad (\text{Eq. 5.4})$$

By taking $\epsilon^2(n) = n(1 + \epsilon)\sigma^2$, for an appropriately bigger value of ϵ , it guarantees the sparse representation to be out of the noise radius with high probability. Thus, by using the estimated sparse representations, the denoised local image blocks can be obtained as $\{\forall_{ij} \hat{\mathbf{x}}_{ij}^n = \mathbf{D}^n \hat{\mathbf{s}}_{ij}^n\}$. Since the increase in block size causes decrease in the

¹All the experimental results are obtained keeping $\lambda = 60/r$

Figure 5.3: Flowchart of the proposed image denoising framework.

correlation between signal and noise, ε is reduced with increase in n to maintain an equal probability of denoising irrespective of block sizes. In spite of that the mean square error ($\frac{1}{n} \|\mathbf{x}_{ij}^n - \hat{\mathbf{x}}_{ij}^n\|_2^2$) varies with block size n . This is because an equal probability of the estimation being away from the noise radius does not imply equal closeness to the signal. As the dictionary of some block size matches better with the signal compared to the other, a minimum mean square error (MMSE) based block size selection becomes essential.

5.3.1 Local Block Size Selection for Denoising

The effect of block size is also very intuitive in denoising using sparse representation: bigger block sizes capture more details from the image, giving rise to more nonzero coefficients. Hence smaller block sizes are preferred for local sparse representation. In contrast, it is hard to distinguish between signal and noise in small sized blocks even in visual perspective, hence bigger block sizes are suitable for denoising. Thus, there exists

a trade-off between the block size and accuracy of fitting. In the absence of the noise free image, some measure need to be derived to reach

$$\min_n MSE_{ij}^n = \min_n \frac{1}{n} \|\mathbf{x}_{ij}^n - \hat{\mathbf{x}}_{ij}^n\|_2^2. \quad (\text{Eq. 5.5})$$

In order to solve the aforementioned problem, an approximation for $\min_n MSE_{ij}^n$ is carried out. It is known that the original image patch $\mathbf{x}_{ij}^n = \mathbf{y}_{ij}^n - \mathbf{v}_{ij}^n$, hence after taking the expectation for the noise, it can be written that

$$\begin{aligned} MSE_{ij}^n &= \frac{1}{n} \mathbf{E} \left[\|\mathbf{y}_{ij}^n - \hat{\mathbf{x}}_{ij}^n - \mathbf{v}_{ij}^n\|_2^2 \right] \\ &= \frac{1}{n} \mathbf{E} \left[\|\mathbf{y}_{ij}^n - \hat{\mathbf{x}}_{ij}^n\|_2^2 \right] - \frac{1}{n} \mathbf{E} \left[\mathbf{v}_{ij}^n (\mathbf{y}_{ij}^n - \hat{\mathbf{x}}_{ij}^n)^T \right] \\ &\quad - \frac{1}{n} \mathbf{E} \left[(\mathbf{y}_{ij}^n - \hat{\mathbf{x}}_{ij}^n) \mathbf{v}_{ij}^{nT} \right] + \frac{1}{n} \mathbf{E} \left[\|\mathbf{v}_{ij}^n\|_2^2 \right]. \end{aligned}$$

Heuristically, for a sufficiently large value of ε in (Eq. 5.3) the estimation $\hat{\mathbf{x}}_{ij}^n$ can be kept away from the noise $\mathbf{v}_{ij}^n$. Thus, $\mathbf{E} \left[\mathbf{v}_{ij}^n (\mathbf{y}_{ij}^n - \hat{\mathbf{x}}_{ij}^n)^T \right] = \mathbf{E} \left[(\mathbf{y}_{ij}^n - \hat{\mathbf{x}}_{ij}^n) \mathbf{v}_{ij}^{nT} \right] \sim \mathbf{E} \left[\|\mathbf{v}_{ij}^n\|_2^2 \right]$, which gives an approximation of MSE_{ij}^n ,

$$\widehat{MSE}_{ij}^n = \frac{1}{n} \mathbf{E} \left[\|\mathbf{y}_{ij}^n - \hat{\mathbf{x}}_{ij}^n\|_2^2 \right] - \frac{1}{n} \mathbf{E} \left[\|\mathbf{v}_{ij}^n\|_2^2 \right].$$

$\widehat{MSE}_{ij}^n$ are computed at each pixel (i, j) for different n , and the block size $\hat{n} = \arg \min_n \widehat{MSE}_{ij}^n$ is obtained empirically. Then in a separate image space $W(i, j) = \hat{n}$ is marked, which gives us a clustered image based on the selected block size.

5.3.2 Implementation Details

The framework is implemented according to the flowchart presented in Figure 5.3. In practice, the comparison of the sample mean square error will be unfair among the blocks of different size $n = n_1 < n_2 < n_3 < \dots$, because the number of samples are different for each block size. In order to stay unbiased, MSE_{ij}^n for each block is computed only over the region covered with the smallest block size n_1 . The comparison is done in terms

Figure 5.4: Illustration of clustering based on window selection for AWGN of various σ .

of $\widehat{MSE}_{ij}^n = \frac{1}{n_1} \|\mathbf{R}_{ij}^{n_1} \mathbf{R}_{ij}^{n_1 T} (\mathbf{y}_{ij}^n - \hat{\mathbf{x}}_{ij}^n)\|_2^2 - \frac{1}{n_1} \|\mathbf{v}_{ij}^n\|_2^2$, where $\mathbf{R}_{ij}^{n_1} \mathbf{R}_{ij}^{n_1 T}$ extracts the common pixels that are covered with block size n_1 .

It is also important to ensure that, irrespective of n , each estimated $\hat{\mathbf{x}}_{ij}^n$ is noise free with equal probability. Hence, the following result is established to maintain equal lower bound probabilities of denoising across n .

Lemma 5.1 *For an additive zero mean white Gaussian noise $\mathbf{v}_{ij}^n \in \mathbf{N}[0, \mathbf{I}_n \sigma^2]$, and the observed signal $\mathbf{y}_{ij}^n = \mathbf{D}^n \mathbf{s}_{ij}^n + \mathbf{v}_{ij}^n$, we will have a constant lower-bound for probability $\Pr(\|\mathbf{y}_{ij}^n - \mathbf{D}^n \mathbf{s}_{ij}^n\|_2^2 < n(1 + \varepsilon)\sigma^2)$ over n , by taking $\varepsilon = \frac{\varepsilon_0}{\sqrt{n}}$.*

Proof: $\|\mathbf{v}_{ij}^n\|_2^2$ is a random variable formed out of sum squared of n Gaussian random variables, and $E[\|\mathbf{v}_{ij}^n\|_2^2] = n\sigma^2$. Using Chernoff bound [44], it can be stated that

$$\Pr(\|\mathbf{v}_{ij}^n\|_2^2 \geq n(1 + \varepsilon)\sigma^2) \leq e^{-c_0 \varepsilon^2 n}.$$

The minimum possible estimation error is $\|\mathbf{y}_{ij}^n - \mathbf{D}^n \mathbf{s}_{ij}^n\|_2^2 = \|\mathbf{v}_{ij}^n\|_2^2$, and $\Pr(\|\mathbf{v}_{ij}^n\|_2^2 < n(1 + \varepsilon)\sigma^2) = 1 - \Pr(\|\mathbf{v}_{ij}^n\|_2^2 \geq n(1 + \varepsilon)\sigma^2)$. For $\varepsilon = \frac{\varepsilon_0}{\sqrt{n}}$, it gives

$$\Pr(\|\mathbf{y}_{ij}^n - \mathbf{D}^n \mathbf{s}_{ij}^n\|_2^2 < n(1 + \varepsilon)\sigma^2) > 1 - e^{-c_0 (\frac{\varepsilon_0}{\sqrt{n}})^2 n} = 1 - e^{-c_0 \varepsilon_0^2},$$

which is a constant lower-bound irrespective of n .

Similar to the inpainting problem, the common denoised pixels are extracted as per the smallest block size n_1 after block size is selected for any pixel location (i, j) , i.e. $\hat{\mathbf{x}}_{ij}^{n_1} = \mathbf{R}_{ij}^{n_1} \mathbf{R}_{ij}^{n_1 T} \hat{\mathbf{x}}_{ij}^n$. Then the overlapping local patches are average to recover each pixel of the image,

$$\hat{X} = \left[\lambda \mathbf{I}_N + \sum_{ij} \mathbf{R}_{ij}^{n_1 T} \mathbf{R}_{ij}^{n_1} \right]^{-1} \left[\lambda Y + \sum_{ij} \mathbf{R}_{ij}^{n_1 T} \hat{\mathbf{x}}_{ij}^{n_1} \right], \quad (\text{Eq. 5.6})$$

which is same as the MAP based local to global recovery in the previous chapter.

It is known that a better dictionary produces a better denoising result, and that the dictionary training algorithms are capable of performing in presence of noise. Hence, from the noisy image, trained dictionaries are obtained, similar to the previous chapter, and then the image are denoised using the block size selection framework presented in Figure 5.3.

5.4 Experimental Results

5.4.1 Inpainting

To validate the proposed framework of image inpainting, it is experimented on Barbara image with pixels missing at random locations, and the image of girls spoiled by mascara. The results are compared with some of the recently proposed inpainting frameworks “MCA” (morphological component analysis) [12] and “EM” (expectation maximization) [13]. Local blocks centering over each pixel are extracted for 256×256 images, whereas local blocks centering over each alternating pixel location of the alternating rows are extracted for 512×512 images. Overcomplete discrete cosine transform (DCT) dictionary is taken with $K = 4n$ number of atoms for sparse representation. The error tolerance for sparse representation is set as $\epsilon(n) = 3\sqrt{n}$. A local block size selection is performed by taking increasing square block sizes 15×15 , 17×17 and 19×19 as described in section 5.2.1. Block size based clustered images for different masks B are shown in Figure 5.2 (the gray levels are in increasing order of block size).

After the block sizes have been identified for every location, inpainting is performed for every single local block. Global recovery is done by averaging the overlapped regions as per (Eq. 5.2). The inpainting results for both [12] and [13] are obtained using the MCALab toolbox provided in [45]. A visual comparison between the proposed framework and the algorithms in [12] and [13] is presented in Figure 5.5, where mascara is removed

Mascara on Girls

Text on Lena

80% missing pixel Barbara

EM [13]

EM [13] (PSNR 31.26 dB)

EM [13] (PSNR 27.13 dB)

MCA [12]

MCA [12] (PSNR 34.18 dB)

MCA [12] (PSNR 26.62 dB)

Proposed

Proposed (PSNR 34.57 dB)

Proposed (PSNR 27.14 dB)

Figure 5.5: Visual comparison of inpainting performance across the methods.

