Sparsity and Manifold Methods in Image and Higher Dimensional Data Representation

BY

LINGDAO SHA
B.S., Beijing University of Posts & Telecom, Beijing, China, 2011
M.S., University of illinois at Chicago, Chicago, 2017

Submitted as partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Chicago, 2018

Chicago, Illinois

Defense Committee:

Dan Schonfeld, Advisor and Chair
Jing Wang, Co-Advisor and Co-Chair, Math, Stats, & Computer Sci
Peter Gann, Pathology
Joe Zhou, Radiology, Neurosurgery, and Bioengineering
Mojtaba Soltanalian

To my family,

for their love and support.

ACKNOWLEDGMENTS

I would like to thank my advisor, Prof. Dan Schonfeld for his guidance, help, vision and encouragement during my PhD research that will benefit me more in my future life.

I would also like to thank my co-advisor Prof. Jing Wang for her concern, help, and detail oriented guidance which leverages my mathematical skills. It is also my honor to work with Prof. Peter Gann on medical imaging related cancer prognosis problems, which attracts my interests and leads to my current career path. I would also like to extend my gratitude to my dissertation committee members Prof. Joe Zhou and Prof. Mojtaba Soltanalian for valuable discussions and suggestions on shaping my thesis.

Finally, I would like to thank my wife, for her love, understanding, and support, my family, for their supports at all times.

LS

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF TABLES

LIST OF FIGURES

FIGURE **PAGE**

LIST OF FIGURES (Continued)

FIGURE **PAGE**

LIST OF FIGURES (Continued)

FIGURE **PAGE**

LIST OF ABBREVIATIONS

SC	Sparse Coding
DL	Dictionary Learning
MAC	Morphological Component Analysis
OGLR	Optimal Graph Laplacian Regularizer
SSC	Simultaneous Sparse Coding
GSM	Gaussian Scale Mixture
SSC-GSM	Simultaneous Sparse Coding with GSM
BM3D	Block-matching and 3D Filtering
PSNR	Peak Signal-to-noise Ratio
NMF	Nonnegative Matrix Factorization
LARS	Least Angle Regression
Kron	Kronecker
Kron-LARS	Kronecker Least Angle Regression
NBS-LARS	N-dimensional Block Sparse Least Angle Regression
OMP	Orthogonal Matching Pursuit
NBOMP	N-dimensional Block Sparse OMP
MP	Matching Pursuit
CS	Compressed Sensing / Compressive Sensing
IR	Image Restoration
TV	Total Variation
ICA	Independent Component Analysis
KSVD	K Sigular Value Decomposition
PCA	Principle Component Analysis
GNMF	Graph Regularized NMF
GraphSC	Graph Regularized Sparse Coding
MAP	Maximum A Posteriori Estimation
DGSC	Dual Graph Regularized Sparse Coding
FS-LD	Feature-sign Search with Lagrange Dual

LIST OF ABBREVIATIONS (Continued)

LARS-BCD	Least Angle Regression with Block Coordinate Descent
SVM	Support Vector Machine
NMI	Normalized Mutual Information
WT	Wavelet Transform
DCT	Discrete Cosine Transform
NSP	Null Space Property
RIP	Restricted Isometry Property
CT	Computed Tomography
MRI	Magnetic Resonance Imaging
HCI	Hyperspectral Compressed Imaging
LLE	Locally Linear Embed
LLESC	Locally Linear Embedded Sparse Coding
MODL	Modified Online Dictionary Learning

SUMMARY

One of the most important concept underlines recent development of image and signal processing is sparsity. Specifically, most applications in getting signals/images of interest can be well approximated by a linear combination of few active elements of a dictionary. The dictionary or base is a super set that can be used to represent the internal structure of signals. Utilization of sparsity can simplify the problem of singals/images processing, storage and representation. However, getting a good sparsity representation and efficient recovery from sparse data can be a very complex problem. Therefore, great efforts have been devoted to find the optimal way of sparse representation as well as recovery algorithms based on different purposes of applications. This thesis is dedicated to the study of sparse representation in image processing (denoising, deblurring), image representation (clustering, classification), tensor compressed sensing, and medical image processing.

We first present a graph Laplacian regularized sparse coding method in image representation and restoration. As most existing sparse coding approaches failed to consider the fact that high dimensional data naturally reside on geometrical structure of the data space. In this thesis, we proposed a generalized framework for image restoration and representation by combining sparse coding and graph Laplacian algorithms. We show that by adding structural and high dimensional information as regularization terms, sparse representation can be boosted in terms of image processing and representation.

We then present a kronecker least angle regression (Kron-LARS) algorithm as a generalization of classical vector version least-angle regression (LARS) algorithm for solving underdetermined linear algebra problem. We demonstrated that our Kron-LARS algorithm can be used to efficiently solving high-dimensional data (tensor) compressed sensing problem where vector version LARS will be bottlenecked in terms of running memory and computational complexity. We also proposed a more efficient N dimensional block sparse LARS (NBS-LARS) by utilizing the block sparsity property of high-dimensional data.

Thirdly, we present a graph regularized sparse non-negative matrix factorization algorithm in medical image color normalization. We show that our proposed unsupervised color normaliztion algorithm outperforms existing popular algorithms such as ICA, PCA, NMF and SNMF both qualitatively and quantitively.

Finally, we present a locally linear embedded sparse coding algorithm for image representation. To solve the proposed sparse coding problem, we proposed an efficient modified online dictionary learning algorithm which converges faster than the existing algorithms for solving graph regularized sparse coding problem.

CHAPTER 1

INTRODUCTION

Sparse representation has draw great attention recently in the field of signal and image processing. Researchers found that natural signals/images are intrinsically sparse, where signals/images can be effectively represented with only few active elements (from a dictionary). Though attractive in concept, finding the optimal sparse representation and recover the original data is not a trivial problem. With the success of customized dictionary such as DCT (in JPEG) and Wavelet (in JPEG2000) in image compression, recently, dictionary learning algorithm such as KSVD and online dictionary learning have proved its better fidelity in terms image restoration and representation. However, most existing sparse representation approach failed to consider the fact that high dimensional data naturally reside on geometrical structure of the data space. A combination of sparse representation with graph regularizer becomes a natural derivation. Another successful application of sparse representation is compressed sensing, where sparsity property can be utilized from the data acquisition process. This is very important for applicatoins that is very expensive or time consuming in data retrival. However, when dealing high-dimentional data (most of the cases in real world signals), most existing compressed sensing algorithms relying converting high-dimensional data into vectors which is high memory and computational expensive. Utilizing the Kronecker structure of high-dimensional data becomes a natural thought. Finally, sparse representations are also widely used in medical image processing, as fast digtal scanners are becoming more and more popular in medical. In this thesis, we dedicated our efforts in optimizing the applications of sparse representation in the fields mentioned above. We proposed both regularization objective functions as well as optimization algorithms to get the best outputs based on assumptions.

The first part of this thesis is about graph Laplacian regularized sparse coding method in image restoration and representation. Sparse coding is widely used in image denoising, deblurring, clustering, and classification. However, most existing approaches to sparse coding failed to consider the fact that high dimensional data naturally reside on geometrical structure of the data space. It has been shown that geometric information of the data is important for both inversion and discrimination. Here, we proposed a generalized framework for image restoration and representation by combining sparse coding and graph algorithms. In image denoising and deblurring problems, an image is first decomposed into cartoon layer (piecewise-smooth contents) and texture layer (textures and sharp edges) using morphological component analysis (MCA); then optimal graph Laplacian regularizer (OGLR) algorithm and simultaneous sparse coding with gaussian scale mixture prior (SSC-GSM) algorithm are applied to cartoon layer and texture layer respectively; final restored image is by adding the outcomes from two algorithms. Our proposed hybrid image restoration algorithm outperforms state-of-the-art image denoising algorithms, such as BM3D on natural images, by 2 to 5 dB in terms of PSNR (not as good in SSIM index) depending on noise level, performs comparatively in terms of image deblurring. In image clustering and classification problems, our graph Laplacian regularized sparse coding framework demonstrate that high dimension problem can be converted to linear constraint by manifold learning and solved together by sparse coding. We convert our generalized framework into a novel dual graph regularized sparse coding method to transform the nonlinear data space and feature space into linear space for sparse coding. Two efficient optimization algorithms are provided for numerical implementation.

Experimental results show that our generalized graph Laplacian and sparse coding framework performs competitively with popular denoising, deblurring, clustering, and classification (none deep learning) methods.

The second part of this thesis is called tenser least-angle regression. Sparse representation of signals has drawn great attention recently under the assumption that signal can be well approximated by a simple linear combination of few active elements (from a basis). Many algorithms have been proposed to find the sparsest solution of a signal by solving an underdetermined linear system of algebraic equations. Among them, one of the most important algorithm is called Least Angle Regression (LARS), which can be used to solve least square problem with both ℓ_0 norm and ℓ_1 norm constrain. However, most modern signals are two dimensional or even higher, current state-of-the-art solutions are relying on converting high dimensional into vectors and then solve the equation accordingly. Research has shown that signals with multidimensional structure can be converted from vectors to tenors (multiway arrays) by using the Tucker model. Thus, the goal is to solve a underdetermined linear system of equations possessing a Kronecker structure. Here, we proposed Kronecker Least Angle Regression (Kron-LARS) algorithm as a generalization of the classic vector version (LARS) algorithm for tensors. We demonstrate that by utilizing the multidimensional structure of signal, Kron-LARS as an equivalent conversion of LARS can reduce the recovery complexity and memory usage significantly. Additionally, by exploiting not only the Kronecker structure but also block sparsity of signals, our Kron-LARS can be easily extended as N-dimensional block sparse LARS (NBS-LARS), which is dramatically fast and memory efficient. We theoretically demonstrate that NBS-LARS algorithm not only has considerably lower recovery complexity but also has better precision under same percentage of sampling.

The third part of this thesis is a graph regularized non-negative matrix factorization algorithm in medical image processing. Computer based automatic medical image processing and quantification are becoming popular in digital pathology. However, preparation of histology slides can vary widely due to differences in staining equipment, procedures and reagents, which can reduce the accuracy of algorithms that analyze their color and texture information. To reduce the unwanted color variations, we proposed an unsupervised color normalization algorithm. We validated the performance of proposed algorithm qualitatively (applying it on several difficult cases and compare with other algorithms).

The fourth part of this thesis is an locally linear embedded sparse coding algorithm for image clustering and classification. Recently, sparse coding has been widely and successfully used in image classification, noise reduction, texture synthesis and audio processing. Although traditional sparse coding method with fixed dictionaries like wavelet and curvelet can produce promising results, unsupervised sparse coding has shown its advantage by optimizing the dictionary adaptively. However, existing unsupervised sparse coding failed to consider the high dimensional manifold information within data. Recently, a graph regularized sparse coding method has shown outstanding performance by incorporating graph laplacian manifold information. In this paper, we proposed a sparse coding method called locally linear embedded sparse coding, to consider the local manifold structure as well as learning the sparse representation. We also provided a novel modified online dictionary learning method which iteratively utilizes modified least angle regression and block coordinate descent method to solve the problem. Instead of getting entire coefficient matrix then update dictionary matrix, our method updates coefficient vector and dictionary matrix in each inner iteration. Extensive experimental results have demonstrated the efficiency and accuracy of our method in image clustering.