Table 5.1: Image inpainting performance comparison in PSNR

missing pixel	Images							Method
	Barbra	Lena	Man	Couple	Hill	Boat	Stream	
50%	32.95	34.16	29.23	31.10	31.92	31.83	25.93	EM
	31.79	32.90	29.01	30.73	31.45	31.21	26.53	MCA
	34.63	36.53	31.09	32.95	33.89	33.27	27.29	Proposed
80%	17.13	29.91	24.84	26.56	27.96	26.91	22.31	EM
	26.61	28.53	24.73	26.22	27.44	26.49	22.94	MCA
	27.14	29.94	25.45	26.82	28.47	26.55	23.17	Proposed

from Girls image, text is removed from the Lena, and 80% of the missing pixels are filled in Barbra image. It can be seen that the images inpainted by the proposed framework are subjectively better in comparison to the rest, since it has more details and fewer artifacts. In terms of quantitative comparison, the proposed framework has also achieved a better Peak Signal to Noise Ratio (PSNR), which is presented in Table 5.1 for the cases of random missing pixels.

5.4.2 Denoising

To validate the proposed framework of image denoising, it is experimented on some well known gray scale images corrupted with AWGN ($\sigma = 5, 15$ and 25). The obtained results are compared with [11] (K -SVD), and one of its close competitor [29] (K -LLD). K -LLD is a recently proposed denoising framework, which tried to exceed K -SVD's denoising performance by clustering the extracted local image blocks, and by performing sparse representation on each cluster through locally learned dictionaries ².

In the experimental set up, local blocks centering over each pixel are extracted for 256×256 images, whereas local blocks centering over each alternating pixel location of the alternating rows are extracted for 512×512 images. The number of atoms are kept

²The PCA frame derived from the image blocks of each cluster is defined as the locally learned dictionary. Please note that, number of clusters K of [29] is not the same as number of atoms in the dictionary of the proposed framework, it is just a coincidence.

as $K = 4n$ for each block size n . For each block size, to get more than 96% probability of denoising as per (Eq. 5.5), the value of $\varepsilon = 2.68$ is kept in accordance with Lemma 5.1. Increasing square blocks of size 11×11 , 13×13 and 15×15 are taken, and selected the local block size as described in section 5.3.1. The selected block size based clustered images are shown in Figure 5.4 (the gray levels are in increasing order of block size). It can be seen clearly that there exists a tradeoff between the noise level and local block size used for sparse representation. When the noise level goes up, a total shift of the clusters from smooth region to texture like region is observed.

For each block size, the trained dictionaries are obtained from a corrupt image using SGK, in the same manner as the denoising experiment of previous chapter. However, number of SGK iterations used are different for different block sizes. Since K -SVD has used 10 iterations for 8×8 blocks, $\lceil 10 \frac{n}{64} \rceil$ iterations are used for $\sqrt{n} \times \sqrt{n}$ blocks. After obtaining the trained dictionaries, the best block size for each location is decided. Then, the image is recovered by averaging the overlapped regions as per (Eq. 5.6), by taking $\lambda = 30/\sigma$.

A visual comparison between the proposed framework and the algorithms in [11, 29] is presented in Figure 5.6, where the images are heavily corrupted by AWGN $\sigma = 25$. In comparison to the rest, it can be seen that the proposed denoising framework produces subjectively better results, since it has more details and fewer artifacts. Notably, the edges in the house image, the complex objects in the man image, and the joint between the mandibles of the parrot image are well recovered. In Figure 5.7 a visual comparison is made for the denoising performance on these diverse and irregular objects. It can be seen that the proposed framework is better. In K -LLD denoised image irregularities are heavily smoothed, and a curly artifact is spreading all over. Frameworks like K -LLD has the potential to recover the images better, by taking advantage of self similarity inside the images. However, they have a clear drawback when the image has diversity and

Noisy Parrot

Noisy Man

Noisy House

K -SVD[11] (PSNR 28.43 dB)

K -SVD[11] (PSNR 28.11 dB)

K -SVD[11] (PSNR 32.10 dB)

K -LLD[29] (PSNR 27.89 dB)

K -LLD[29] (PSNR 28.26 dB)

K -LLD[29] (PSNR 30.67 dB)

Proposed (PSNR 28.48 dB)

Proposed (PSNR 28.37 dB)

Proposed (PSNR 32.51 dB)

Figure 5.6: Visual comparison of the denoising performances for AWGN ($\sigma = 25$).

Figure 5.7: Visual inspection at irregularities

irregular discontinuity, which has been taken care by block size selection in the proposed frame work.

A quantitative comparison by PSNR is also made, and results are shown in Table

Table 5.2: Image denoising performance comparison in PSNR

σ	Images							Method
	CamMan	Parrot	Man	Montage	Peppers	Aerial	House	
5	37.90	37.57	36.78	40.17	37.87	35.57	39.45	<i>K</i> -SVD
	36.98	36.65	36.44	39.46	37.09	35.23	37.89	<i>K</i> -LLD
	37.66	37.42	36.77	39.96	37.72	35.33	39.51	Proposed
15	31.38	30.98	30.57	33.77	32.21	28.64	34.32	<i>K</i> -SVD
	30.78	30.76	30.76	33.14	31.96	28.55	33.89	<i>K</i> -LLD
	31.31	30.90	30.74	33.78	32.25	28.49	34.60	Proposed
25	28.81	28.43	28.11	30.97	29.74	25.95	32.10	<i>K</i> -SVD
	27.96	27.89	28.26	29.52	28.94	25.78	30.67	<i>K</i> -LLD
	28.96	28.48	28.37	31.21	29.91	25.98	32.51	Proposed
50	25.66	25.35	24.99	27.12	26.16	22.44	28.03	<i>K</i> -SVD
	20.30	20.11	20.36	20.39	20.34	19.62	20.90	<i>K</i> -LLD
	25.92	25.51	25.24	27.35	26.48	22.85	28.66	Proposed

5.2. It can be seen that the proposed framework produces a better PSNR compare to the frameworks in [29]. In the case of higher noise level ($\sigma \geq 25$), the proposed framework performs better in comparison to both [11] and [29].

5.5 Discussions

In this chapter, image inpainting and denoising using local sparse representation are illustrated in a framework of location adaptive block size selection. This framework is motivated by the importance of block size selection in inferring the geometrical structures and details in the images. It starts with clustering the image based on the block size selected at every location that minimizes the local MSE. Subsequently it aggregates the individual local estimations to estimate the final image. The experimental results show their potential in comparison to the state of the art image recovery techniques. While this chapter addresses recovery of gray scale images, it can also be extended to color images. The present work provides stimulating results with an intuitive platform for further investigation.

In the present framework, the block sizes are prefixed. However, the bounds on the local block size is an interesting topic to explore further. In the present framework of aggregation, all the pixels of the recovered blocks are given equal weight. An improvement may be achieved by deriving an aggregation formula with adaptive weights per pixel for the recovered local window.

5.6 Summary

In order to have a better recovery (inpainting and denoising) of underlying image details, an adaptive local block size based sparse representation framework is proposed. A simple local block size selection criterion was introduced for image inpainting. A maximum a

posteriori probability (MAP) based aggregation formula is derived to inpaint the global image from the overlapping local inpainted blocks. The proposed inpainting framework produces a better inpainting result compared to the state of the art image inpainting techniques. A simple local block size selection criterion was introduced for image denoising. A block size based representation error threshold is derived to perform equiprobable denoising of the image blocks of different size. In the case of heavy noise, the proposed local block size selection based denoising framework produced a relatively better denoising compared to some of the recently proposed image denoising frameworks based on sparse representation.

Chapter 6

Extended Orthogonal Matching Pursuit

In order to achieve the benchmark performance of BP many variants of OMP have been proposed in recent years, e.g. regularized OMP [46], stagewise OMP [47], backtracking based adaptive OMP [48], etc. However, a well known behavior of basic OMP still remains unexplored. Experiments suggest OMP can produce superior result by going beyond m -iterations [49, chapter 8, footnote 6]. The aim of this chapter is to provide an analytical result that can bring down the gap between practice and theory. The main result is the following theorem:

Theorem 6.1 (OMP with Admissible Measurements) *Fix $\alpha \in [0, 1]$, and choose $d \geq C_0 m \ln \frac{K}{\lfloor \alpha m \rfloor + 1}$, where C_0 is an absolute constant. Suppose that $\mathbf{s}$ is an arbitrary m -sparse signal in $\mathbb{R}^K$, and draw a random $d \times K$ admissible measurement matrix Φ independent from the signal. Given the data $\mathbf{z} = \Phi \mathbf{s}$, OMP can reconstruct the signal with probability exceeding $1 - e^{-c_0 \frac{d}{m} (\lfloor \alpha m \rfloor + 1)}$ in at most $m + \lfloor \alpha m \rfloor$ iterations.*

The above result brings the number of measurements for BP and OMP to the same order, when $\alpha \rightarrow 1$. Being motivated by this result, a further extension to OMP is proposed for CS recovery, which does not require any prior knowledge of sparsity. The

result presented in this chapter is mostly inspired by Tropp and Gilbert's analysis of OMP for m -iterations [15], and it simplifies to their result when $\alpha = 0$. Similar to [15], the obtained result is valid for random independent atoms. In contrast, the result for BP shows uniform recovery of all sparse signals over a single set of random measurement vectors. Nevertheless, OMP remains a valuable tool along with its inherent advantages, which makes theorem 6.1 more attractive.

6.1 OMP for CS Recovery

In the problem of CS recovery using OMP, the sparsity of the measured signal $\mathbf{s}$ is known *a priori*, that is $\mathbf{s}$ has non-zero entries only at m unknown indices. Let's define the unknown support of $\mathbf{s}$ as I , and $\|\mathbf{s}\|_0 = |I| = m$. The atoms φ_j corresponding to these indices $j \in I$ is referred as correct atoms, and rest $\varphi_j : j \notin I$ as wrong atoms. OMP identifies I by selecting one candidate index in each iteration, as described in chapter-2 (Algorithm 2 in section 2.4).

At each iteration t , the residue $\mathbf{r}_{t-1}$ is always orthogonal to all the selected atoms $\Phi_{\Lambda_{t-1}}$. That means the non-zero correlation $\langle \varphi_j, \mathbf{r}_{t-1} \rangle \neq 0$ will only occur for those atoms, which are not linear combinations of atoms in $\Phi_{\Lambda_{t-1}}$. Thus at iteration t , OMP will select an atom φ_{λ_t} which is linearly independent from the previously selected atoms $\Phi_{\Lambda_{t-1}} = \{\varphi_{\lambda_1}, \varphi_{\lambda_2}, \dots, \varphi_{\lambda_{t-1}}\}$, i.e. $\lambda_t \in \{j : \varphi_j \notin \mathbf{R}(\Phi_{\Lambda_{t-1}})\}$. Therefore, the obvious choice for m -sparse signal recovery is to identify m correct atoms in $t_{\max} = m$ iterations of OMP [15]. The following proposition provides the recovery scenarios.