CHAPTER 2

GRAPH LAPLACIAN REGULARIZATION WITH SPARSE CODING FOR IMAGE RESTORATION AND REPRESENTATION

2.1 ABSTRACT

Sparse coding is widely used in image denoising, deblurring, clustering, and classification. However, most existing approaches to sparse coding failed to consider the fact that high dimensional data naturally reside on geometrical structure of the data space. It has been shown that geometric information of the data are important for both inversion and discrimination. In this paper, we proposed a generalized framework for image restoration and representation by combining sparse coding and graph based algorithms. In image denoising and deblurring problems, an image is first decomposed into cartoon layer (piecewise-smooth contents) and texture layer (textures and sharp edges) using morphological component analysis (MCA); then optimal graph Laplacian regularizer (OGLR) algorithm and simultaneous sparse coding with gaussian scale mixture prior (SSC-GSM) algorithm are applied to cartoon layer and texture layer respectively; final restored image is by adding the outcomes from two algorithms. Our proposed hybrid image restoration algorithm outperforms state-of-the-art image denoising algorithms, such as BM3D on natural images, by 2 to 5 dB in terms of PSNR (not as good in SSIM index) depending on noise level, performs comparatively in terms of image deblurring. In image clustering and classification problems, our graph Laplacian regularized sparse coding framework demonstrate that high dimension problem can be converted to linear constraint by manifold learning and solved together by

sparse coding. We convert our generalized framework into a novel dual graph regularized sparse coding method to transform the nonlinear data space and feature space into linear space for sparse coding. Two efficient optimization algorithms are provided for numerical implementation. Experimental results show that our generalized graph Laplacian and sparse coding framework performs competitively with popular denoising, deblurring, clustering, and classification (none deep learning) methods.

2.2 INTRODUCTION

Image restoration and representation are two of the major problems in computer vision, numerous studies and research work have been dedicated into this area since decades ago. However, with emerging of machine learning and sparse representation technology, we've witnessed abundance novel algorithms with competitive results.

2.2.1 Image Restoration

Image restoration such as image denoising and deblurring are basic but challenging problems in computer vision.

Reconstruction of a high quality image from its degraded versions (e.g., noisy, blurry) plays an important role in remote sensing, medical imaging, entertainment, and surveillance, etc. With a degraded image $\mathbf{y}$, we can formulate the image restoration (IR) problem as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v}, \tag{2.1}$$

where $\mathbf{H}$ is a degradation matrix, $\mathbf{x}$ is the original image vector and $\mathbf{v}$ is the additive noise vector. With different degradation matrix $\mathbf{H}$, equation (Equation 2.1) can represent different IR problems. When

$\mathbf{H}$ is an identity matrix, it's an image denoising problem; when $\mathbf{H}$ is a blurring operator, it's an image deblurring problem. Extensive studies have been conducted on developing various IR approaches during the past decades [7–15]. Due to the ill-posed nature of IR, appropriate prior knowledge is of great importance for regularization-based techniques to be applied. Well proposed image priors in literature include total variation (TV) [16], sparsity prior [1] and autoregressive prior [17]. Recently, a relatively new prior–*graph signal processing* (GSP)–has been applied in image restoration problems [18] [19], which delivers much improve performance in piece-wise-smooth image (e.g. depth map) denoising. In the mean time, numerous sparse coding image denoising algorithms have been proposed, among those, [20] and [2] demonstrated stably better performance by combining the idea of nonlocal assumption [21] in natural images. Numerous publications demonstrate that sparse coding methods show boosted performance when restore images with numerous edges, curves and texture information, this can be explain by the fact that sparse coding which could be explained by the fact that sparse coding methods such as ICA [22] and KSVD [23] discover edge and texture like dictionaries.

In a seminal work [24], Meyer proposed to decompose images into cartoon (piece-wise-smooth) and texture (texture and sharp edges) parts. Various methods could be used to separate the edges from smooth parts, while others focus on texture modeling. This approach has proposed a vibrant research on the three involved aspects: edges, textures, and decomposition.

The idea of the decomposition can be simply expressed as a decomposition of image $\mathbf{y}$ into cartoon layer $\mathbf{y}_\mathrm{c}$ and texture layer $\mathbf{y}_\mathrm{t}$, as follows:

$$\mathbf{y} = \mathbf{y}_\mathrm{c} + \mathbf{y}_\mathrm{t} \tag{2.2}$$

where $\mathbf{y}_{\mathrm{c}}$ represents the cartoon layer and $\mathbf{y}_{\mathrm{t}}$ for texture layer. In real cases, noise can hidden within both $\mathbf{y}_{\mathrm{c}}$ and $\mathbf{y}_{\mathrm{t}}$.

In [25], a new biologically motivated efficient encoding algorithm has been proposed for representing natural images by projecting image into an edge subspace spanned using ICA [26] basis to capture the texture and sharp edge features, while the residual subspace approximated using a mixture of probabilistic principle component analyzer model. This is a similar alternative to image decomposition method, which demonstrate the advantage of separated processing.

In this paper, we propose a novel image restoration algorithm by first decompose image into cartoon (piecewise-smooth) layer and texture layer. Then restore image patches in each image layer by optimized sparse coding algorithm and graph Laplacian regularizer respectively.

2.2.2 Image Representation

Sparse coding enables efficient representation of signals with only a few active elements. It has shown promising results in ordinary signal processing tasks like image denoising and restoration, audio and video processing, also enhances more complicated tasks like image classification and clustering. When applied to natural images, sparse coding produces learned bases that can resemble the receptive fields of neurons in the visual cortex [27]. Compared with popular methods like PCA and ICA, sparse coding can learn over-complete basis set and doesn't require statistical-independence of the dictionary elements. In machine learning and statistics, a variant of sparse coding – non-negative matrix factorization (NMF) [28] – has been successfully used to obtain interpretable basis elements.

Sparse coding finds intrinsic information within data, however, it does poorly when dealing with high dimensional information such as image clustering and classification. To deal with high dimensional data, numerous dimension reduction methods (manifold methods) have been proposed [29–31]. One problem of manifold methods is lack of theory proof. In [32], Belkin and Niyogi take a step towards closing the gap between theory and practice for a class of Laplacian-based manifolds by showing that under centain conditions, the graph Laplacian of a point cloud converges to the Laplace-Beltrami operator on the underlying manifold.

With graph Laplacian converting high dimensional data space into low dimensional manifold space, sparse coding for efficient representation, graph regularized sparse coding becomes a reasonable thought. Cai [33] proposed a graph regularized nonnegative matrix factorization (GNMF) method. Inspired by his work, Gao [34] and Zheng [35] proposed graph regularized sparse coding (GraphSC), which explicitly considers the local geometrical structure of the data. GNMF and GraphSC show improvements on image clustering and classifcation compared with original NMF [36] and sparse coding [37]. Gu [38] proposed a co-clustering on manifold method to consider the duality between data points (e.g. documents) and features (e.g. words) which can be considered as extension of GNMF [33] and ONMF [39]. In this paper, we propose a dual graph regularized sparse coding method to consider both data and feature space information which is an extension of [40], [41] and [42]. Compared with NMF based methods, our method provides more sparsity and flexibility.

2.2.3 Contribution

Contributions of this paper are mainly three folds. 1) We propose a generalized graph regularized sparse coding framework which can be applied to image restoration (denoising, deblurring) and image representation (clustering, classification) problems; 2) By specializing the framework, we propose a novel image restoration algorithm by first decomposing image into cartoon (piecewise-smooth) layer and texture (texture and sharp edges) layer based on MCA algorithm [43], then restoring the two image layers using optimal graph Laplacian regularizer [19] and optimal sparse coding algorithm [2] respectively. Final restored image is produced by adding the two processed image layers; 3)We propose another special case of the generalized framework for image representation called dual graph regularized saprse coding method with two optimization algorithms to consider both data and feature space to enhance applications of image clustering and classification.

Our paper is organized as follows. We first present present notations and preliminaries of graph Laplacian and sparse coding in Section II. Then, in Section III, we propose a generalized framework of graph regularized sparse coding method for image restoration and representation, then show that the objective functions of our novel image restoration and representation methods are simply special cases of the generalized framework. Then, in Section IV, we present the novel graph regularizer with sparse coding algorithm for image restoration. After that, in Section V, we illustrate the novel dual graph regularized sparse coding algorithm for image clustering and classification. Then, in Section VI, extensive experiments are presented to demonstrate the competitive performance of our algorithms in image restoration and representation. Finally, we conclude this paper in Section VII.

2.3 Backgrounds and Related Work

We first review the concepts and related work in image denoising and deblurring with sparse coding methods and graph Laplacian related methods. Then we go through the related work in image clustering and classification using sparse coding and graph regularized methods.

2.3.1 Image Restoration

2.3.1.1 Image Decomposition

Decomposing a signal or image into superposed contributions from different sources is one of the fundamental problem in mathematics, signal and image processing, where morphologically decompose signals into their building blocks is one of the successful attempts. In image processing, one of the interesting and complex source separation problem is to decompose an image into piecewise-smooth (cartoon) and texture parts.

In [43], based on sparse representation of signals, a novel decomposition method called morphological component analysis (MCA) was proposed. The basic assumption behind MCA is that a sigal is made of a linear mixture of different layers, which is called morphological components (morphologically distinct from each other). To achieve this assumption, the behavior of each component should be able to separate and there must exist a dictionary of elements can be used to construct a sparse representation. Then, each morphological component is further assumed to be sparsely represented in specific transform domain. When transformation is made to a corresponding morphological component, it should lead to a sparse representation over the component and highly inefficient in representing other components. We can use a pursuit aglorithm to get the sparset representation which leads to optimal separation if those dictionaries are identified. We use MCA as our driving force to decompose image into cartoon layer and texture layer due to its robust performance across images and settings.

2.3.1.2 Sparse Coding Related Methods

Redundant representations and sparsity have been driving forces for restoration of signals in the past descade or so. Shrinage algorithm is the sparsity candidate by considering the sparsity of wavelet coefficients. One reason to use redundant representations is having the shift invariant property. Also, as regular 1-D wavelets are not optimal for processing images, several novel directional redundant and multiscale transforms were proposed, including the contourlet [44], curvelet [45], bandlet [46], wedgelet [47], and the steerable wavelet. At the same time, the introduction of greedy and convex optimization algorithms such as matching pursuit [48] and the basis pursuit [49] in denoising application directed image denoising problem as a direct sparse decomposition technique over redundant dictionaries. All these lead to some of the best image denoising methods available today.

In [2], the authors proposed a nonlocal extension of Gaussian scale mixture (GSM) [50] model using simultaneous sparse coding (SSC) to set regularization parameters in a principled yet spatially adaptive fashion, which outperform all existing sparse coding related image denoising algorithms. It has been shown that sparse representation with proper prior can better preserve sharp edges and textures. In this paper, we adopt GSM model from the work of [2] as the optimal sparse coding to model the texture layer of image.

2.3.1.3 Graph Laplacian Related Methods

Recently, researchers have pay much attention to the linkage between graphs and manifold counterparts. In [51], [32], [52] and [53], the authors showed that for both uniform measure and non-uniform measure on submanifold, the graph Laplacian operator will converge to the continuous Laplace-Beltrami operator. In [52], Hein further showed the convergence of the graph Laplacian regularizer to a functional for Hölder functions on Riemannian manifolds. With this result in mind, we can conduct analysis in both discrete domain and continuous domain depending on problems to be solved.

Based on heat diffusion equation on a graph, a new method was proposed for smoothing both gray-scale and color images in [54]. Image smoothing is accomplished by convolving the heat kernel with the image, and its numerical implementation is realized by using the Krylov subspace technique. In [18], a nonlocal based graph Laplacian regularizer method was proposed for denoising piecewise smooth images (e.g. depth map), which gives state-of-the-art denoising result for depth map images. In [19], the authors take a step further to analyze graph Laplacian regularizer on the image denoising problem by constructing neighborhood graphs of pixel patches as discrete counterparts of Riemannian manifolds and perform analysis in the continuous domain. It is shown that graph Laplacian regularizer algorithm outperforms all other algorithms in terms of piecewise smooth image denoising. We adopt the methods in [19] as the optimal graph Laplacian regularizer to restore cartoon layer of an image.

2.3.2 Image Representation

In 2001, Shi and Malik [55] proposed a novel approach called normalized cut (Ncut) for solving the perceptual grouping problem in vision by using the Laplacian eigenvalues, the algorithm can be used both in image segmentation and clustering. Since then, numerous research results related to graph Laplacian started to coming out. In [56], a unifying view of segmentation using eigenvectors was proposed which can be easily applied in image clustering. Later, Belkin and Niyogi [57] proposed a Laplacian eigenmaps method for dimensionality reduction, and data representation draws on the correspondence between the graph Laplacian, the Laplace Beltrami operator on the manifold, and the connections to the heat equation. Laplacian eigenmaps provide a computationally efficient approach to nonlinear dimensionality reduction that has locality-preserving properties and a natural connection to clustering. Recently, a Laplacian regularized low-rank representation [58] has been proposed to take into account the non-linear geometric structures within data which extends the application of low rank representation, as low rank representation has similarity with sparse coding in terms of dimension reduction, a reasonable thought is graph Laplacian regularized sparse coding method which has been proposed in [35].

In this paper, our proposed dual graph regularized sparse coding model also considers both data space (column space) and feature space (row space) in a given data matrix. However, instead of using low-rank representation, our model is a more generalized sparse coding method by iteratively updating learned dictionary and coefficients matrix with sparsity constrain. We show that our model is not only very functional in terms of delivering state-of-the-art clustering performance but also can enhance classification performance.

2.4 Notation and Preliminaries

2.4.1 Graph Laplacian

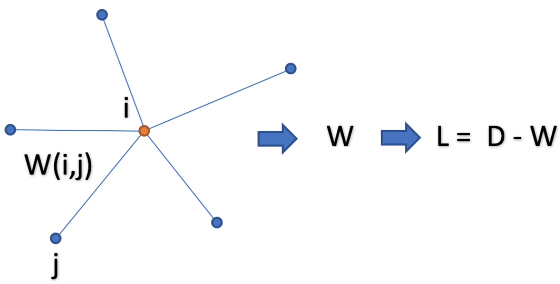


Figure 1. Illustration of graph Laplacian

A weighted graph $G = (V, E)$ is a set of vertices $\{v_1, ..., v_n\} \in V$ and weighted edges connecting these vertices represented by an adjacency matrix W. W is a symmetric matrix with nonnegative entries. Recall that the Laplacian matrix of a weighted graph G is the matrix $L = D - W$ as shown in Figure 1, where D is a diagonal matrix $D(i,i) = \sum_i W(i,j)$.

With a set of point $\{\mathbf{x}_1, ..., \mathbf{x}_n$ in $\mathbb{R}^k\}$, we can construct a graph G, whose vertices are data points. If we use heat kernel $W_n^t(i,j) = \exp\{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{4t}\}$ to calculate weight matrix, the corresponding graph Laplacian can be represented as $L_n^t = D_n^t - W_n^t$. The relationship between heat equation and Laplacian operator is in Appendix A.

Given a 2D image with *t* and *n* fixed, the unnormalized graph Laplacian–the most basic type of graph Laplacian–is defined as

$$\mathbf{L} = \mathbf{D} - \mathbf{W} \tag{2.3}$$

where $\mathbf{L}$ has 0 as its smallest eigenvalue and a constant vector as the corresponding eigenvector; it is symmetric and positive semi-definite [59].

2.4.2 Sparse coding

Given a data matrix $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_m] \in \mathbb{R}^{n \times m}$, we can represent it as a product of dictionary matrix $\mathbf{D}$ and coefficient matrix $\mathbf{A}$, where $\mathbf{D} = [\mathbf{d}_1, ..., \mathbf{d}_k] \in \mathbb{R}^{n \times k}$, each $\mathbf{d}_i$ represents a basis column in the dictionary, $\mathbf{A} = [\boldsymbol{\alpha}_1, ..., \boldsymbol{\alpha}_m] \in \mathbb{R}^{k \times m}$, each column is a sparse representation for a data point.

Following [60] [37], we use ℓ_1 norm instead of ℓ_0 norm to produce similar results with affordable computational cost. The objective function then becomes

$$\{\hat{\mathbf{D}}, \hat{\mathbf{A}}\} = \underset{\mathbf{D},\mathbf{A}}{\arg\min} \|\mathbf{X} - \mathbf{DA}\|_F^2 + \beta \sum_{i=1}^{m} \|\boldsymbol{\alpha}_i\|_1 \quad \text{s.t.} \quad \|\mathbf{d}_i\|^2 \leqslant c, i = 1,...,k \tag{2.4}$$

where β is a regularization parameter for sparsity constraint. Although the objective function is not convex with **D** and **A** together, it is convex with either one fixed. We iteratively optimize the objective function by minimizing over one variable with the other one fixed [61]. Thus, it becomes a ℓ_1-regularized least squares problem with a ℓ_2-constrained least square problem.

2.5 GENERALIZED FORMULA

Graph Laplacian regularizer, sparse coding and graph regularized sparse coding are powerful methods in image denoising, deblurring, clustering and classification.

In this section, we propose a generalized framework for graph regularized sparse representation and illustrate novel algorithms in image restoration and representation as special cases of this general framework.

Recall the definition of graph Laplacian in Eq. (Equation 2.3) and sparse coding in Eq. (Equation 2.4). By combining sparse coding with graph Laplacian regularizer, we get the following generalized framework in Eq. (Equation 2.5).

$$\begin{gathered}\{\hat{\mathbf{D}}, \hat{\mathbf{A}}\} = \underset{\mathbf{D},\mathbf{A}}{\arg\min} \|\mathbf{X} - \mathbf{DA}\|_F^2 + \gamma \cdot \mathrm{Tr}(\mathbf{ULU}^T) + \beta \sum_{i=1}^{m} \|\boldsymbol{\alpha}_i\|_1 \\ \text{s.t.} \quad \|\mathbf{d}_i\|^2 \leqslant c, i = 1,...,k\end{gathered} \tag{2.5}$$

where $\mathbf{U}$ is the generalized term for graph Laplacian regularization, $\mathbf{X}$, $\mathbf{D}$, $\mathbf{L}$, $\mathbf{d}$ and $\boldsymbol{\alpha}$ are defined as in Eq. (Equation 2.3) and (Equation 2.4), γ is the constraint parameter of graph Laplacian regularizer, β is sparsity constraint parameter.

2.5.1 Image Restoration Formula

Image restoration is performed by using two special case of Eq. (Equation 2.5). As our image restoration method is patch based (8×8 in general), for simplicity, we represent the formula in vector form.

2.5.1.1 Sparse Coding

In case of sparse coding, we would like to model the fast intensity changes, γ is forced to be 0. We get the following image patch restoration equation.

$$\hat{\alpha} = \arg\min_{\boldsymbol{\alpha}} \|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|^2 + \beta\|\boldsymbol{\alpha}\|_1 \quad \text{s.t. } \|\mathbf{d}_i\|^2 \leqslant c, i = 1,...,k \tag{2.6}$$

where $\mathbf{x} \in \mathbb{R}^n$ (n is size of image patches, $n = 8$ in this paper) is an image patch vector, $\mathbf{D} \in \mathbb{R}^{n \times K}(n \leq K)$ is the dictionary, $\boldsymbol{\alpha}$ is coefficient vector under some sparsity constraint, β is the regularization parameter for sparsity.

2.5.1.2 Graph Laplacian Regularizer

In case of graph Laplacian regularizer, we are modeling the piecewise-smooth intensity variation within images, therefore sparse constraint is not needed, thus β equals to 0. The optimal graph Laplacian $\hat{\mathbf{L}}$ is calculated from [19]. We could assume $\mathbf{D}$ is an identity matrix. To denoise an observed pixel patch

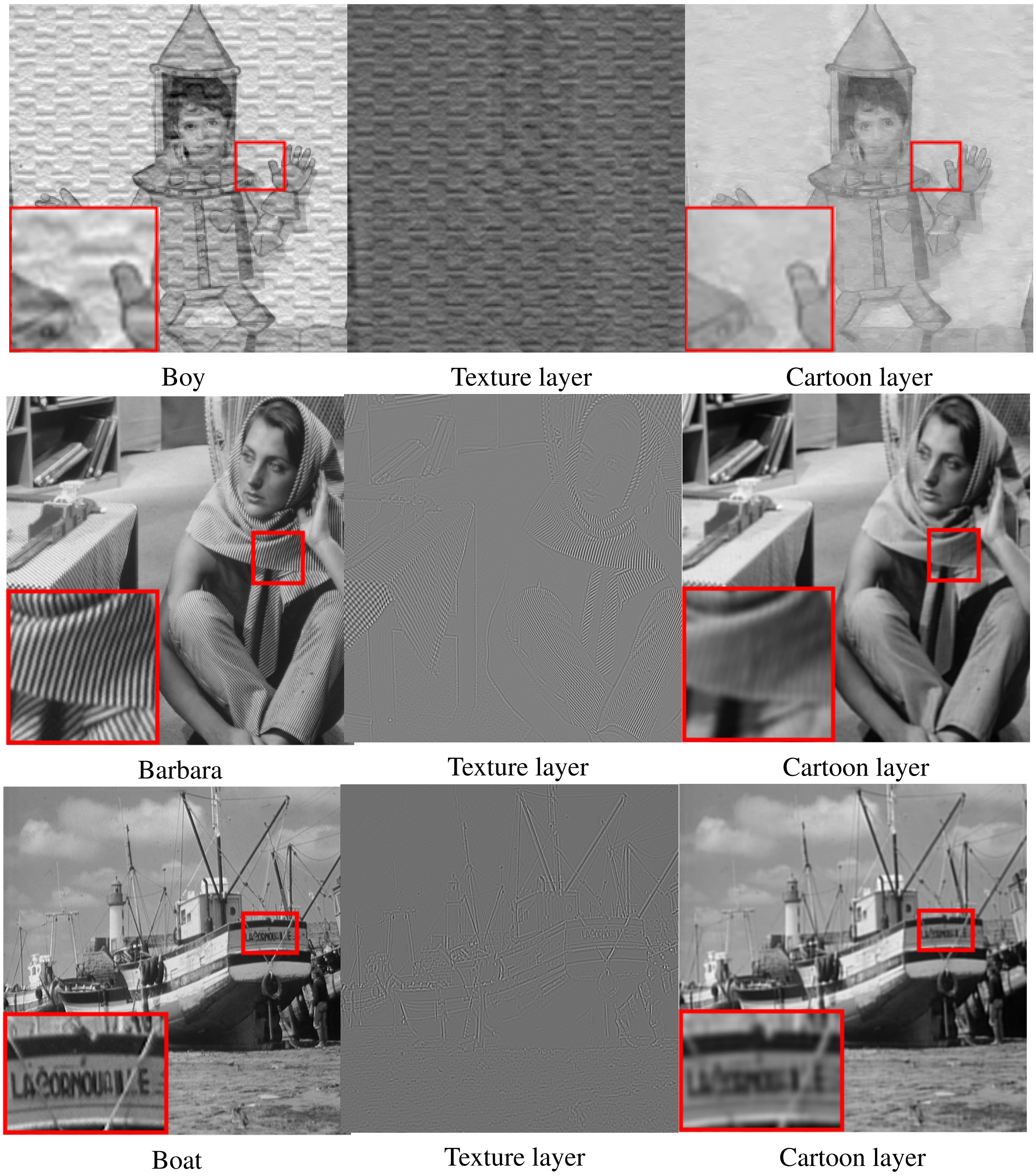


Figure 2. MCA decomposition of images into texture layer (texture and sharp edges) and cartoon layer (piece-wise-smooth).