Proposition 6.1 *Take an arbitrary m -sparse signal $\mathbf{s}$ in $\mathbb{R}^K$, and let Φ be any $d \times K$ measurement ensemble with the property that any $2m$ atoms are linearly independent. Given the data vector $\mathbf{z} = \Phi\mathbf{s}$,*

- *OMP for $t_{\max} < m$ will result in $\mathbf{r}_{t_{\max}} \neq 0$;*

- OMP for $t_{\max} = m$ will result in $\mathbf{r}_{t_{\max}} \neq 0$, if $\hat{\mathbf{s}} \neq \mathbf{s}$;
- OMP for $t_{\max} = m$ will result in $\mathbf{r}_{t_{\max}} = 0$, if $\hat{\mathbf{s}} = \mathbf{s}$.

Proof: It can easily be proved by contradiction. If signal residue vanishes i.e. $\mathbf{r}_{t_{\max}} = 0$ after any $t_{\max}$ iterations, that means a $t_{\max}$ -sparse solution $\mathbf{z} = \Phi \hat{\mathbf{s}}$ is found. As there exists a generating m -sparse solution $\mathbf{s}$, it can be stated as $\Phi(\hat{\mathbf{s}} - \mathbf{s}) = 0$, where the signal $(\hat{\mathbf{s}} - \mathbf{s})$ can have a maximum of $t_{\max} + m$ nonzero coefficients i.e. $\|\hat{\mathbf{s}} - \mathbf{s}\|_0 \leq t_{\max} + m$. For $t_{\max} \leq m$ it becomes contradictory, if Φ has a property that any $2m$ columns of it are linearly independent. Hence it is proved that for such Φ , the signal residue of OMP will not vanish for $t_{\max} < m$, or $t_{\max} = m$ and $\hat{\mathbf{s}} \neq \mathbf{s}$.

Note 6.1 Proposition 6.1 is a general version of proposition 7 of [15], with similar arguments. [15] only considers $t_{\max} = m$ and random Φ case.

- Note that since Restricted Isometry Property (RIP) of order $2m$ ensures that any $2m$ columns of Φ are linearly independent, any Φ satisfying RIP of order $2m$ will satisfy the above proposition.
- Note that since Gaussian or Bernoulli measurement ensemble of any $2m$ columns are linearly independent with probability close to one for $d \geq 2m$ [50, 51], any Φ made out of these random ensemble will satisfy the above proposition with a very high probability.

RIP of order $2m$ requires $d = O\left(m \ln \frac{K}{m}\right)$ in the case of random measurement matrices. While proposition 6.1 says that a RIP of order $2m$ is necessary for a unique solution $\mathbf{s}$ at $t_{\max} = m$ for which the residue vanishes, it cannot guarantee that OMP will obtain a solution at $t_{\max} = m$ with high probability. In order for that to happen with high probability, OMP needs $d = O(m \ln K) > O\left(m \ln \frac{K}{m}\right)$ measurements. This is because, further to RIP of order $2m$, the probability of selecting m correct atoms in m iterations decides the requirement of d for OMP.

6.2 Extended OMP for CS Recovery

Identifying a m -sparse signal in only m selections is a sheer restriction to OMP, which has motivated many backtracking based greedy algorithms, like regularized OMP [46], stagewise OMP [47], backtracking based adaptive OMP [48], etc. These algorithms work with the main strategy of selecting more and then tracking back to m atoms. However, the fundamental behavior of OMP when it selects more atoms is the point of interest in this work.

It can be observed that, when OMP has failed to pick m correct atoms out of Φ_I in m iterations, it has not reached a solution and $\mathbf{r}_m \neq 0$. However, if the iteration is extended beyond m , then the chances of selecting m correct atoms will increase. Even though there are no published experimental results, this scenario is well known to the researchers working on greedy pursuits [49, chapter 8, footnote 6]. [52] proposes to run OMP for $O(m^{1.2})$ iterations, and analytically shows that if $d = O(m^{1.6} \log K)$, any m -sparse signal can be recovered with a high probability in its version of extended run OMP. The required d is higher than both BP and OMP [15], and the complexity increases to order of $O(m^{1.2}dK)$.

In this work, the run of OMP is linearly extended beyond m iterations. Run of OMP for $t_{\max} = m + \lfloor \alpha m \rfloor$ iterations is proposed, which is referred as OMP_α here onwards, where $\alpha \geq 0$. This extended run may increase the computational cost of OMP only by a factor $1 + \alpha$, but it will still be of order $O(mdK)$.

Algorithm 6.3 (OMP_α for CS recovery) *The only change is at step vii of OMP algorithm described in chapter 2 (OMP for CS recovery) with an additional input of α :*

vii) Go to Step.2 if $t < m + \lfloor \alpha m \rfloor$, else terminate;

By allowing an additional selection of $\lfloor \alpha m \rfloor$ atoms, the chance of acquiring m correct atoms is increased. Thus, the conventional use of OMP for CS recovery can be viewed as a limiting case of OMP_α where $\alpha = 0$. By using its orthogonality property, and RIP of the sensing matrix, the following proposition shows how OMP_α can identify the m correct atoms from the $m + \lfloor \alpha m \rfloor$ selections.

Proposition 6.2 *Take an arbitrary m -sparse signal $\mathbf{s} \in \mathbb{R}^K$, and let Φ be an $d \times K$ measurement ensemble satisfying RIP of order $m + \lfloor \alpha m \rfloor$. Given the data vector $\mathbf{z} = \Phi \mathbf{s}$;*

- (S) *OMP_α will successfully identify any m -sparse signal $\mathbf{s}$, and $\mathbf{r}_{m+\lfloor \alpha m \rfloor} = 0$, if $I \subseteq \Lambda_{m+\lfloor \alpha m \rfloor}$,*
- (F) *OMP_α will fail to identify any m -sparse signal $\mathbf{s}$, irrespective of $\mathbf{r}_{m+\lfloor \alpha m \rfloor}$, if $I \not\subseteq \Lambda_{m+\lfloor \alpha m \rfloor}$.*

Proof: At t^{th} iteration, OMP_α will find a t -term least square approximation $\hat{\mathbf{s}}_{\Lambda_t} = \Phi_{\Lambda_t}^\dagger \mathbf{z}$. The best least square approximation for any linear system is the exact solution, leading to $\Phi \hat{\mathbf{s}} = \mathbf{z} \implies \mathbf{r}_t = 0$, which can only be possible if $\mathbf{z}$ lies in the column space $\mathbf{R}(\Phi_{\Lambda_t})$. Since $I \subseteq \Lambda_{m+\lfloor \alpha m \rfloor}$ and $\mathbf{z} \in \mathbf{R}(\Phi_I)$ implies $\mathbf{z} \in \mathbf{R}(\Phi_{\Lambda_{m+\lfloor \alpha m \rfloor}})$, the obtained $(m + \lfloor \alpha m \rfloor)$ -term solution is exact, i.e. $\mathbf{z} = \Phi \hat{\mathbf{s}}$. However, this makes $\Phi(\hat{\mathbf{s}} - \mathbf{s}) = 0$, which implies that Φ contains less than or equal to $m + \lfloor \alpha m \rfloor$ linearly dependent atoms, because $\|\hat{\mathbf{s}} - \mathbf{s}\|_0 \leq m + \lfloor \alpha m \rfloor$. It becomes contradictory since Φ satisfies RIP of order $m + \lfloor \alpha m \rfloor$. Therefore $\hat{\mathbf{s}} = \mathbf{s}$, and OMP_α successfully identifies the s -sparse signal.

Conversely, $I \not\subseteq \Lambda_{m+\lfloor \alpha m \rfloor} \implies \mathbf{R}(\Phi_I) \not\subseteq \mathbf{R}(\Phi_{\Lambda_{m+\lfloor \alpha m \rfloor}})$, then $\hat{\mathbf{s}}_{\Lambda_{m+\lfloor \alpha m \rfloor}}$ will either produce a $(m + \lfloor \alpha m \rfloor)$ -term least square solution leading to signal residue $\mathbf{r}_{m+\lfloor \alpha m \rfloor} = 0$, or a $(m + \lfloor \alpha m \rfloor)$ -term least square approximation with signal residue $\mathbf{r}_{m+\lfloor \alpha m \rfloor} \neq 0$. In either case OMP_α has failed to identify the exact m -term solution using columns of Φ_I .

Figure 6.1: The percentage of signal recovered in 1000 trials with increasing α , for various m -sparse signals in dimension $K = 1024$, from their $d = 256$ random measurements.

The event (S) stands for successful recovery in proposition 6.2, which is a super set to the event of success in standard OMP. It is intuitive that the occurrence of event (S) has a higher probability for $\alpha > 0$ than for $\alpha = 0$. In order to see the behavior of event (S), an empirical observation of probability vs α is plotted in Fig. 6.1, which shows the increase in probability of recovery with α .

6.3 Analysis of OMP_α

In order to function as a recovery algorithm, OMP_α only requires RIP of order $(m + \lfloor \alpha m \rfloor)$. This means for $\alpha = 0$ (i.e OMP), only RIP of order m is enough to function. However, for the event (S) to occur with high probability, the requirement of d is more, as discussed in section 2.4 of chapter 2. Choosing $\alpha > 0$ is expected to reduce this required d . In order to provide unique measurements, Φ is required to follow theorem 1.1

by satisfying RIP of order $2m$ for $m \in (0, K/2)$. Thus α may be as large as 1 without requiring higher order of RIP, and α is restricted to the range $[0, 1]$.

Given the unique measurement vector $\mathbf{z} = \Phi \mathbf{s}$ from a $d \times K$ measurement ensemble satisfying RIP of order $2m$, what is the constraint on d for success of OMP_α ? With the obtained constraint, how will the probability of success of OMP_α behave? In order to answer these questions, a set of admissible measurement matrices will be defined based on the properties of Gaussian/Bernoulli sensing matrices. Then, the success of OMP_α will be analyzed using the properties of these admissible matrices.

6.3.1 Admissible Measurement Matrix

Matrices $\Phi \in \mathbb{R}^{d \times K}$ with entries $\Phi(i, j)$ as i.i.d. Gaussian random variable $(0, \frac{1}{\sqrt{d}})$ or i.i.d. Bernoulli random variable with sample space $\{\frac{1}{\sqrt{d}}, -\frac{1}{\sqrt{d}}\}$ are considered to be good choices for the measurement matrix. These matrices are known to satisfy RIP of order $2m$ [53]. Let's assume that $d \geq C_1 m \ln \frac{K}{m}$, such that theorem 1.1 holds for Φ . Apart from this, four other useful properties of Φ are the following.

(P₀) Independence: Columns of Φ are statistically independent.