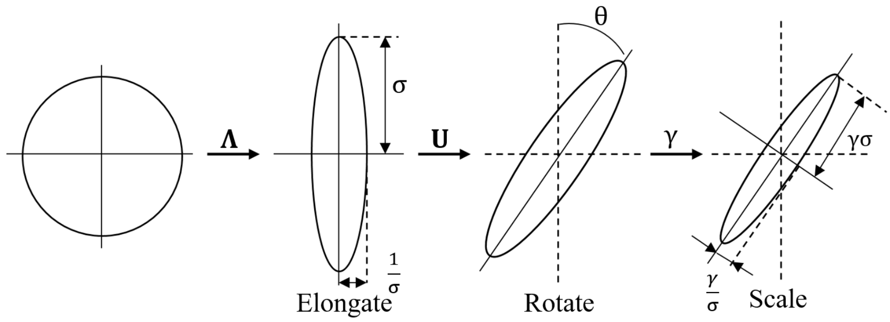


Figure 3. Schematic representation illustrating the effects of the steering matrix and it components ($\mathbf{G} = \gamma \mathbf{U} \boldsymbol{\Lambda} \mathbf{U}^{\mathsf{T}}$) on the size and shape of the graph Laplacian

$\mathbf{x}$, we combine the graph Laplacian regularizer prior $\mathbf{u}^{\mathsf{T}} \hat{\mathbf{L}} \mathbf{u}$ with an ℓ_2 fidelity term computing the noisy observation $\mathbf{x}$ and denoised patch $\mathbf{u}$ as follows

$$\hat{\mathbf{u}} = \arg\min_{\mathbf{u}} \|\mathbf{x} - \mathbf{u}\|^2 + \gamma(\mathbf{u}^{\mathsf{T}} \hat{\mathbf{L}} \mathbf{u}) \tag{2.7}$$

where γ is the regularization parameter to constrain graph Laplacian prior.

2.5.2 Image Representation Formula

In image representation problem, we model the objective function in matrix format as follows

$$\begin{gathered}
\{\hat{\mathbf{D}}\,\hat{\mathbf{A}}\} = \arg\min_{\mathbf{D},\mathbf{A}} \|\mathbf{X} - \mathbf{D}\mathbf{A}\|_F^2 + \gamma \cdot \mathrm{Tr}(\mathbf{U}\mathbf{L}\mathbf{U}^{\mathsf{T}}) + \beta \sum_{i=1}^{m} \|\boldsymbol{\alpha}_i\|_1 \\
\mathbf{U} = (\sqrt{\frac{\lambda}{\gamma}}\mathbf{A} \quad \sqrt{\frac{\mu}{\gamma}}\mathbf{D}^{\mathsf{T}}), \quad \mathbf{L} = \begin{pmatrix} \mathbf{L}^A & \mathbf{0} \\ \mathbf{0} & \mathbf{L}^D. \end{pmatrix}
\end{gathered} \tag{2.8}$$

We have proposed dual graph regularized sparse coding methods for image representation here, λ and μ are constraints of graph Laplacian regularizers. Definition of $\mathbf{X}$, $\mathbf{D}$, $\mathbf{A}$, $\mathbf{L}^{A}$ and $\mathbf{L}^{D}$ are in Eq. (Equation 2.21), (Equation 2.22), (Equation 2.23), and (Equation 2.24).

2.6 GLSC FOR IMAGE RESTORATION

Here GLSC represent Graph Laplacian regularization with sparse coding methods.

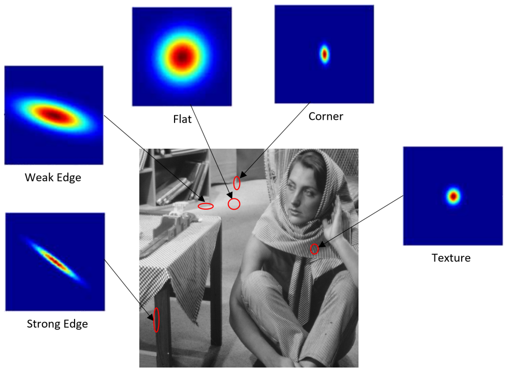
2.6.1 Image Decomposition

Decomposition of an image into cartoon layer (piecewise-smooth) and texture layer (texture and sharp edges) seems to be analogous to the high-pass and low-pass decomposition in image filtering. However, this doesn't work as what we want: both cartoon layer and texture layer can contain high frequencies, so a simple filtering could not separate those two layers.

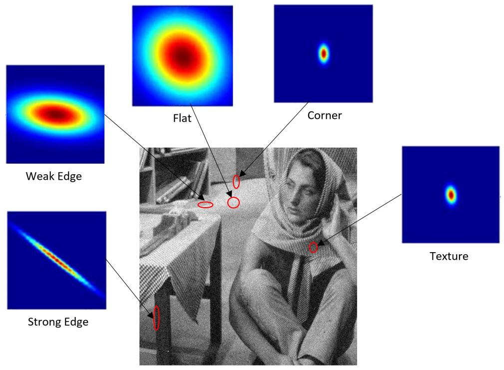
MCA method combines the basis pursuit denoising (BPDN) algorithm and the total-variation (TV) regularized scheme. The basic idea of MCA is the use of two appropriate dictionaries, one for the representation of textures and the other for the natural scene parts assumed to be piecewise smooth. Both dictionaries are chosen such that they lead to sparse representations over one type of image-content (either texture or piecewise smooth). As the need to choose proper dictionaries is generally hard, a TV regularization is employed to better direct the separation process and reduce ringing artifacts.

MCA algorithm can be formulated as follows

$$\{\hat{\alpha}_t, \hat{\alpha}_c\} = \arg\min_{\alpha_t, \alpha_c} \|\boldsymbol{\alpha}_t\|_1 + \|\boldsymbol{\alpha}_c\|_1 + \tilde{\tau}\|\mathbf{x} - \mathbf{D}_t\boldsymbol{\alpha}_t - \mathbf{D}_c\boldsymbol{\alpha}_c\|_2^2 + \tilde{\psi}\mathrm{TV}(\mathbf{D}_c\boldsymbol{\alpha}_c) \tag{2.9}$$



Original



Noisy, 22.13dB

Figure 4. Effects of steering matrix $\mathbf{G}$ for noiseless case and noisy case with AWGN $\sigma = 20$

where $\{\mathbf{D}_\mathrm{c}, \boldsymbol{\alpha}_\mathrm{c}\}$ and $\{\mathbf{D}_\mathrm{t}, \boldsymbol{\alpha}_\mathrm{t}\}$ are dictionary matrix and coefficient vector for cartoon layer and texture layer respectively. TV is a total variation penalty in [16] that can direct image $\mathbf{D}_\mathrm{c}\boldsymbol{\alpha}_\mathrm{c}$ to fit the piece-wise-smooth model.

Here, we separate dictionary columns into two dictionaries by thresholding the total variation of dictionary atoms. Figure 2 displays the results of decomposing each of *Boy*, *Barbara*, and *Boat* into cartoon and texture parts.

We assume that random gaussian noise is evenly distribution within both cartoon layer and texture layer. Thus the variance of noise in each layer will be half of the image noise before decomposition.

2.6.2 Optimal Cartoon Restoration Algorithm

Graph Laplacian regularizer [19] outperforms state-of-the-art image denoising algorithms such as BM3D significantly in terms of depth image denoising. As depth images are mostly piecewise smooth, it's a natural thought that graph based method could also perform particularly well in cartoon (piece-wise-smooth) image denoising.

With this observation, various graph-based smoothness priors [18] has been proposed to restore piece-wise-smooth images, [19] take a step further by providing theoretically justified explanation with derived optimal graph Laplacian regularizer for image restoration.

The core idea of [19] is to first prove that graph Laplacian regularizer converges to its continuous counter part. Then obtains the optimal solution of discrete Laplacian regularizer by exploring the optimal solution in continuous domain.

The advantage of doing analysis in continuous domain is that it's possible to utilize existing kernel based method. Similar to steering kernel [62], OGLR utilizes gradient covariance to model local edge structure.

The local edge structure is related to the gradient covariance (or equivalently, the locally dominant orientation), where a naive estimate of this covariance matrix may be obtained as follows [62]:

$$\mathbf{G} = \begin{bmatrix} \sum_{\mathbf{x}_i \in \omega} g_{x_1}(\mathbf{x}_i) g_{x_1}(\mathbf{x}_i) & \sum_{\mathbf{x}_i \in \omega} g_{x_1}(\mathbf{x}_i) g_{x_2}(\mathbf{x}_i) \\ \sum_{\mathbf{x}_i \in \omega} g_{x_1}(\mathbf{x}_i) g_{x_2}(\mathbf{x}_i) & \sum_{\mathbf{x}_i \in \omega} g_{x_2}(\mathbf{x}_i) g_{x_2}(\mathbf{x}_i), \end{bmatrix} \tag{2.10}$$

where $\mathbf{G}$ is a covariance matrix and often called steering matrix, $g_{x_1}(\dots)$ and $g_{x_2}(\dots)$ are the first order derivatives along x_1 and x_2 directions and ω is the local analysis window around point of interest. As $\mathbf{G}$ is a symmetric covariance matrix, eigenvalue decomposition is proposed to model is more efficiently as follows

$$\mathbf{G} = \gamma \mathbf{U} \boldsymbol{\Lambda} \mathbf{U}^{\mathsf{T}} \tag{2.11}$$

$$\mathbf{U} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \quad \boldsymbol{\Lambda} = \begin{bmatrix} \sigma & 0 \\ 0 & \sigma^{-1} \end{bmatrix} \tag{2.12}$$

where γ is a scaling factor, $\mathbf{U}$ is rotation matrix, $\boldsymbol{\Lambda}$ is the elongation matrix. Figure 3 explains how these parameters affect the spreading of covariance matrix $\mathbf{G}$ along x_1 and x_2 direction derivative.

Figure 4 is a visual illustration of steering matrix on a variety of image structures (texture, flat, strong edge, corner, and weak edge) of the Barbara image for both noiseless and noisy cases. We can find that the shape and orientation of the estimated metrics (rotation, elongation and scaling) are very close in both noisy and noiseless cases. Also, depending on the underlying features, in the flat areas, they are relatively more spread to reduce the noise effects, while in texture areas, their spread is very close to the noiseless case which reduces blurriness.

As graph Laplacian regularizer $\mathbf{u}^{\mathsf{T}}\mathbf{L}\mathbf{u}$ converges to a continuous metric space that can utilize the power of steering matrix, we could derive the optimal discrete graph Laplacian regularizer by doing analysis in continuous domain, detail and proofs in [19].

2.6.3 Optimal Texture Restoration Algorithm

In this section, we explain why we choose simultaneous sparse coding with Gussian scale mixture prior (SSC-GSM) [2] as the optimal texture restoration algorithm.

Nonlocal image restoration techniques have attracted increasing attention since it's publication. The key motivation behind lies in the observation that in natural images, many image structures such as edges and textures are abundance of self-repeating patterns. Such observation has led to the formulation of simultaneous sparse coding (SSC) [20]. However, how to achieve (local) spatial adaption within the framework of nonlocal image restoration still remains an open question.

SSC-GSM algorithm successfully addresses the question by connecting Gaussian scale mixture (GSM) with SSC. The idea is to model each sparse coefficient as a Gaussian distribution with a positive scaling variable and impose a sparse distribution prior (Jeffreys prior [63] in model) over the positive scaling variables.

Experimental results also shown that, compared with other sparse coding algorithms, SSC-GSM based image restoration can better preserve edge sharpness and suppress undesirable artifacts in the restored images. As texture layer (such as in Figure 2) are mostly composed of shape edges and textures, we choose SSC-GSM as the optimal texture restoration algorithm.

Figure 5 shows the comparison between SSC-GSM and KSVD in sharp edge preservation. SSC-GSM denoised image shows sharper edges and textures which is visually satisfactory.

Derivation of SSC-GSM is based on MAP estimator. For a given observation $\mathbf{x} = \mathbf{D}\boldsymbol{\alpha} + \mathbf{n}$, where $\mathbf{x}$ is an image patch vector, $\mathbf{D}$ is dictionary matrix, $\boldsymbol{\alpha}$ is the sparse coefficient vector, $\mathbf{n} \sim N(0, \sigma_n^2)$ denote the additive Gaussian noise, we can formulate the following MAP estimator

$$\begin{aligned} \boldsymbol{\alpha} &= \arg\max_{\boldsymbol{\alpha}} \log P(\mathbf{x}|\boldsymbol{\alpha})P(\boldsymbol{\alpha}) \\ &= \arg\max_{\boldsymbol{\alpha}} \{\log P(\mathbf{x}|\boldsymbol{\alpha}) + \log P(\boldsymbol{\alpha})\} \end{aligned} \tag{2.13}$$

As the GSM [50] model of coefficients $\boldsymbol{\alpha}$ decomposes coefficient vector $\boldsymbol{\alpha}$ into the point-wise product of a Gaussian vector $\boldsymbol{\beta}$ and a hidden multiplier $\boldsymbol{\theta}$, i.e. $\alpha_i = \theta_i \beta_i$, where θ_i is the positive scaling

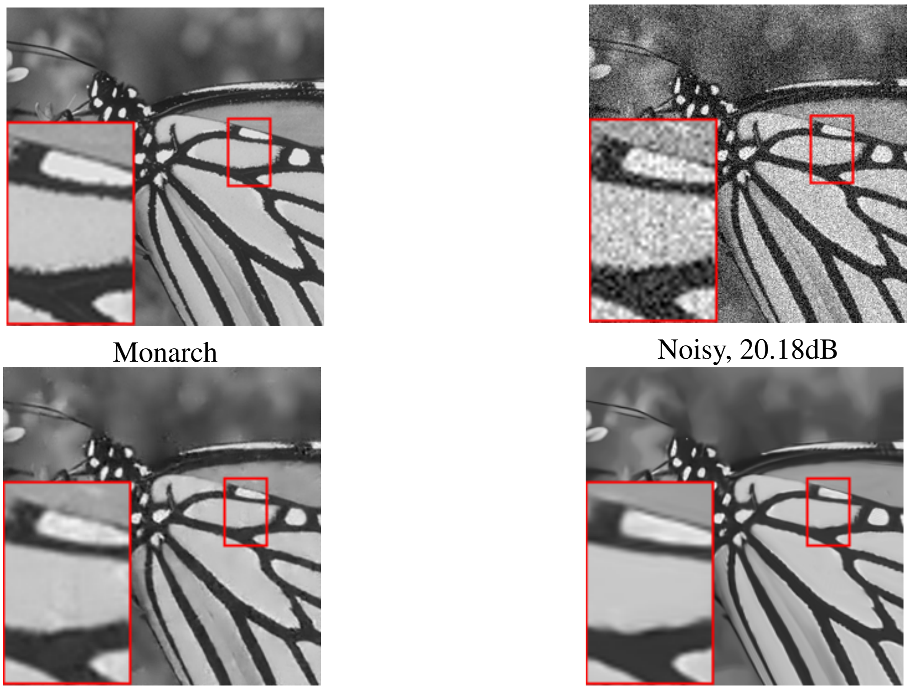


Figure 5. Denoising performance comparison between KSVD [1] and SSC-GSM [2].

variable with probability $P(\theta_i)$. Given θ_i, a coefficient α_i is Gaussian with the standard deviation of θ_i. With the assumption that θ_i are i.i.d and independent of β_i, the GSM prior of $\boldsymbol{\alpha}$ can be expressed as

$$P(\boldsymbol{\alpha}) = \prod_i P(\alpha_i), \quad P(\alpha_i) = \int_0^\infty P(\alpha_i|\theta_i)P(\theta_i)d\theta_i \tag{2.14}$$

Then the MAP estimator of $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$ becomes

$$(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\theta}}) = \arg\max_{\boldsymbol{\alpha},\boldsymbol{\theta}} \log P(\mathbf{x}|\boldsymbol{\alpha}, \boldsymbol{\theta}) P(\boldsymbol{\alpha}, \boldsymbol{\theta})$$
$$= \arg\max_{\boldsymbol{\alpha},\boldsymbol{\theta}} \log P(\mathbf{x}|\boldsymbol{\alpha}) + \log P(\boldsymbol{\alpha}|\boldsymbol{\theta}) + \log P(\boldsymbol{\theta}) \tag{2.15}$$

In Eq. (Equation 2.15), $P(\mathbf{x}|\boldsymbol{\alpha})$ is a likelihood term characterized by Gaussian function with variance σ_n^2. As coefficient α_i is Gaussian with standard derivation of θ_i. Given θ_i, the prior term $P(\boldsymbol{\alpha}|\boldsymbol{\theta})$ can be expressed as

$$P(\boldsymbol{\alpha}|\boldsymbol{\theta}) = \prod_i P(\alpha_i|\theta_i) = \prod_i \frac{1}{\theta_i\sqrt{2\pi}} \exp\Big(-\frac{(\alpha_i - \mu_i)^2}{2\theta_i^2}\Big). \tag{2.16}$$

Instead of assuming the mean $\mu_i = 0$, SSC-GSM used a biased mean μ_i for α_i where μ_i is calculated based on nonlocal means approach [21] given m similar image patches.

By using a noninformative prior $P(\theta_i) \approx \frac{1}{\theta_i}$ – Jeffrey's prior, Eq. (Equation 2.15) becomes

$$(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\theta}}) = \arg\min_{\boldsymbol{\alpha},\boldsymbol{\theta}} \frac{1}{2\sigma_n^2} \|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \sum_i \log(\theta_i^2\sqrt{2\pi})$$
$$+ \sum_i \frac{(\alpha_i - \mu_i)^2}{2\theta_i} \tag{2.17}$$

Where $\log P(\boldsymbol{\theta}) = \sum_i \log \theta_i$. Optimization and all other details are in [2].

2.6.4 Algorithm Summary

Our proposed image restoration algorithm is a hybrid algorithm by first using MCA algorithm to decompose image into cartoon layer and texture layer with local DCT and curvelet with fixed threshold, then use optimal graph Laplacian regularizer [19] to restore cartoon layer, use optimal sparse coding algorithm [2] to restore texture layer. Final restored image is by adding the processed outcomes of two optimal algorithms.

As our algorithm is a reasonable combination of MCA, graph Laplacian regularizer and sparse coding algorithm, we call it MCA-GSC algorithm and summaried in **Algorithm** 1.

Algorithm 1 Image Restoration with MCA-GSC

1: **Input**: Noisy image $\mathcal{I}$, noisy variance $\sigma_{\mathcal{I}}^2$, blur matrix $\mathcal{H}$.
2: Using MCA algorithm [43] to decompose degraded image into cartoon layer and texture layer with selected dictionaries $\mathbf{D}_c$ and $\mathbf{D}_t$ (KSVD [23] learned dictionary with DCT initialization, calculate the normalized total gradient variation and thresholded by 0.27).
3: Restore cartoon layer with optimal graph Laplacian regularizer (OGLR) [19], where cartoon layer noise variance estimated as $\sigma_c^2 = \frac{1}{2}\sigma_{\mathcal{I}}^2$ with 11 iterations. Reconstruct whole cartoon layer image by Eq. (Equation 2.20).
4: Aggregate those two restored image layers.
5: **Output**: Restored image

The standard image degradation can be modeled as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v}, \tag{2.18}$$

where $\mathbf{x} \in \mathbb{R}^N$ and $\mathbf{y} \in \mathbb{R}^M$ are the original and degraded images, $\mathbf{H} \in \mathbb{R}^{M \times N}$ is a degradation matrix, and $\mathbf{v}$ is the additive white Gaussian noise with $\mathbf{v} \sim N(0, \sigma^2)$. The whole image reconstruction can be formulated as

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \phi \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \sum_i \|\tilde{\mathbf{x}}_i - \mathbf{R}_i\mathbf{x}\|_2^2 \tag{2.19}$$

where in graph Laplacian regularizer method (OGLR), $\tilde{\mathbf{x}}_i = \hat{\mathbf{u}}$ as in Eq. (Equation 2.7), in sparse coding method (SSC-GSM), $\tilde{\mathbf{x}}_i = \mathbf{D}\hat{\boldsymbol{\alpha}}_i$ as in Eq. (Equation 2.6), $\mathbf{R}_i \in \mathbb{R}^{n\times N}$ denotes a matrix extracting the i-th patch $\mathbf{x}_i$ from $\mathbf{x}$, ϕ is regularization parameter. This is a quadratic form and has a closed-form solution of the form

$$\hat{\mathbf{x}} = \left(\phi\mathbf{H}^T\mathbf{H} + \sum_i \mathbf{R}_i^T\mathbf{R}_i\right)^{-1}\left(\phi\mathbf{H}^T\mathbf{y} + \sum_i \mathbf{R}_i^T\tilde{\mathbf{x}}_i\right) \tag{2.20}$$

In the situation of image denoising, $\mathbf{H}$ is an identity matrix, while in case of image restoration, $\mathbf{H}$ is a blur kernel.

2.7 Dual Graph Regularized Sparse Coding for Image Representation

In this section, we introduce a novel dual graph regularized sparse coding (DGSC) method for image representation, as a special case of our generalized framework in Eq. (Equation 2.5). We first introduce data graph and feature graph as the key concepts to build our method, then we provide two optimization algorithm to efficiently solve the objective function.

2.7.1 Data Graph and Feature Graph

Give data matrix $\mathbf{X} \in \mathcal{R}^{n\times m}$, we first construct a data graph $\mathcal{G}_A$ whose vertices $\{\mathbf{x}_{\cdot 1}, ..., \mathbf{x}_{\cdot m}\}$ are columns of $\mathbf{X}$, then we build a feature graph $\mathcal{G}_D$ whose vertices $\{\mathbf{x}_{1\cdot}, ..., \mathbf{x}_{n\cdot}\}$ are rows of $\mathbf{X}$. The binary nearest neighbor weight matrices for data and feature are defined as

$$W_{ij}^A = \begin{cases} 1, \text{ if } \mathbf{x}_{\cdot i} \in \mathcal{N}(\mathbf{x}_{\cdot j}) \text{ or } \mathbf{x}_{\cdot j} \in \mathcal{N}(\mathbf{x}_{\cdot i}) \\ 0, \text{ otherwise.} \end{cases} \tag{2.21}$$

$$W_{ij}^{D} = \begin{cases} 1, \text{ if } \mathbf{x}_{i\cdot} \in \mathcal{N}(\mathbf{x}_{j\cdot}) \text{ or } \mathbf{x}_{j\cdot} \in \mathcal{N}(\mathbf{x}_{i\cdot}) \\ 0, \text{ otherwise.} \end{cases} \tag{2.22}$$

where $\mathcal{N}(\mathbf{x})$ represents nearest neighbor of vector $\mathbf{x}$.

We define $\mathbf{e}_i^A = \sum_{j=1}^{m} W_{ij}^A$, $\mathbf{e}_i^D = \sum_{j=1}^{n} W_{ij}^D$ and $\mathbf{E}^A = diag(\mathbf{e}_1^A, ..., \mathbf{e}_m^A)$, $\mathbf{E}^D = diag(\mathbf{e}_1^D, ..., \mathbf{e}_n^D)$. According to [31], the reasonable mappings from data graph $\mathcal{G}_A$ to coefficient matrix $\mathbf{A}$ and from feature graph $\mathcal{G}_D$ to dictionary matrix $\mathbf{D}$ are

$$\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}(\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_j)^2 W_{ij}^A = \mathrm{Tr}(\mathbf{A}\mathbf{L}^A\mathbf{A}^T) \tag{2.23}$$

$$\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}(\mathbf{d}_{i\cdot} - \mathbf{d}_{j\cdot})^2 W_{ij}^D = \mathrm{Tr}(\mathbf{D}^T\mathbf{L}^D\mathbf{D}) \tag{2.24}$$

where $\mathbf{L}^A = \mathbf{E}^A - \mathbf{W}^A$, $\mathbf{L}^D = \mathbf{E}^D - \mathbf{W}^D$.