(P₁) Normalized: $\forall j \quad \mathbb{E}[\|\varphi_j\|_2^2] = 1$.

(P₂) Correlation: Let $\mathbf{u}$ be a vector whose ℓ_2 norm $\|\mathbf{u}\|_2 = 1$, and φ be a column of Φ independent of $\mathbf{u}$. Then, for any $\varepsilon > 0$, the probability

$$\mathbb{P}\{|\langle \varphi, \mathbf{u} \rangle| \geq \varepsilon\} \leq 2e^{-c_2 \varepsilon^2 d}.$$

The above inequality can easily be verified from the tail bound of any probability distribution (Gaussian and Bernoulli).

(P₃) Bounded singular value: For a given $d \times m$ submatrix Φ_I from Φ , the singular values $\sigma(\Phi_I)$ satisfy,

$$\mathbb{P} \{ \sigma(\Phi_I) \geq (1 - \delta) \} \geq 1 - e^{-c_1 d}$$

where $0 \leq \delta < 1$. This is equivalent to state that for any vector $\mathbf{x}$,

$$\mathbb{P} \{ \|\Phi_I \mathbf{s}_I\|_2^2 \geq (1 - \delta) \|\mathbf{s}_I\|_2^2 \} \geq 1 - e^{-c_1 d}$$

which is obvious, as Φ satisfies theorem 1.1.

6.3.2 Probability of Success

OMP _{α} works by selecting the candidate atoms φ_j one after another by looking at their correlation with the residue $\mathbf{r}_{t-1}$. Let's partition the measurement matrix into two sets of atoms, i.e. $\Phi = [\Phi_I, \Phi_{I^c}]$, where $\Phi_I \stackrel{\text{def}}{=} \{\varphi_j : j \in I\}$ is the set of correct atoms, and $\Phi_{I^c} \stackrel{\text{def}}{=} \{\varphi_j : j \in I^c\}$ is set of the remaining atoms (also termed as wrong atoms). Using correlation of the partitioned Φ it can be classified whether OMP _{α} will reliably select a correct atom from Φ_I or a wrong atom from Φ_{I^c} .

$$\text{Correct atom: } \iff \max_{j \in I^c} |\langle \varphi_j, \mathbf{r}_{t-1} \rangle| < \|\Phi_I^T \mathbf{r}_{t-1}\|_\infty.$$

$$\text{Wrong atom: } \iff \exists_{j \in I^c} |\langle \varphi_j, \mathbf{r}_{t-1} \rangle| \geq \|\Phi_I^T \mathbf{r}_{t-1}\|_\infty.$$

It is important to note that in the case of $|\langle \varphi_j, \mathbf{r}_{t-1} \rangle| = \|\Phi_I^T \mathbf{r}_{t-1}\|_\infty$, selections of both wrong and correct atoms are possible. In order to keep the analysis simple, this tie scenario is classified as selection of wrong atoms.

In order to analyze the probability of success, let's specify the outcome of a run of OMP _{α} as $\Lambda_{m+\lfloor \alpha m \rfloor} = \{\lambda_1, \lambda_2, \dots, \lambda_{m+\lfloor \alpha m \rfloor}\}$, where $\lambda_t \in \{1, 2, \dots, K\}$ denotes the index of the atom chosen in iteration t . Since the exact sequence these atoms appear is not important in determining the success or failure, the set of indices $\{\lambda_t\}$ is only considered. Let's define the set of correct selections as $J_C = \{\lambda_t : \lambda_t \in I\}$, which means for these

iterations $\max_{j \in I^c} |\langle \varphi_j, \mathbf{r}_{t-1} \rangle| < \|\Phi_I^T \mathbf{r}_{t-1}\|_\infty$. Let's also define $J_W = \{\lambda_t : \lambda_t \in I^c\}$, which in turn means that $\max_{j \in I^c} |\langle \varphi_j, \mathbf{r}_{t-1} \rangle| = |\langle \varphi_{\lambda_t}, \mathbf{r}_{t-1} \rangle| \geq \|\Phi_I^T \mathbf{r}_{t-1}\|_\infty$ denoting selection of a wrong atom. Using these sets the *Success* (S) and *Failure* (F) of the OMP_α can be explained.

(S) After $m + \lfloor \alpha m \rfloor$ steps if $|J_C| = m$ and $|J_W| = \lfloor \alpha m \rfloor$ is obtained, then certainly $I \subseteq \Lambda_{m+\lfloor \alpha m \rfloor}$. Note that $\alpha = 0$ implies success in conventional OMP, while $0 < \alpha \leq 1$ implies success in OMP_α .

(F) After $m + \lfloor \alpha m \rfloor$ steps if $|J_C| < m$ and $\lfloor \alpha m \rfloor + 1 \leq |J_W| \leq \lfloor \alpha m \rfloor + m$ is obtained, Then $I \not\subseteq \Lambda_{m+\lfloor \alpha m \rfloor}$ (excluding tie scenario) and OMP_α has failed.

With the conservative definition of failure as described earlier, the event of all possible failures is defined as

$$E_{\text{fail}} \stackrel{\text{def}}{=} \bigcup_{k=\lfloor \alpha m \rfloor+1}^{\lfloor \alpha m \rfloor+m} \left\{ \bigcup_{|J_W|=k} J_W \right\} \quad (\text{Eq. 6.1})$$

and the complementary event of success is defined as E_{succ} . Thus OMP_α 's success probability for any conditional event Σ can be written as $\mathbb{P}(E_{\text{succ}}|\Sigma) = 1 - \mathbb{P}(E_{\text{fail}}|\Sigma)$.

6.3.3 Main Result

Theorem 6.1 (OMP with Admissible Measurements) Fix $\alpha \in [0, 1]$, and choose $d \geq C_0 m \ln \frac{K}{\lfloor \alpha m \rfloor + 1}$, where C_0 is an absolute constant. Suppose that $\mathbf{s}$ is an arbitrary m -sparse signal in $\mathbb{R}^K$, and draw a random $d \times K$ admissible measurement matrix Φ independent from the signal. Given the data $\mathbf{z} = \Phi \mathbf{s}$, OMP can reconstruct the signal with probability exceeding $1 - e^{-c_0 \frac{d}{m} (\lfloor \alpha m \rfloor + 1)}$ in at most $m + \lfloor \alpha m \rfloor$ iterations.

Proof: The success probability $\mathbb{P}(E_{\text{succ}}) = \mathbb{P}(E_{\text{succ}}, \Sigma) + \mathbb{P}(E_{\text{succ}}, \Sigma^c)$, where conditional event Σ means Φ satisfies RIP of order $2m$, or

$$\mathbb{P}(\Sigma) = \mathbb{P}\{(1 + \delta) \geq \sigma(\Phi_{\Lambda_{2m}}) \geq (1 - \delta)\} \geq 1 - e^{-c_1 d}.$$

This also means Φ will satisfy RIP of order $m + \lfloor \alpha m \rfloor$ for $\alpha \in [0, 1]$, with probability exceeding $1 - e^{-c_1 d}$. The occurrence of the event Σ is very essential for OMP_α to function (see proposition 6.2). This implies $\mathbb{P}(E_{\text{succ}}, \Sigma^c) \rightarrow 0$ and may be ignored.

$$\mathbb{P}(E_{\text{succ}}) > \mathbb{P}(E_{\text{succ}}, \Sigma) = \mathbb{P}(\Sigma) (1 - \mathbb{P}(E_{\text{fail}}|\Sigma)).$$

Since $\mathbb{P}(\Sigma) \rightarrow 1$, the above inequality can be expressed as

$$\mathbb{P}(E_{\text{succ}}) \geq (1 - \mathbb{P}(E_{\text{fail}}|\Sigma)). \quad (\text{Eq. 6.2})$$

Thus, a lesser value of $\mathbb{P}(E_{\text{fail}}|\Sigma)$ means a better chance of success. Let's now estimate the failure probability from equation (Eq. 6.1) using union bound,

$$\begin{aligned} \mathbb{P}(E_{\text{fail}}) &\leq \sum_{k=\lfloor \alpha m \rfloor + 1}^{\lfloor \alpha m \rfloor + m} \mathbb{P} \left\{ \bigcup_{|J_W|=k} J_W \right\} \\ &\leq \sum_{k=\lfloor \alpha m \rfloor + 1}^{\lfloor \alpha m \rfloor + m} \binom{K-m}{k} \mathbb{P} \{ J_W |_{|J_W|=k} \} \end{aligned} \quad (\text{Eq. 6.3})$$

where $\bigcup_{|J_W|=k} J_W$ denotes all possible J_W having size k , and $J_W |_{|J_W|=k}$ denotes one such J_W . Due to the property (P₀), $\mathbb{P} \{ J_W |_{|J_W|=k} \}$ is same for any J_W having size k , and does not depend on the specific atomic indices in it.

$|J_W| = k$ means, OMP_α has selected k *wrong atoms*, i.e. $\bigcap_{\lambda_t \in J_W} |\langle \varphi_{\lambda_t}, \mathbf{r}_{t-1} \rangle| \geq \|\Phi_I^T \mathbf{r}_{t-1}\|_\infty$ irrespective of iteration of occurrence t . Property (P₀) states that φ_{λ_t} are independent, and a pessimistic assumption is made that each event of unreliable selection is independent of each other. Thus using (P₁) it can be stated that

$$\begin{aligned} \mathbb{P} \{ J_W |_{|J_W|=k} \} &= \mathbb{P} \left\{ \bigcap_{\lambda_t \in J_W} |\langle \varphi_{\lambda_t}, \mathbf{r}_{t-1} \rangle| \geq \|\Phi_I^T \mathbf{r}_{t-1}\|_\infty \right\} \\ &\simeq \mathbb{P}^k \{ |\langle \varphi_{\lambda_t}, \mathbf{r}_{t-1} \rangle| \geq \|\Phi_I^T \mathbf{r}_{t-1}\|_\infty \} \\ &= \mathbb{P}^k \{ |\langle \varphi, \mathbf{r}_{t-1} \rangle| \geq \|\Phi_I^T \mathbf{r}_{t-1}\|_\infty \} \end{aligned}$$

since the probability on the right side is same for any $\varphi \in \Phi_{I^c}$.

In order to simplify the derivation let's normalize the residue vector to $\mathbf{u} = \frac{\mathbf{r}_{t-1}}{\|\mathbf{r}_{t-1}\|_2}$, which makes $\|\mathbf{u}\|_2 = 1$. Normalizing $\mathbf{r}_{t-1}$ on both the sides will not affect the probability estimation, thus

$$\mathbb{P} \left\{ J_W \middle|_{|J_W|=k} \right\} = \mathbb{P}^k \left\{ |\langle \varphi, \mathbf{u} \rangle| \geq \|\Phi_I^T \mathbf{u}\|_\infty \right\}.$$

It is known that $\forall x \in \mathbb{R}^m$, $\|x\|_\infty \geq \frac{\|x\|_2}{\sqrt{m}}$. As $\Phi_I^T \mathbf{u}$ is a m -dimensional vector, it is true that $\|\Phi_I^T \mathbf{u}\|_\infty \geq \frac{\|\Phi_I^T \mathbf{u}\|_2}{\sqrt{m}}$. Thus it can be stated that

$$\mathbb{P} \left\{ J_W \middle|_{|J_W|=k} \right\} \leq \mathbb{P}^k \left\{ |\langle \varphi, \mathbf{u} \rangle| \geq \frac{\|\Phi_I^T \mathbf{u}\|_2}{\sqrt{m}} \right\}$$

Since the left side event is a subset of the right side event, the upper bound on its probability will remain true for any given condition. By taking the conditional event as Σ and using property (P₃), it can be said that $\|\Phi_I^T \mathbf{u}\|_2 \geq \sqrt{(1-\delta)}\|\mathbf{u}\|_2$. This makes

$$\mathbb{P} \left\{ J_W \middle|_{|J_W|=k} \middle| \Sigma \right\} \leq \mathbb{P}^k \left\{ |\langle \varphi, \mathbf{u} \rangle| \geq \sqrt{\frac{(1-\delta)}{m}} \middle| \Sigma \right\}.$$

Thus by using the property (P₂) of sensing matrices, i.e. the Gaussian tail probability, it can be written that

$$\mathbb{P} \left\{ J_W \middle|_{|J_W|=k} \middle| \Sigma \right\} \leq \left[2e^{-c_2 \frac{(1-\delta)}{m} d} \right]^k. \quad (\text{Eq. 6.4})$$

Using this bound of the conditional failure probability of equation (Eq. 6.4), the combination inequality $\binom{A}{B} \leq \left(\frac{eA}{B}\right)^B$, and equation (Eq. 6.3), it can be written that

$$\begin{aligned} \mathbb{P}(E_{\text{fail}} | \Sigma) &\leq \sum_{k=\lfloor \alpha m \rfloor + 1}^{\lfloor \alpha m \rfloor + m} \left[\frac{e(K-m)}{k} \cdot 2e^{-c_2 \frac{(1-\delta)}{m} d} \right]^k \\ &= \sum_{k=\lfloor \alpha m \rfloor + 1}^{\lfloor \alpha m \rfloor + m} e^{\left[\ln \frac{2e(K-m)}{k} - c_2 \frac{(1-\delta)}{m} d \right] k}. \end{aligned}$$

Changing the variables $i = k - \lfloor \alpha m \rfloor$ and $c_3 = c_2(1 - \delta)$,

$$\begin{aligned}
 \mathbb{P}(E_{\text{fail}}|\Sigma) &\leq \sum_{i=1}^m e^{\left[\ln \frac{2e(K-m)}{\lfloor \alpha m \rfloor + i} - c_3 \frac{d}{m}\right](\lfloor \alpha m \rfloor + i)} \\
 &\leq m e^{\left[\ln \frac{2e(K-m)}{\lfloor \alpha m \rfloor + 1} - c_3 \frac{d}{m}\right](\lfloor \alpha m \rfloor + 1)} \\
 &= e^{\left[\ln \frac{2e(K-m)}{\lfloor \alpha m \rfloor + 1} + \frac{\ln m}{\lfloor \alpha m \rfloor + 1} - c_3 \frac{d}{m}\right](\lfloor \alpha m \rfloor + 1)} \tag{Eq. 6.5}
 \end{aligned}$$

In the range of $m \geq 1$ and $0 \leq \alpha \leq 1$, it can be found that $\frac{\ln m}{\lfloor \alpha m \rfloor + 1} \leq \ln \frac{2m}{\lfloor \alpha m \rfloor + 1}$. Please refer to the appendix for the derivation of this inequality. Thus, the above upper bound can be expressed as

$$\mathbb{P}(E_{\text{fail}}|\Sigma) \leq e^{\left[\ln \frac{4e(K-m)m}{(\lfloor \alpha m \rfloor + 1)^2} - c_3 \frac{d}{m}\right](\lfloor \alpha m \rfloor + 1)}$$

Using the fact $(K - m)m \leq K^2/4$, it can be stated that

$$\mathbb{P}(E_{\text{fail}}|\Sigma) \leq e^{\left[2 \ln \frac{K}{\lfloor \alpha m \rfloor + 1} + 1 - c_3 \frac{d}{m}\right](\lfloor \alpha m \rfloor + 1)} \tag{Eq. 6.6}$$

The dominant variable term absorbs the constant, hence it can be stated that $2 \ln \frac{K}{\lfloor \alpha m \rfloor + 1} + 1 \leq C_4 \ln \frac{K}{\lfloor \alpha m \rfloor + 1}$. By taking $d \geq C_0 m \ln \frac{K}{\lfloor \alpha m \rfloor + 1}$ for $C_0 \geq \frac{C_4}{c_3}$, a failure probability $\mathbb{P}(E_{\text{fail}}|\Sigma) \leq e^{-c_0 \frac{d}{m}(\lfloor \alpha m \rfloor + 1)}$ can be ensured, where $c_0 \geq c_3 - \frac{C_4}{C_0}$. Using (Eq. 6.2), it can be said that OMP_α will succeed with probability $\mathbb{P}(E_{\text{succ}}) \geq 1 - e^{-c_0 \frac{d}{m}(\lfloor \alpha m \rfloor + 1)}$.

6.3.4 OMP as a Special Case

OMP can be viewed as a limiting case of OMP_α , where the extended run factor $\alpha = 0$. Thus, it should show its convergence to OMP. When it is stopped after m iterations $\mathbb{P}(E_{\text{fail}}|\Sigma)$ has a different form, which can be obtained by substituting $\alpha = 0$ in equation (Eq. 6.5):

$$\begin{aligned}
 \mathbb{P}(E_{\text{fail}}|\Sigma) &\leq e^{\left[\ln\{2e(K-m)\} + \ln m - c_3 \frac{d}{m}\right]} \\
 &\leq e^{\left[\ln\{2e(K-m)m\} - c_3 \frac{d}{m}\right]}
 \end{aligned}$$

Using the fact $(K - m)m \leq d^2/4$, it can be stated that

$$\mathbb{P}(E_{\text{fail}}|\Sigma) \leq e^{\left[\ln \frac{ed^2}{2} - c_3 \frac{d}{m}\right]} \leq e^{\left[2 \ln K + \ln \frac{e}{2} - c_2 \frac{d}{m}\right]} \quad (\text{Eq. 6.7})$$

The dominant variable term can absorb the constant, hence $2 \ln K + \ln \frac{e}{2} \leq C_4 \ln K$. By taking $d \geq C_0 m \ln K$ for $C_0 \geq \frac{C_4}{c_3}$, a failure probability $\mathbb{P}(E_{\text{fail}}|\Sigma) \leq e^{-c_0 \frac{d}{m}}$ can be ensured, where $c_0 \geq c_3 - \frac{C_4}{C_0}$. Using (Eq. 6.2), it can be said that OMP will succeed with probability $\mathbb{P}(E_{\text{succ}}) \geq 1 - e^{-c_0 \frac{d}{m}}$.

It serves as another validation of OMP_α , because the limiting result for $\alpha = 0$ coincides with the result of OMP in [15]. It also proves that OMP_α would require a reduced number of measurements for the same success probability.

6.4 Practical OMP_α

In order to simplify the explanation, OMP_α has been stated only with a simple halting criteria $t_{\text{max}} = m + \lfloor \alpha m \rfloor$. However, an additional halting criteria $\mathbf{r}_t = 0$ can be imposed to reduce the computational load without affecting the outcome.

Algorithm 6.4 (OMP_α with Less Computation) *The only change is at step vii of OMP_α algorithm (OMP_α for CS recovery):*

vii) Go to Step.2 if $t < m + \lfloor \alpha m \rfloor$ & $\mathbf{r}_t \neq 0$, else terminate;

It can easily be interpreted in the success scenario; i.e. $I \subseteq \Lambda_t$ for $t < m + \lfloor \alpha m \rfloor$, resulting in $\mathbf{r}_t = 0$. When continued after reaching $\mathbf{r}_t = 0$, algorithm 6.3 may either repeatedly reselect an atom till it reaches $t = m + \lfloor \alpha m \rfloor$, or it may select some more wrong atoms to form $\Lambda_{m+\lfloor \alpha m \rfloor}$. However, the outcome of algorithm 6.4 and algorithm 6.3 will be indifferent, as $I \subseteq \Lambda_t \subseteq \Lambda_{m+\lfloor \alpha m \rfloor}$ (it can easily be perceived from the proof of

proposition 6.2). Thus, the core idea of OMP_α to run OMP for $m + \lfloor \alpha m \rfloor$ iterations remains unaffected by algorithm 6.4.

A question may arise when after reaching $\mathbf{r}_t = 0$ algorithm 6.4 halts in the failure scenario; i.e. $I \not\subseteq \Lambda_t$ for $t < m + \lfloor \alpha m \rfloor$. One may wonder if proceeding further might have allowed OMP_α to obtain $I \subseteq \Lambda_t$. The following proposition shows that after arriving at a wrong solution, i.e. $\mathbf{r}_t = 0 : I \not\subseteq \Lambda_t$, running algorithm 6.3 further will never obtain the correct solution.

Proposition 6.3 *Take an arbitrary m -sparse signal $\mathbf{s}$ in $\mathbb{R}^K$, let Φ be a $d \times K$ measurement ensemble satisfying RIP of order $m + \lfloor \alpha m \rfloor$, and execute OMP_α with the data $\mathbf{z} = \Phi \mathbf{s}$. If OMP_α arrives at $\mathbf{r}_t = 0 : m < t < m + \lfloor \alpha m \rfloor$, and $I \not\subseteq \Lambda_t$, then it has already selected more than $\lfloor \alpha m \rfloor$ wrong atoms. Thus, by completing $m + \lfloor \alpha m \rfloor$ selections it will never achieve $I \subseteq \Lambda_t$.*

Proof: If signal residue vanishes i.e. $\mathbf{r}_t = 0$ after any t iterations, that means a t -sparse solution $\mathbf{z} = \Phi \hat{\mathbf{s}}$ is obtained. Let's assume that in this t -sparse solutions p such atoms are obtained which are not from Φ_I . As there exists a generating m -sparse solution $\mathbf{s}$ using atoms of Φ_I , it can be stated that $\Phi(\hat{\mathbf{s}} - \mathbf{s}) = 0$, where the signal $(\hat{\mathbf{s}} - \mathbf{s})$ has $p + m$ nonzero coefficients i.e. $\|\hat{\mathbf{s}} - \mathbf{s}\|_0 = p + m$. It implies, Φ contains $p + m$ linearly dependent atoms, which is only possible if $p > \lfloor \alpha m \rfloor$ because Φ obeys RIP of order $m + \lfloor \alpha m \rfloor$. Hence it is proved that OMP_α has already selected more than $\lfloor \alpha m \rfloor$ wrong atoms. Thus, by completing $m + \lfloor \alpha m \rfloor$ selections it will never achieve $I \subseteq \Lambda_t$.