By adding data graph (Equation 2.23) and feature graph (Equation 2.24) to the original sparse coding function (Equation 2.4), we get objective function

$$\mathcal{O} = \min_{D,A}\{\|\mathbf{X} - \mathbf{D}\mathbf{A}\|_F^2 + \lambda\mathrm{Tr}(\mathbf{A}\mathbf{L}^A\mathbf{A}^T) + \mu\mathrm{Tr}(\mathbf{D}^T\mathbf{L}^D\mathbf{D}) + \beta\sum_{i=1}^{m}\|\boldsymbol{\alpha}_i\|_1\} \tag{2.25}$$

where $\mathbf{D}$ is the dictionary matrix, $\mathbf{A}$ is the coefficient matrix with sparsity constraint, λ and β are regularizer constraints for graph Laplacian and sparsity respectively.

Fixing dictionary matrix $\mathbf{D}$, regularizer $\mathrm{Tr}(\mathbf{D}^T\mathbf{L}^D\mathbf{D})$ becomes a constant value, so we can eliminate it in this step of minimization.

As reconstruction error $||\mathbf{X}-\mathbf{D}\mathbf{A}||^2$ can be written as $\sum_{i=1}^{m}||\mathbf{x}_i-\mathbf{D}\boldsymbol{\alpha}_i||^2$ and graph data regularizer $\mathrm{Tr}(\mathbf{A}\mathbf{L}^A\mathbf{A}^T)$ is same as $\sum_{i,j=1}^{m}L^A{}_{ij}\boldsymbol{\alpha}_i^T\boldsymbol{\alpha}_j$.

$$\min_{\alpha_i}\sum_{i=1}^{m}||\mathbf{x}_i-\mathbf{D}\boldsymbol{\alpha}_i||^2+\lambda L^A{}_{ii}\boldsymbol{\alpha}_i^T\boldsymbol{\alpha}_i+\boldsymbol{\alpha}_i^T\mathbf{h}_i+\beta\sum_{i=1}^{m}||\boldsymbol{\alpha}_i||_1 \tag{2.26}$$

where $\mathbf{h}_i=2\lambda(\sum_{j\neq i}\boldsymbol{\alpha}_j)$ is a summation of $i\neq j$ terms.

2.7.2 Optimization Algorithms

In this section, we present two optimization algorithms: 1) Feature-sign search [64] with Lagrange dual for learning bases (FS-LD); 2) Least angle regression [65] with Block coordinate descent for dictionary update (LARS-BCD).

2.7.2.1 Feature-Sign Search with Lagrange Dual

In [35], the author utilized feature-sign search algorithm for solving graph regularized sparse coding problem. Since we get the same objective function when dictionary **D** is fixed, the implementation process is the same for this step.

Bases are updated by using Lagrange dual, which is an algorithm for solving optimization problem over dictionary bases **D** with fixed coefficients **A**. We get the following objective function

$$\min_{\mathbf{D}}\{||\mathbf{X}-\mathbf{D}\mathbf{A}||_F^2+\mu||\mathbf{D}^T\mathbf{L}^D\mathbf{D}||\}\quad \text{s.t. } ||\mathbf{d}_i||^2\leq c, i=1,2,...,k. \tag{2.27}$$

Let $\boldsymbol{\gamma} = [\gamma_1, ..., \gamma_k]$, and γ_i be the Lagrange multiplier associated with the ith inequality constraint $\|\mathbf{d}_i\|^2 \leq c$, then the Lagrange dual function is given by

$$g(\boldsymbol{\gamma}) = \inf_{\mathbf{D}} L(\mathbf{D}, \boldsymbol{\gamma}) = \inf_{\mathbf{D}} \{\|\mathbf{X} - \mathbf{D}\mathbf{A}\|_F^2 + \mu \mathrm{Tr}(\mathbf{D}^T \mathbf{L}^D \mathbf{D}) + \sum_{i=1}^{k} \gamma_i (\|\mathbf{d}_i\|^2 - c)\}. \tag{2.28}$$

Let $\boldsymbol{\Gamma}$ be the $k \times k$ diagonal matrix whose diagonal entry $\Gamma_{ii} = \gamma_i$ for all i. Then $L(\mathbf{D}, \boldsymbol{\gamma})$ can be written as

$$\begin{aligned} L(\mathbf{D}, \boldsymbol{\gamma}) &= \|\mathbf{X} - \mathbf{D}\mathbf{A}\|_F^2 + \mathrm{Tr}(\mathbf{D}^T \mathbf{D} \boldsymbol{\Gamma}) c(\mathrm{Tr}(\boldsymbol{\Gamma})) + \mu \mathrm{Tr}(\mathbf{D}^T \mathbf{L}^D \mathbf{D}) \\ &= \mathrm{Tr}(\mathbf{X}^T \mathbf{X}) - 2\mathrm{Tr}(\mathbf{D}^T \mathbf{X} \mathbf{A}^T) + \mathrm{Tr}(\mathbf{A}^T \mathbf{D}^T \mathbf{D} \mathbf{A}) + \mathrm{Tr}(\mathbf{D}^T \mathbf{D} \boldsymbol{\Gamma}) \\ &- c(\mathrm{Tr}(\boldsymbol{\Gamma})) + \mu \mathrm{Tr}(\mathbf{D} \mathbf{D}^T \mathbf{L}^D) \end{aligned} \tag{2.29}$$

The optimal solution $\mathbf{D}^*$ can be obtained by letting the first order derivative of (Equation 2.29) equal to zero, i.e.

$$\mathbf{D}^* \mathbf{A} \mathbf{A}^T - \mathbf{X} \mathbf{A}^T + \mathbf{D}^* \boldsymbol{\Gamma} + \mu \mathbf{L}^D \mathbf{D}^* = 0, \tag{2.30}$$

$$\mathbf{D}^* = \mathbf{X} \mathbf{A}^T (\mathbf{A} \mathbf{A}^T + \boldsymbol{\Gamma} + \mu \mathbf{L}^D)^{-1}. \tag{2.31}$$

Substituting (Equation 2.31) into (Equation 2.29), the Lagrange dual function becomes (Equation 2.32).

$$g(\boldsymbol{\gamma}) = \mathrm{Tr}(\mathbf{X}^T \mathbf{X}) - 2\mathrm{Tr}(\mathbf{D}^T \mathbf{X} \mathbf{A}^T) - c(\mathrm{Tr}(\boldsymbol{\Gamma})) + \mathrm{Tr}(\mathbf{X} \mathbf{A}^T (\mathbf{A} \mathbf{A}^T + \boldsymbol{\Gamma} + \mu \mathbf{L}^D)^{-1} \mathbf{A} \mathbf{X}^T) \tag{2.32}$$

By optimizing function (Equation 2.32) with $\mathbf{\Gamma}$, we get (Equation 2.33).

$$\begin{gathered}\min_{\mathbf{\Gamma}} \mathrm{Tr}(\mathbf{X}\mathbf{A}^{\mathsf{T}}(\mathbf{A}\mathbf{A}^{\mathsf{T}} + \mathbf{\Gamma} + \mu\mathbf{L}^{D})^{-1}\mathbf{A}\mathbf{X}^{\mathsf{T}}) + c(\mathrm{Tr}(\mathbf{\Gamma})) \\ \text{s.t. } \gamma_i \geq 0, i = 1, ..., k.\end{gathered} \tag{2.33}$$

By solving this problem with Newton's method, we get the optimal dictionary $\mathbf{D}^* = \mathbf{X}\mathbf{A}^{\mathsf{T}}(\mathbf{A}\mathbf{A}^{\mathsf{T}} + \mathbf{\Gamma}^* + \mu\mathbf{L}^{D})^{-1}$, where $\mathbf{\Gamma}^*$ is the optimal solution from (Equation 2.33).

2.7.2.2 Least-Angle Regression with Block Coordinate Descent

Least-angle regression (LARS) [65] is a regression method that provides a general version of forward selection, which is highly efficient in solving LASSO [66]. We follow the steps presented in [67]. In step 7 of Algorithm 2, instead of using the ordinary least square solution (Equation 2.34), we utilize the dual graph regularized least square solution (Equation 2.35) to incorporate graph information.

Substitute $\boldsymbol{\alpha}^{(k+1)}_{\mathrm{OLS}}$ with $\boldsymbol{\alpha}^{(k+1)}_{\mathrm{DGLS}}$

$$\boldsymbol{\alpha}^{(k+1)}_{\mathrm{OLS}} = (\mathbf{D}^{\mathsf{T}}_{\mathcal{A}}\mathbf{D}_{\mathcal{A}})^{-1}\mathbf{D}^{\mathsf{T}}_{\mathcal{A}}\mathbf{y} \tag{2.34}$$

$$\boldsymbol{\alpha}^{(k+1)}_{\mathrm{DGLS}} = (\mathbf{D}^{\mathsf{T}}_{\mathcal{A}}\mathbf{D}_{\mathcal{A}} + \lambda {L^{A}}_{kk}\mathbf{I})^{-1}(\mathbf{D}^{\mathsf{T}}_{\mathcal{A}}\mathbf{x} - \mathbf{h}_k/2) \tag{2.35}$$

Where $\mathbf{I}$ is identity matrix, $\mathbf{h}_k = 2\lambda(\sum_{k\neq j} {L^{A}}_{kj}\boldsymbol{\alpha}_j)$, k is the kth step and $\mathcal{A}$ is the active set in LARS algorithm.

By fixing coefficients matrix $\mathbf{A}$, we have the same objective function as (Equation 2.27). We use block coordinate descent with warm restarts for dictionary update that doesn't need any learning rate tuning process. Since block coordinate descent method converges in polynomial time, we update dictionary $\mathbf{D}$ with each input sparse vector $\boldsymbol{\alpha}_i$. Different from the block coordinate descent method used in [68], which updates one column vector $\mathbf{d}_j$ of dictionary $\mathbf{D}$ at a time, our method updates one row vector $\mathbf{d}_{j\cdot}$ of $\mathbf{D}$ each time to incorporate the feature graph constraint. We can rewrite objective function (Equation 2.27) as follows

$$\mathcal{O} = \frac{1}{2}\sum_{i=1}^{n}(\mathbf{x}_{i\cdot} - \mathbf{d}_{i\cdot}\mathbf{A})^2 + \mu \sum_{i,j=1}^{n} \mathbf{L}^D_{ij}\mathbf{d}^T_{j\cdot}\mathbf{d}_{i\cdot} \tag{2.36}$$

By setting first order derivative of $\mathcal{O}$ with respect to row vector $\mathbf{d}_{j\cdot}$ equal to 0, we have (Equation 2.37).

$$\begin{aligned} 0 &= (\mathbf{d}_{j\cdot}\mathbf{A} - \mathbf{x}_{j\cdot})\mathbf{A}^T + \mu(\sum_{i \neq j}^{n} L^D_{ij}\mathbf{d}_{i\cdot} + L^D_{jj}\mathbf{d}_{j\cdot}) \\ \Rightarrow \mathbf{d}_{j\cdot} &= [\mathbf{x}_{j\cdot}\mathbf{A}^T - \mu((\mathbf{L}^D_j)^T\mathbf{D} - L^D_{jj}\mathbf{d}_{j\cdot})](\mathbf{A}\mathbf{A}^T + \mu L^D_{jj}\mathbf{I})^{-1} \end{aligned} \tag{2.37}$$

where $\mathbf{d}_{i\cdot}$ and $\mathbf{d}_{j\cdot}$ are rows of dictionary matrix $\mathbf{D}$. We then normalize the row vector $\mathbf{d}_{j\cdot}$ to let $\|\mathbf{d}_{j\cdot}\|_2 \leq 1$.