It may be concluded that by halting at $\mathbf{r}_t = 0$, the outcome of algorithm 6.3 is not being changed. OMP_α succeeds only when all m correct atoms are inside it's selection. OMP_α will fail in all the events when more than $\lfloor \alpha m \rfloor$ wrong atoms are selected. Being pessimistic in the analysis, all possible events of wrong selection exceeding $\lfloor \alpha m \rfloor$ is taken in equation (Eq. 6.3). However, if algorithm 6.4 halts at $\lfloor \alpha m \rfloor + m'$, considering only the

events of wrong selection $[\lfloor \alpha m \rfloor + 1, \lfloor \alpha m \rfloor + m'] : m' \leq m$, would not affect the proof of theorem 6.1. Because, it would have replaced the term $\frac{\ln m}{\lfloor \alpha m \rfloor + 1}$ with $\frac{\ln m'}{\lfloor \alpha m \rfloor + 1}$ in equation (Eq. 6.5), which still would satisfy the upper bound in equation (Eq. 6.6).

6.4.1 OMP_α without Prior Knowledge of Sparsity (OMP_∞)

The superior execution speed of OMP comes with two drawbacks in its present form of CS recovery. First, it needs more number of measurements in comparison to BP for recovering the same signal. Second, it requires prior knowledge of the sparsity m , whereas no such information is needed for BP. Through the scheme of OMP_α , the gap between OMP and BP is brought down in terms of required d both in theory and practice. However, the dependence on knowledge of m still remains.

In principle, the unnecessary bound on the number of iterations can be removed in OMP_α , which requires prior knowledge of m . The bound of $m + \lfloor \alpha m \rfloor$ iterations for $\alpha \in [0, 1]$ is only required to prove its mathematical stance (Theorem 6.1). Even if the possibility of improvement is ignored, going for more iterations will never degrade the performance of OMP. Thus, iteration number based halting criteria can be removed from step 7 of algorithm 6.4.

Algorithm 6.5 (OMP_∞ with No Prior Information) *The only change is at step vii of OMP_α algorithm (OMP_α for CS recovery):*

vii) Go to Step.2 if $t < d$ & $\mathbf{r}_t \neq 0$, else terminate;

Algorithm 6.5 will never get trapped in an infinite loop, but will always converge with surety. Since OMP always selects a set of linearly independent atoms, so in the worst case scenario, it may end up selecting d linearly independent vectors that spans the whole $\mathbb{R}^d$ space to reach $\mathbf{r}_d = 0$. However, it may result in computational complexity of order $O(d^2 K)$, which is still less than BP.

Corollary 6.1 (OMP with Admissible Measurements) *Choose $d \geq C_1 m \ln \frac{K}{m}$. Suppose that $\mathbf{s}$ is an arbitrary m -sparse signal in $\mathbb{R}^K$, and draw a random $d \times K$ admissible measurement matrix Φ independent from the signal. Given the data $\mathbf{z} = \Phi \mathbf{s}$, OMP can reconstruct the signal with probability exceeding $1 - e^{-c_1 d}$ in at most d iterations.*

Execution of OMP_∞ can be viewed as running $\lim_{\alpha \rightarrow \infty} \text{OMP}_\alpha$. Consider an inadequate number of measurements d_0 for some sparsity m_0 , and let's interpret the outcome with increasing α . It can be observed from equation (Eq. 6.6) that the conditional failure probability $\mathbb{P}(E_{\text{fail}}|\Sigma) \approx 1$, till it reaches

$$\frac{1}{2} \left(c_3 \frac{d_0}{m_0} - 1 \right) > \ln \frac{K}{\lfloor \alpha m_0 \rfloor + 1}.$$

Afterwards, it will start decaying exponentially with α , which can be continuously approximated as

$$\mathbb{P}(E_{\text{fail}}|\Sigma) \leq e^{-c_5 \left(\alpha + \frac{1}{m_0} \right) d_0}.$$

Here $c_5 = c_3 - \frac{m_0}{d_0} \left(2 \ln \frac{K}{\lfloor \alpha m_0 \rfloor + 1} + 1 \right)$. However, since $\mathbb{P}(E_{\text{succ}}, \Sigma^c) \rightarrow 0$ and may be ignored, the final probability of successful recovery of a sparse vector can be expressed as

$$\mathbb{P}(E_{\text{succ}}) \simeq \mathbb{P}(E_{\text{succ}}, \Sigma) = \mathbb{P}(\Sigma) (1 - \mathbb{P}(E_{\text{fail}}|\Sigma)).$$

While increasing α , a point will be achieved where $\mathbb{P}(E_{\text{fail}}|\Sigma) \rightarrow 0$, and the final success probability

$$\mathbb{P}(E_{\text{succ}}) \simeq \mathbb{P}(\Sigma),$$

which can be verified from Fig.6.1.

In other words, success of OMP_∞ depends on the probability that Φ obeys a RIP of order $2m$. In the case of Gaussian random matrices, RIP of order $2m$ holds for entire range of $m \in (0, K/2)$ with high probability exceeding $1 - e^{-c_1 d}$, if $d \geq C_1 m \ln \frac{K}{m}$.

Hence, OMP_∞ will serve as a greedy alternative to BP, which has lesser computations. It maximizes the performance of OMP without any prior knowledge of m .

Figure 6.2: (A) The percentage of input signals of dimension $K = 256$ exactly recovered as a function of numbers of measurements (d) for different sparsity level (m). (B) The minimum number of measurements d required to recover any m -sparse signal of dimension $K = 256$ at least 95% of the time.

6.5 Experiments

The proposed extension of OMP is validated in this section. It is experimentally illustrated that OMP_α has not only improved the performance of OMP but also it has been competitive to BP. As per theorem 6.1, the algorithm is validated on random sensing matrices. The obtained results for Bernoulli ensemble are strikingly indifferent to Gaussian, thus only the results on Gaussian ensemble are presented. The practical question is to determine how many measurements d are needed to recover a m -sparse signal in $\mathbb{R}^K$ with high probability. Thus the experimental set up is the following.

The probability of success is viewed as the percent of a m -sparse signal recovered successfully out of 1000 trials, where successful recovery means the distance between the original and recovered sparse signal is insignificant i.e. $\|\hat{\mathbf{s}} - \mathbf{s}\|_2 \leq 10^{-6}$. For each trial the m -sparse signal $\mathbf{s}$ is generated by setting nonzero values at m random locations of a K -dimensional null vector. The measurement matrix Φ is constructed by generating $d \times K$ Gaussian random variables of parameters $(0, 1/\sqrt{d})$. The recovered signal $\hat{\mathbf{s}}$ is obtained performing BP, OMP, OMP_α and OMP_∞ on the measurement $\mathbf{z} = \Phi\mathbf{s}$. Though it is possible to obtain different set of results in OMP_α by varying the extended run factor $0 < \alpha \leq 1$, but the results presented here are for $\alpha = 0.25$.

Table 6.1: Linear Fitting of Fig. 6.2(B)

Algorithm	Expression
OMP	$1.504m \ln K + 9.0$
OMP_α	$1.288m \ln \left(\frac{K}{0.25m+1} \right) + 14.87$
OMP_∞	$1.962m \ln \left(\frac{K}{m} \right) + 3.134$
BP	$1.596m \ln \left(\frac{K}{m} \right) + 0.991$

The nonzero coefficients in $\mathbf{s}$ play an important role in the performance of matching based greedy algorithms from a practical point of view. The measurement matrix Φ is obtained using zero mean random variables. Thus, when all the nonzero coefficients

become equal, the measurement $\mathbf{z} = \Phi \mathbf{s}$ becomes the scaled sample mean of the random variables making it very close to zero i.e. $\mathbf{z} \rightarrow 0$. This scenario degrades the performance of the matching step of the algorithm depending on the precision of the computer. Hence, all the results are obtained for this extreme scenario, when the sparse coefficients are set equal i.e. $\mathbf{s}_I = 1$ (same as the experimental setup in [15]).

Signal dimension is taken as $K = 256$ and each m -sparse signal is recovered from the number of measurements starting with $d = 4$ to $d = 256$ in steps of 4. The percentage of successful trials is plotted against measurement (d) in plot (A) of Fig.6.2.

With the same philosophy it is interesting to know, for a given sparsity level how many measurements will be needed to ensure a recovery with certain probability of success (for example 0.95 or 95%). As the %-success vs. d is increasing in nature, the number of measurement (d) can empirically be obtained where it first achieved success rate of 95%. Plot (B) of Fig.6.2 shows the plot d vs. m for 95% success. In order to study the characteristic of d vs. m data points, a linear curve fitting is done using Matlab toolbox. The results are tabulated in Table.6.1, which shows $O(m \ln K)$ nature of OMP and $O(m \ln \frac{K}{\alpha m + 1})$ nature of OMP_α , but $O(m \ln \frac{K}{m})$ nature of OMP_∞ and BP.

In order to validate theorem 6.1, the curve fitting result for OMP_α is obtained for $\alpha = 0, 1/16, 1/8, 1/4, 1/2$ in similar manner. However, the signal dimension is increased to $K = 1024$, which is to acquire more integer points for better curve fitting. Fig. 6.3 shows a tight fitting of the curve $C_0 m \ln \frac{K}{\alpha m + 1} + C_6$ with the obtained data points, and the values of C_0 and C_6 for various α are tabulated in Table. 6.2.

Table 6.2: Linear Fitting of $C_0 m \ln \frac{K}{\alpha m + 1} + C_6$ in Fig. 6.3

α	0	1/16	1/8	1/4	1/2
C_0	1.418	1.089	1.119	1.199	1.434
C_6	17.73	43.17	33.73	29.25	13.84

Figure 6.3: The minimum number of measurements (d) required to recover an m -sparse signal of dimension $K = 1024$ at least 95% of the time.

6.6 Discussions

Greedy pursuit is advantageous in terms of computational cost, which interests researchers to improve its performance towards the benchmark of convex relaxation (BP). The proposed OMP_α uses the orthogonality property of OMP and the probabilistic linear independences of random ensemble to enhance its performance. Its required number of measurements for high probability signal recovery follows a logarithmic trend like BP, instead of linear trend as OMP. Further, the proposed OMP_∞ shows an overwhelming

improvement in OMP by bringing it close to BP in terms of both required order of measurements and knowledge of sparsity. The theoretical guarantee of OMP_α along with the obtained empirical results make OMP_α more compelling.

Convex relaxation has rich varieties of results including the cases when the measured signal is not exactly sparse or is contaminated by noise. The presented results for OMP_α are focused on strictly sparse signals, and how OMP_α behaves recovering the measurements contaminated by noise is an interesting direction to pursue.

6.7 Summary

OMP for CS recovery of the sparse signals is analyzed in depth, where a proposition is stated to highlight the behavior of OMP. As a result of this analysis, an extended run of OMP called OMP_α is proposed to improve the CS recovery performance of the sparse signals. A proposition is stated to describe the events of success and failure for OMP_α , which leads to the analysis of its recovery performance. Through the event analysis of OMP_α , the required number of measurements for exact recovery is derived, which is in the same order as that of BP. The motivation of extended run results in another scheme call OMP_∞ that does not need any prior knowledge of sparsity similar to BP. A corollary is stated showing the required number of measurements for OMP_∞ is tending to that of BP. Through these results of OMP_α and OMP_∞ , OMP can successfully compete with BP in terms of required number of measurements, as well as in terms of the philosophy of not being aware of sparsity.

Chapter 7

Summary and Future Work

This chapter summarizes the works presented in the thesis. It also gives some possible future works from the works presented.

7.1 Summary

The works presented in the thesis revolve around sparsity. When a signal becomes sparse in a transform domain or in a dictionary, many signal processing problems can be solved taking sparsity *a priori*. Alongside, the sparse representation of the signal reveals that the signal can be compressed. The trending field of research is to acquire the sparse signal efficiently through compressed sensing. Hence, the thesis starts with its contributions to the field of sparse representation of the signal and its application. Next it presents the contribution with a major focus on reconstructing the sparse signal from its compressed sensing measurements. The thesis can be summarized as follows:

- The dictionary training algorithms MOD and K -SVD are presented in line with K -means clustering for VQ. It is shown that MOD simplifies to K -means, while K -SVD fails to simplify due to its principle of updating. As MOD does not need to update the sparse representation vector during dictionary update stage, it is compatible to any *structured/constrained* sparsity model such as K -means. However,

since MOD is not sequential, a sequential generalization to K -means is proposed that avoids the difficulties of K -SVD. Computational complexity for all algorithms are derived, and MOD is shown to be the least complex followed by SGK under a dimensionality condition, which is true for many practical applications. Through synthetic data experiment, it is shown that all the algorithms perform equally well with marginal differences. Thus, MOD being the fastest among all, remains the dictionary training algorithm of choice for any kind of sparse representation. However, if sequential update becomes essential, SGK should be chosen.

- Through a framework of image compression the advantage of SGK over K -SVD is highlighted. The effectiveness of SGK in the image inpainting framework is also validated. To further illustrate the effectiveness of SGK in practice, it is incorporated into the framework of image denoising via sparse representation. SGK is shown to be a simpler and intuitive implementation compared to K -SVD. Through rigorous experiments it is shown that SGK performs as effectively as K -SVD, and needs lesser computations. Hence, K -SVD can be replaced with SGK in the image denoising framework and all its extensions. Similarly, it is also possible to extend the use of SGK to other applications of sparse representation.
- Image recovery using local sparse representation is illustrated in a framework of location adaptive block size selection. This framework is motivated by the importance of block size selection in inferring the geometrical structures and the details in the image. First, it clusters the image based on block size selected at each location to minimize the local MSE. Subsequently, it aggregates all the estimated image blocks of respective sizes to estimate the final image. By experimenting on some well known images, the potential of the proposed framework is illustrated in

comparison to the state of the art image recovery techniques. Although the recovery of gray scale images are only addressed, the framework can also be extended to color images. It can be said that the present work provides stimulating results with an intuitive platform for further investigation.

- In order to improve the performance of OMP towards the benchmark of convex relaxation (BP), OMP_α is proposed. OMP_α uses the orthogonality property of OMP and the probabilistic linear independences of random ensemble to enhance its performance. It is shown that OMP_α 's required number of measurements for high probability signal recovery follows a logarithmic trend like BP, instead of a linear trend as OMP. Further, OMP_∞ is proposed as a simple extension of OMP_α . It is shown that OMP_∞ brings an overwhelming improvement in OMP by bringing it close to BP, both in terms of required order of measurements, and in not requiring prior knowledge of sparsity. The theoretical guarantee of OMP_α along with the obtained empirical results make OMP_α more compelling.

7.2 Future Work

Some of the possible interesting future directions based on the thesis are as follows.

- In the practical problems, a sparsifying dictionary is obtained for a given set of training signals. The outcome of the dictionary training is greatly influenced by the choice of initial dictionary. However, the atom-by-atom sequential update gives a freedom to reinitialize an atom individually instead of updating it. In the case when the update of an atom does not provide much of an improvement in the MSE, a strategic reinitialization may produce a better dictionary.
- Similarly, when the training signals are contaminated by noise, there is a good chance of noise being adapted to the dictionary atoms. Thus by taking advantage

of the sequential update, a noise handling scheme needs to be introduced, which can avoid the noise incursion.

- Though, the intention is not to propose any new image compression framework in chapter 4, certain things can be optimized for a better compression. For simplicity, a uniform quantization of the coefficients is used; and a simple coding is used to store the number of coefficients, the indices, and the coefficients. However, a better quantization strategy with entropy coding can further improve the compression ratio/BPP.
- In the present framework of chapter 5, the block sizes are prefixed. However, the bounds on the local block size is an interesting topic to explore further. In the present framework of aggregation, all the pixels of the recovered blocks are given equal weight. An improvement may be achieved by deriving an aggregation formula with adaptive weights per pixel for the recovered local window.
- The results for OMP_α in chapter 6 are focused on strictly sparse signals. The decay of MSE in the case of recovering not exactly sparse but compressible signals using OMP_α can be studied similar to other greedy pursuits. Also, the recovery of measurements contaminated by noise is an interesting direction to pursue.

Appendix

For an appropriate c_7 ,

$$\frac{\ln m}{\lfloor \alpha m \rfloor + 1} \leq \ln \frac{c_7 m}{\lfloor \alpha m \rfloor + 1}, \quad (\text{Eq. 7.1})$$

where sparsity $m \geq 1$ and $0 \leq \alpha \leq 1$.

For $m = 1$

Let's substitute the limiting value $m = 1$ in inequality (Eq. 7.1).

$$0 \leq \ln \frac{c_7}{\lfloor \alpha \rfloor + 1} \implies c_7 \geq \lfloor \alpha \rfloor + 1.$$

As $\alpha \leq 1$, inequality (Eq. 7.1) will be true for $c_7 \geq 2$.

For $m \geq 2$

The inequality (Eq. 7.1) can be rearranged as following

$$\begin{aligned} \ln \frac{\lfloor \alpha m \rfloor + 1}{c_7} &\leq \left(1 - \frac{1}{\lfloor \alpha m \rfloor + 1}\right) \ln m \\ \implies \log_m \frac{\lfloor \alpha m \rfloor + 1}{c_7} &\leq \left(1 - \frac{1}{\lfloor \alpha m \rfloor + 1}\right) \\ \implies \frac{\lfloor \alpha m \rfloor + 1}{c_7} &\leq \frac{m}{m^{\frac{1}{\lfloor \alpha m \rfloor + 1}}} \\ \implies c_7 &\geq \frac{(\lfloor \alpha m \rfloor + 1)m^{\frac{1}{\lfloor \alpha m \rfloor + 1}}}{m} \end{aligned} \quad (\text{Eq. 7.2})$$

Interestingly, the condition on c_7 is a function of α and m , $f(m, \alpha) = \frac{(\alpha m + 1)m^{\frac{1}{\alpha m + 1}}}{m}$. For any give m , if we set

$$c_7 \geq \max_{0 \leq \alpha \leq 1} f(m, \alpha) \quad (\text{Eq. 7.3})$$

inequality (Eq. 7.1) would be valid for all range of $\alpha \in [0, 1]$. It can be seen that

$$\begin{aligned}\frac{\partial f(m, \alpha)}{\partial \alpha} &= m^{\frac{1}{\alpha m + 1}} \left[1 - \frac{\ln m}{\alpha m + 1} \right] \\ &< 0 \text{ for } \alpha < \frac{\ln m - 1}{m} \\ &= 0 \text{ at } \alpha = \frac{\ln m - 1}{m} \\ &> 0 \text{ for } \alpha > \frac{\ln m - 1}{m}.\end{aligned}$$

This implies, $f(m, \alpha)$ decreases with α until $\alpha = \frac{\ln m - 1}{m}$, and then increases. However, $f(m, \alpha)$ is a monotonically increasing function of α for $m < e$, because $\ln m < 1$ makes $\frac{\partial f(m, \alpha)}{\partial \alpha} > 0$ unconditionally. Hence,

$$c_7 \geq \max \{f(m, 0), f(m, 1)\} = f(m, 1) \quad (\text{Eq. 7.4})$$

since

$$f(m, 1) = \left(1 + \frac{1}{m}\right) m^{\frac{1}{m+1}} \geq 1 = f(m, 0).$$

If we set

$$c_7 \geq \max_{2 \leq m} f(m, 1) \quad (\text{Eq. 7.5})$$

inequality (Eq. 7.1) would be valid for all $m \geq 2$. The derivative

$$\frac{\partial f(m, 1)}{\partial m} = \frac{(m+1)m^{\frac{1}{m+1}}}{m} \left[\frac{-\ln m}{(m+1)^2} \right] < 0$$

shows that $f(m, 1)$ is a decreasing function of m . Hence,

$$c_7 \geq \max_{2 \leq m} f(m, 1) = f(2, 1) = \frac{3}{2} 2^{\frac{1}{3}}.$$

However, the previously obtained condition $c_7 \geq 2$ for the case of $m = 1$, is higher than $\frac{3}{2} 2^{\frac{1}{3}}$. Therefore, it is proved that at $c_7 = 2$ the inequality (Eq. 7.1) holds for the entire range of m and α .

Author's Publications

Journal papers

[J1] 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, pp. 587-590, 2013.

Conference papers

[C1] B.J. Falkowski, S.K. Sahoo, and T. Luba, “Two novel methods for lossless compression of fluorescent dye cell images”, *IEEE International Conference on Mixed Design of Integrated Circuits and Systems (MIXDES)*, Lodz, Poland, Jun. 2009.

[C2] S.K. Sahoo, W. Lu, S.D. Teddy, D. Kim, M. Feng, “Detection of atrial fibrillation from non-episodic ECG Data: a review of methods”, *33rd International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Boston, Aug. 2011.

[C3] S.K. Sahoo and W. Lu, “Image inpainting using sparse approximation with adaptive window selection”, *IEEE International Symposium on Intelligent Signal Processing (WISP)*, Floriana, Malta, Sep. 2011.