2.8 EXPERIMENTAL RESULTS

We conducted extensive experiments to demonstrate the superior performance of our proposed algorithms in image restoration (denoising and deblurring) and representation (clustering and classification).

2.8.1 Image Restoration

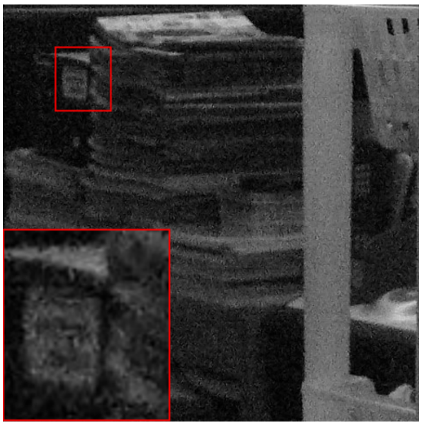
To demonstrate the effectiveness of our proposed MCA-GSC algorithm in image denoising and deblurring. We have compared our algorithm with other popular image restoration algorithms on benchmark natural images with synthesized noise and blur. We also run our MCA-GSC algorithm on real sensor noise and blur images to demonstrate its robust performance.

2.8.1.1 Image Denoising

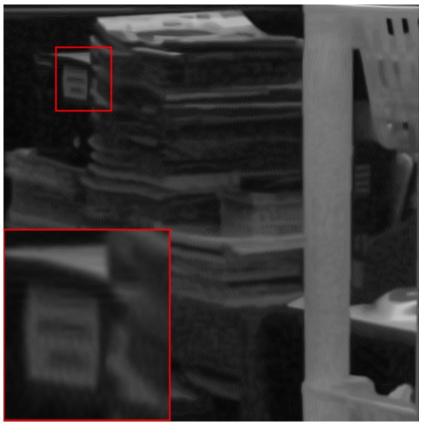
We evaluate our MCA-GSC algorithm using natural images: five 512×512 and one 256×256 benchmark images (in grayscale)–*Lena, Barbara, House (*256×256*), Boat, Peppers,* and *Mandrill*. We compare our MCA-GSC with three recent methods: OGLR [19], SSC-GSM [2], and BM3D [69].

For image denoising evaluation, we corrupted test images by i.i.d. AWGN with standard deviation $\sigma_{\mathcal{I}}$ ranging from 10 to 70. For each $\sigma_{\mathcal{I}}$, we averaged corresponding performence, in terms of PSNR (in dB) and the SSIM index [70], which are tabulated in Table Table I.

Besides benchmark images with synthesized noise, we also performed our algorithm on a real senor noise image, where noise level is estimated by algorithm provided in [71]. Noisy image and denoised pair by our MCA-GSC algorithm are shown in Figure 6.



Real sensor noise image



Denoised by MCA-GSC

Figure 6. Denoise real senor noise image with proposed MCA-GSC

The sparse coding part and graph Laplacian Regularizer part of our MCA-GSC algorithm are all iterative denoising method. We show in Fig Figure 7 the impact of iterative noise reduction, our MCA-GSC algorithm uses 11 iteration to get stable results.

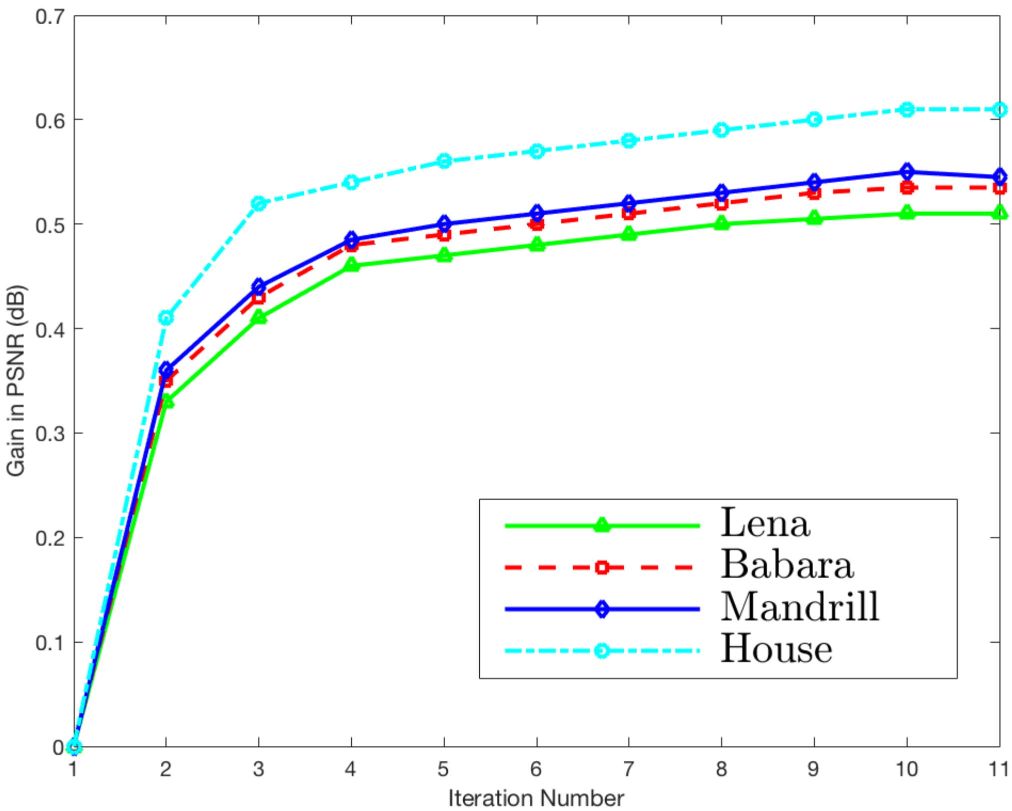
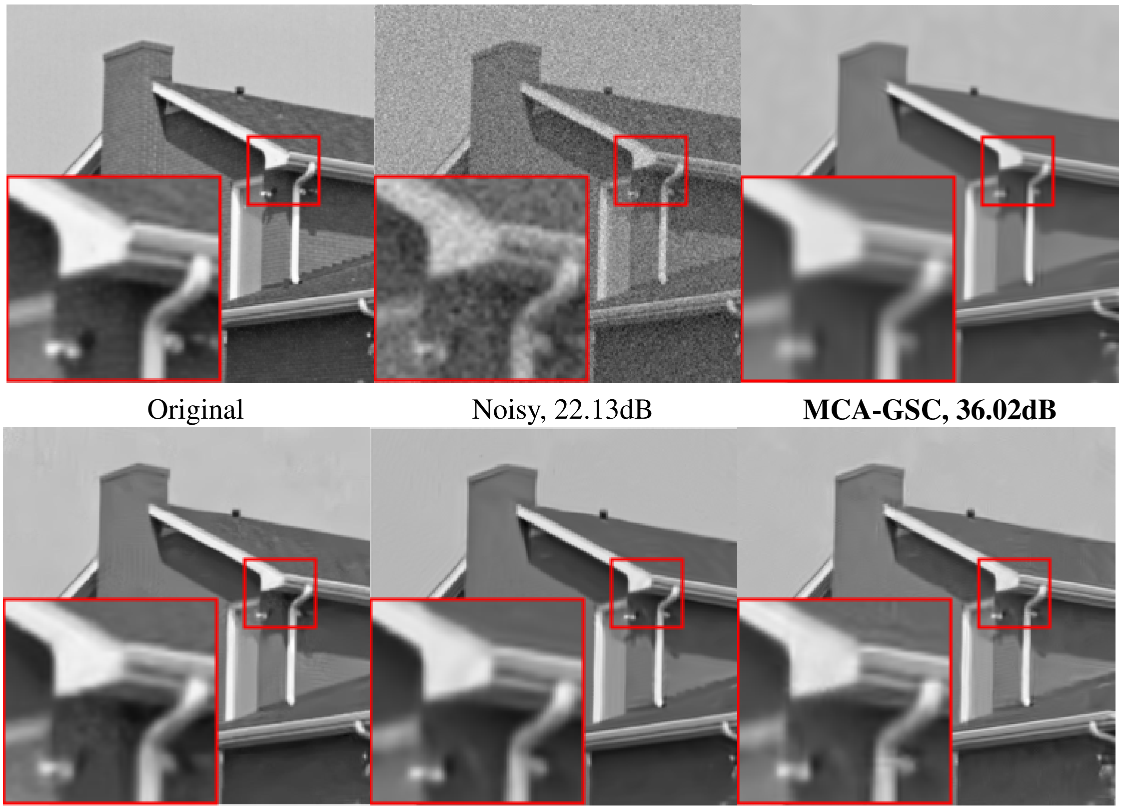


Figure 7. Relation between Iteration Number and Gain in PSNR with AWGN $\sigma = 20$



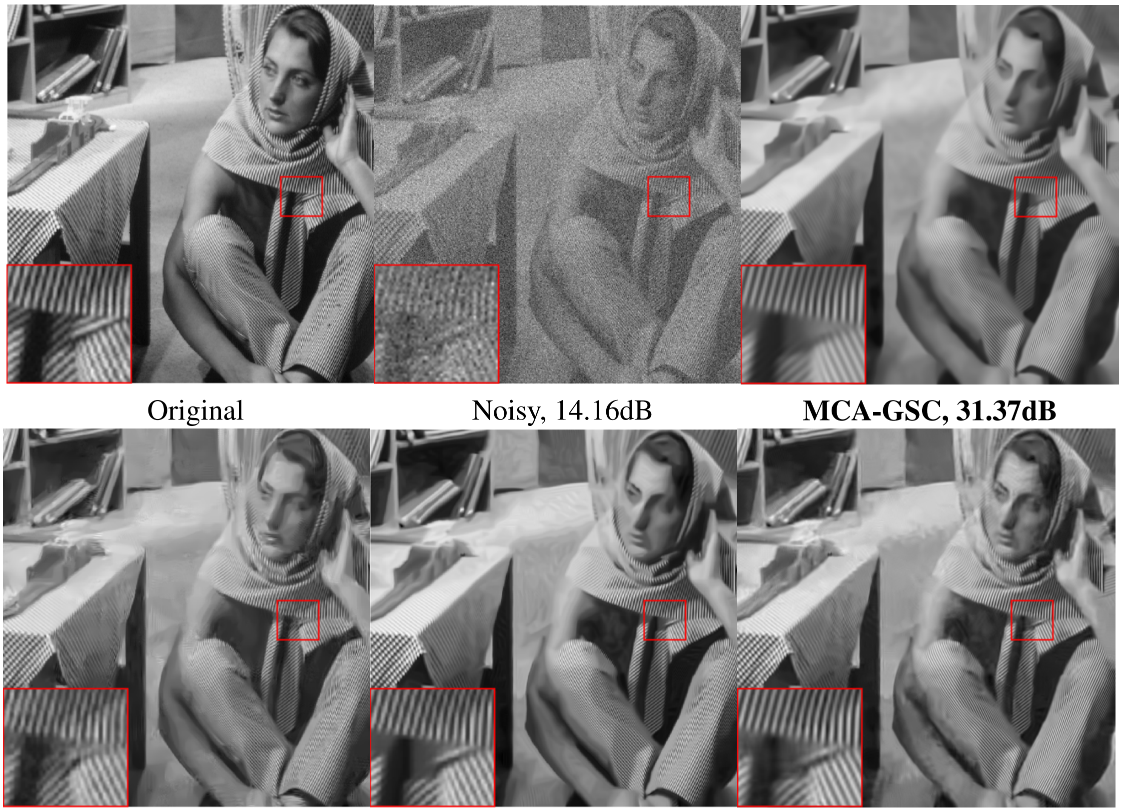
Original Noisy, 22.13dB **MCA-GSC, 36.02dB**

OGLR, 33.79dB SSC-GSM, 34.05dB BM3D, 33.77dB

Figure 8. Denoising of the natural image *Hourse*, where the original image is corrupted by AWGN with $\sigma_{\mathcal{I}} = 20$

Fig Figure 8, Figure 9 and Figure 10 are visual comparisons between our MCA-GSC algorithm and three recent state-of-the-art denoising algorithms. We found that although our algorithm achieves a relatively high PSNR, it's not the most visually pleasant one. Our algorithm tends to blur the texture a little bit and looks more blurry compared with BM3D and SSC-GSM algorithms.