[C4] S.K. Sahoo and W. Lu, “Image denoising using sparse approximation with adaptive window selection”, *International Conference on Information Communication Signal Processing (ICICS)*, Singapore, Dec. 2011.

[C5] S.K. Sahoo and A. Makur, “Image Denoising Via Sparse Representations Over Sequential Generalization of K -means (SGK)”, *International Conference on Information*

Communication Signal Processing (ICICS), Taiwan, Dec. 2013.

[C6] S. Narayanan, S.K. Sahoo and A. Makur, “Modified Adaptive Basis Pursuits for Recovery of Correlated Sparse Signals”, *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Florence, Italy, May. 2014.

References

- [1] J. Ziv and A. Lempel, “A universal algorithm for sequential data compression,” *Information Theory, IEEE Transactions on*, vol. 23, no. 3, pp. 337–343, 1977.
- [2] T. Welch, “A technique for high-performance data compression,” *Computer*, vol. 17, no. 6, pp. 8–19, 1984.
- [3] M. Nelson and J.-L. Gailly, *The data compression book*. M&T Books, 1996.
- [4] S. Mallat, *A Wavelet Tour of Signal Processing*. Elsevier Inc., 2009.
- [5] M. Marcellin, M. Gormish, A. Bilgin, and M. Boliek, “An overview of jpeg-2000,” in *Data Compression Conference, 2000. Proceedings. DCC 2000*, pp. 523–541, 2000.
- [6] K. Engan, S. O. Aase, and J. H. Husøy, “Multi-frame compression: theory and design,” *Signal Processing*, vol. 80, no. 10, pp. 2121 – 2140, 2000.
- [7] S. Lesage, R. Gribonval, F. Bimbot, and L. Benaroya, “Learning unions of orthonormal bases with thresholded singular value decomposition,” in *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP ’05). IEEE International Conference on*, vol. 5, 2005.
- [8] R. Vidal, Y. Ma, and S. Sastry, “Generalized principal component analysis (gpca),” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 12, pp. 1945–1959, 2005.
- [9] M. Aharon, M. Elad, and A. Bruckstein, “k -svd: An algorithm for designing over-complete dictionaries for sparse representation,” *IEEE Trans. Signal Processing*, vol. 54, pp. 4311–4322, November 2006.

REFERENCES

- [10] R. Rubinstein, A. Bruckstein, and M. Elad, “Dictionaries for sparse representation modeling,” *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1045–1057, 2010.
- [11] M. Elad and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” *Image Processing, IEEE Transactions on*, vol. 15, no. 12, pp. 3736–3745, 2006.
- [12] M. Elad, J.-L. Starck, P. Querre, and D. Donoho, “Simultaneous cartoon and texture image inpainting using morphological component analysis (mca),” *Applied and Computational Harmonic Analysis*, vol. 19, no. 3, pp. 340 – 358, 2005.
- [13] M. Fadili, J.-L. Starck, and F. Murtagh, “Inpainting and zooming using sparse representations,” *The Computer Journal*, vol. 52, no. 1, pp. 64–79, 2009.
- [14] E. Candes and M. Wakin, “An introduction to compressive sampling,” *Signal Processing Magazine, IEEE*, vol. 25, pp. 21 –30, march 2008.
- [15] J. Tropp and A. Gilbert, “Signal recovery from random measurements via orthogonal matching pursuit,” *Information Theory, IEEE Transactions on*, vol. 53, pp. 4655 –4666, dec. 2007.
- [16] S. Sardy, A. G. Bruce, and P. Tseng, “Block coordinate relaxation methods for non-parametric wavelet denoising,” *Journal of Computational and Graphical Statistics*, vol. 9, no. 2, pp. 361–379, 2000.
- [17] A. Gersho and R. M. Gray, *Vector quantization and signal compression*. Norwell, MA, USA: Kluwer Academic Publishers, 1991.
- [18] I. Daubechies, “Time-frequency localization operators: a geometric phase space approach,” *Information Theory, IEEE Transactions on*, vol. 34, no. 4, pp. 605–612, 1988.
- [19] R. Coifman and M. Wickerhauser, “Entropy-based algorithms for best basis selection,” *Information Theory, IEEE Transactions on*, vol. 38, no. 2, pp. 713–718, 1992.
- [20] S. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries,” *Signal Processing, IEEE Transactions on*, vol. 41, pp. 3397 –3415, dec 1993.

REFERENCES

- [21] Y. C. Pati, R. Rezaifar, and P. S. Krishnaprasad, “Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition,” in *Conference Record of the Asilomar Conference on Signals, Systems & Computers*, vol. 1, pp. 40–44, 1993.
- [22] I. Gorodnitsky and B. Rao, “Sparse signal reconstruction from limited data using focuss: a re-weighted minimum norm algorithm,” *Signal Processing, IEEE Transactions on*, vol. 45, no. 3, pp. 600–616, 1997.
- [23] S. S. Chen, D. L. Donoho, and M. A. Saunders, “Atomic decomposition by basis pursuit,” *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.
- [24] B. Rao, K. Engan, S. Cotter, J. Palmer, and K. Kreutz-Delgado, “Subset selection in noise based on diversity measure minimization,” *Signal Processing, IEEE Transactions on*, vol. 51, no. 3, pp. 760–770, 2003.
- [25] A. Bugeau, M. Bertalmio, V. Caselles, and G. Sapiro, “A comprehensive framework for image inpainting,” *Image Processing, IEEE Transactions on*, vol. 19, no. 10, pp. 2634–2645, 2010.
- [26] P. Arias, G. Facciolo, V. Caselles, and G. Sapiro, “A variational framework for exemplar-based image inpainting,” *International Journal of Computer Vision*, vol. 93, no. 3, pp. 319–347, 2011.
- [27] A. Buades, B. Coll, and J. Morel, “A review of image denoising algorithms, with a new one,” *Multiscale Modeling and Simulation*, vol. 4, no. 2, pp. 490–530, 2005.
- [28] D. L. Donoho and I. M. Johnstone, “Adapting to unknown smoothness via wavelet shrinkage,” *Journal of the American Statistical Association*, vol. 90, pp. 1200–1224, December 1995.
- [29] P. Chatterjee and P. Milanfar, “Clustering-based denoising with locally learned dictionaries,” *IEEE Trans. Image Processing*, vol. 18, pp. 1438–1451, July 2009.
- [30] E. Candes and T. Tao, “Decoding by linear programming,” *Information Theory, IEEE Transactions on*, vol. 51, pp. 4203 – 4215, dec. 2005.

REFERENCES

- [31] W. Dai and O. Milenkovic, “Subspace pursuit for compressive sensing signal reconstruction,” *Information Theory, IEEE Transactions on*, vol. 55, pp. 2230–2249, may 2009.
- [32] E. J. Candes, “The restricted isometry property and its implications for compressed sensing,” *Comptes Rendus Mathematique*, vol. 346, no. 910, pp. 589–592, 2008.
- [33] N. Yurii and N. Arkadii, *Interior-Point Polynomial Algorithms in Convex Programming*. Society for Industrial and Applied Mathematics, 1994.
- [34] J. Tropp, “Greed is good: algorithmic results for sparse approximation,” *Information Theory, IEEE Transactions on*, vol. 50, pp. 2231–2242, oct. 2004.
- [35] B. Ake, *Numerical Methods for Least Squares Problems*. Society for Industrial and Applied Mathematics, 1996.
- [36] M. Davenport and M. Wakin, “Analysis of orthogonal matching pursuit using the restricted isometry property,” *Information Theory, IEEE Transactions on*, vol. 56, pp. 4395–4401, sept. 2010.
- [37] J. Wang and B. Shim, “On the recovery limit of sparse signals using orthogonal matching pursuit,” *Signal Processing, IEEE Transactions on*, vol. 60, pp. 4973–4976, sept. 2012.
- [38] B. N. Datta, *Numerical Linear Algebra and Applications, Second Edition*. SIAM, 2010.
- [39] R. Rubinstein, M. Zibulevsky, and M. Elad, “Efficient Implementation of the K-SVD Algorithm using Batch Orthogonal Matching Pursuit,” tech. rep., Apr. 2008.
- [40] J. Mairal, M. Elad, and G. Sapiro, “Sparse representation for color image restoration,” *Image Processing, IEEE Transactions on*, vol. 17, no. 1, pp. 53–69, 2008.
- [41] M. Protter and M. Elad, “Image sequence denoising via sparse and redundant representations,” *Image Processing, IEEE Transactions on*, vol. 18, no. 1, pp. 27–35, 2009.

REFERENCES

- [42] J. Mairal, G. Sapiro, and M. Elad, “Learning multiscale sparse representations for image and video restoration,” *Multiscale Modeling & Simulation*, vol. 7, no. 1, pp. 214–241, 2008.
- [43] V. Katkovnik, K. Egiazarian, and J. Astola, “Adaptive window size image de-noising based on intersection of confidence intervals (ici) rule,” *Journal of Mathematical Imaging and Vision*, vol. 16, pp. 223–235, May 2002.
- [44] W. Hoeffding, “Probability inequalities for sums of bounded random variables,” *Journal of the American statistical association*, vol. 58, no. 301, pp. 13–30, 1963.
- [45] J. Fadili, J.-L. Starck, M. Elad, and D. Donoho, “Mcalab: Reproducible research in signal and image decomposition and inpainting,” *Computing in Science Engineering*, vol. 12, pp. 44–63, Jan 2010.
- [46] D. Needell and R. Vershynin, “Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit,” *Selected Topics in Signal Processing, IEEE Journal of*, vol. 4, pp. 310–316, april 2010.
- [47] D. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, “Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit,” *Information Theory, IEEE Transactions on*, vol. 58, pp. 1094–1121, feb. 2012.
- [48] H. Huang and A. Makur, “Backtracking-based matching pursuit method for sparse signal reconstruction,” *Signal Processing Letters, IEEE*, vol. 18, pp. 391–394, july 2011.
- [49] Y. C. Eldar and G. Kutyniok, *Compressed Sensing: Theory and Applications*. Cambridge University Press, 2012.
- [50] D. L. Donoho, “For most large underdetermined systems of linear equations the minimal l_1 -norm solution is also the sparsest solution,” *Communications on Pure and Applied Mathematics*, vol. 59, no. 6, pp. 797–829, 2006.
- [51] J. Kahn, J. Komls, and E. Szemerdi, “On the probability that a random ± 1 -matrix is singular,” *Journal of the American Mathematical Society*, vol. 8, no. 1, pp. 223–240, 1995.

REFERENCES

- [52] E. D. Livshits, “On the efficiency of the orthogonal matching pursuit in compressed sensing,” *Sbornik: Mathematics*, vol. 203, no. 2, p. 183, 2012.
- [53] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, “A simple proof of the restricted isometry property for random matrices,” *Constructive Approximation*, vol. 28, pp. 253–263, 2008.