From the results on benchmark images with synthesized i.i.d Gaussian noise in Table Table I, we can find out that our MCA-GSC significantly outperforms the other 3 state-of-the-art image denoising



Original | Noisy, 14.16dB | **MCA-GSC, 31.37dB**

OGLR, 27.09dB | SSC-GSM, 27.45dB | BM3D, 27.23dB

Figure 9. Denoising of the natural image *Barbara*
The original image is corrupted by AWGN with $\sigma_{\mathcal{I}} = 50$

algorithms in PSNR. However, the performance is not as good if evaluated by SSIM index [70]. PSNR is about variance of image difference, while SSIM index is more related to image structure similarity. We are unclear why our algorithm gets low score in SSIM index, but one reasonable thought is the process of decomposition of image into two different layers could potentially damage the structure of original image.

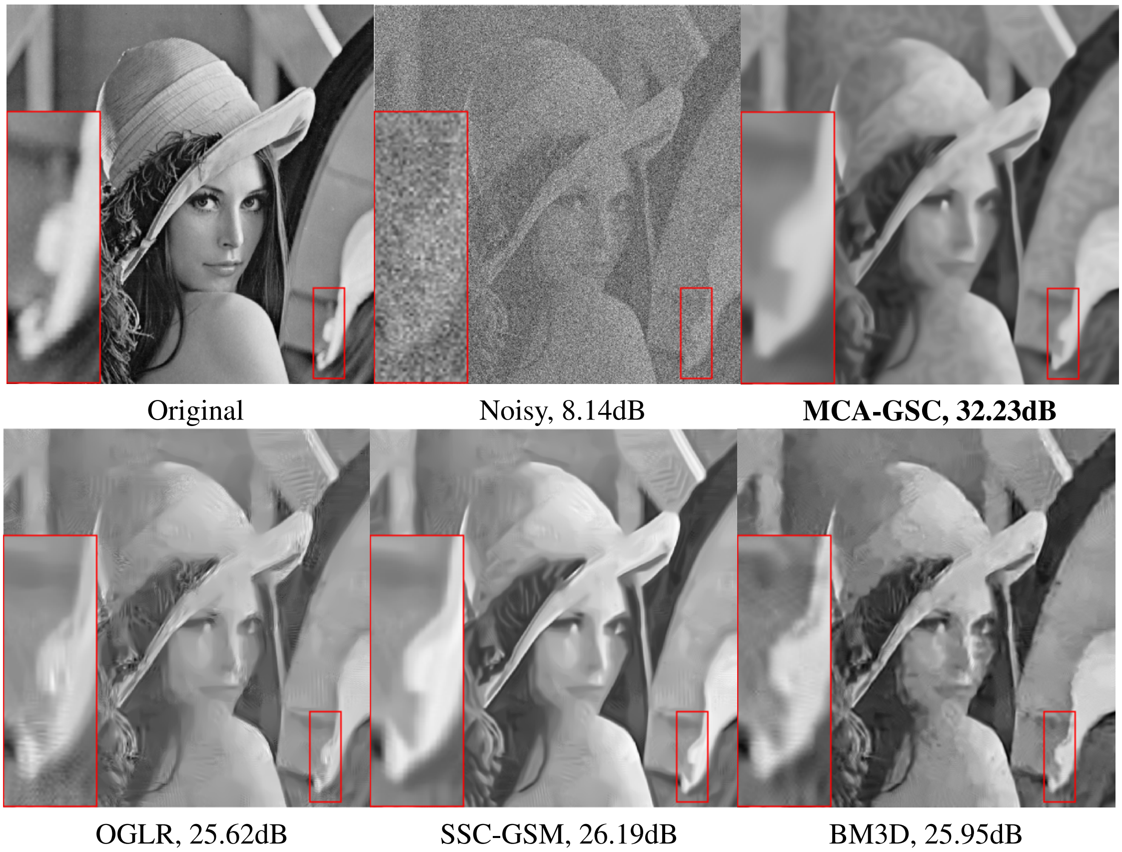


Figure 10. Denoising of the natural image *Lena*
The original image is corrupted by AWGN with $\sigma_{\mathcal{I}} = 100$

2.8.1.2 Image Deblurring

For image deblur evaluation, we corrupted Ten 512×512 natural images by Gaussian blur with standard deviation $\sigma_{\mathcal{I}} = 1.6$. We compare our MCA-GSC with three other deblur algorithms: FISTA [72], IDD-BM3D [73], and SSC-GSM [2], the average performance results, in PSNR (in dB) and SSIM index are tabulated in Table Table III.

Fig Figure 12 shows image deblur visual performance of our proposed MCA-GSC algorithm on three classic benchmark images: *Cameraman, Barbara and Lena*.

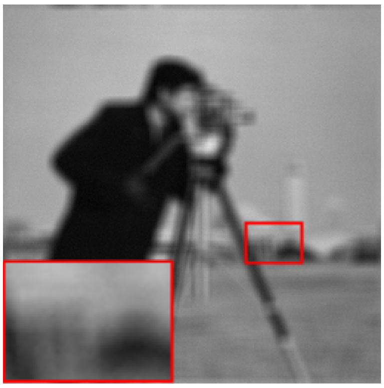
We also applies our algorithm onto real sensor blur image. We use the real blur image case in [3] with its estimated kernel. Figure 11 shows the visual result of real sensor blur image deburred by our MCA-GSC algorithm.

Real sensor blur image | Denoised by MCA-GSC

Figure 11. Restore real senor blur image [3] with proposed MCA-GSC

2.8.1.3 Running Time

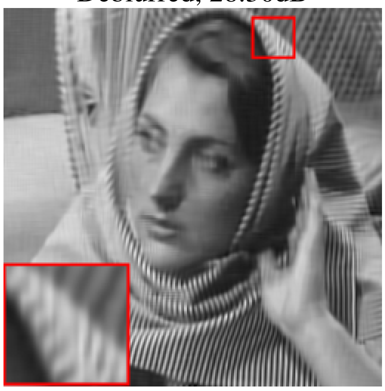
The proposed MCA-GSC algorithm was implemented under Matlab and running on a Windows 10 laptop with i7 2.6 GHz CPU and 8GB RAM. for a gray 512×512 image, the running time of image denoising is $400 \sim 600$ seconds, the running time of image deblurring is $900 \sim 1500$ seconds.



Blurred, 20.76dB

Deblurred, 28.30dB

Blurred, 22.42dB



Deblurred, 29.01dB

Blurred, 22.89dB

Deblurred, 31.51dB

Figure 12. Image deblur by proposed MCA-GSC algorithm on *Cameraman, Barbara and Lena*

2.8.2 Image Representation (Previously published as Lingdao Sha, Dan Schonfeld (2017) Dual graph regularized sparse coding for image representation, 2017 IEEE Visual Communications and Image Processing (VCIP), 14.)

We apply our dual graph regularized sparse coding algorithm into application of image clustering and classification on 4 benchmark datasets: CMU-PIE, COIL20, COIL100, USPS. Data sample of those 4 datasets are shown in Figure 13.

2.8.2.1 Image Clustering

We investigate the clustering performance of our proposed DGSC method on 3 real world datasets: CMU-PIE, COIL20 and COIL100 [1]. CMU-PIE face database contains 1428 images of 68 objects (21 images per object); COIL20 contains 1440 images of 20 objects (72 images per object with rotation from 5to360); COIL100 complete the COIL20 with additional 80 objects and consists of 7200 images. For all of the above three datasets, each image is 32×32 with 256 grey levels per pixel.

To evaluate the clustering results, we adopt the two standard performance measures which are widely used for clustering: Clustering Accuracy (CA) and Normalized Mutual Information (NMI) [4].

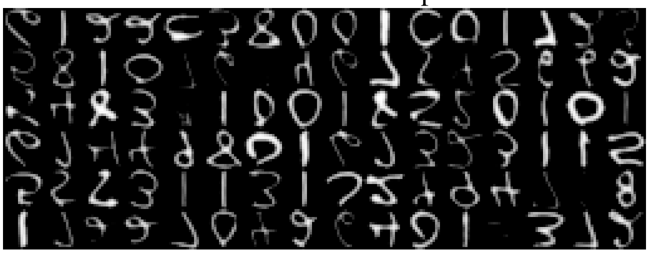
We compare the clustering results of our DGSC method with Kmeans, Normalized Cut (NCut) [55], Sparse Coding (SC), and GraphSC [35]. We also compare the computation efficiency of our proposed optimization algorithms FS-LD and LARS-BCD.

---

[1] http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html

CMU-PIE Dataset Sample

COIL Dataset Sample



USPS Dataset Sample

Figure 13. Samples of CMU-PIE, COIL, and USPS Datasets

In order to show the consistency of clustering performance, we further divide each dataset into 5 ascending sub-datasets. For example, sub-dataset of CMU-PIE has clusters ranging from 4 to 68. Table **Table V**, **Table VI** and **Table VII** are the clustering results on CMU-PIE, COIL20 and COIL100 dataset respectively.

From the image clustering experiments, our DGSC method outperforms all other listed methods on all three datasets, with a slight improvement on CMU-PIE and relatively significant improvements on

COIL-20 and COIL100 datasets. Those promising results corroborate with our thought that both data and feature graph are helpful in image data representation.

We then compare the clustering computational efficiency of FS-LD and LARS-BCD on all three image datasets. Both algorithms are implemented with Matlab and running on a Windows 10 machine with Intel Core i7 2.6GHz CPU and 8GB RAM. While FS-LD algorithm iteratively updates coefficient matrix $\mathbf{A}$ and dictionary matrix $\mathbf{D}$ until converge, LARS-BCD algorithm updates dictionary matrix $\mathbf{D}$ with each sparse vector input $\boldsymbol{\alpha}_i$. As a result, LARS-BCD needs less sparse coding steps than FS-LD, meaning LARS-BCD looks more efficient in image clustering tasks than FS-LD on all three datasets. Figure **Figure 14** illustrates the result.

2.8.2.2 Image Classification

Our proposed method not only enhances image clustering but also improves image classification results. We present experiments on the benchmark USPS handwritten digits dataset [1], which is composed of 7291 training images and 2007 testing images of size 16×16.

The process of using our DGSC method to enhance classification is straightforward, instead of training and testing directly on original image matrix, we do training and testing on the coefficient matrix from DGSC methods. To evaluate the classification performance of our method, we use public available software LIBSVM [74] to train linear SVM classifiers on subset of training set coefficient vectors $\boldsymbol{\alpha}_i$ ($i \in \{1, ..., 7291\}$) and then test on testing set coefficient vectors $\boldsymbol{\alpha}_j$ ($j \in \{7292, ..., 9298\}$). Parameters are tuned by grid search and best pair are used for each corresponding case. We train linear SVM with

[1] http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html

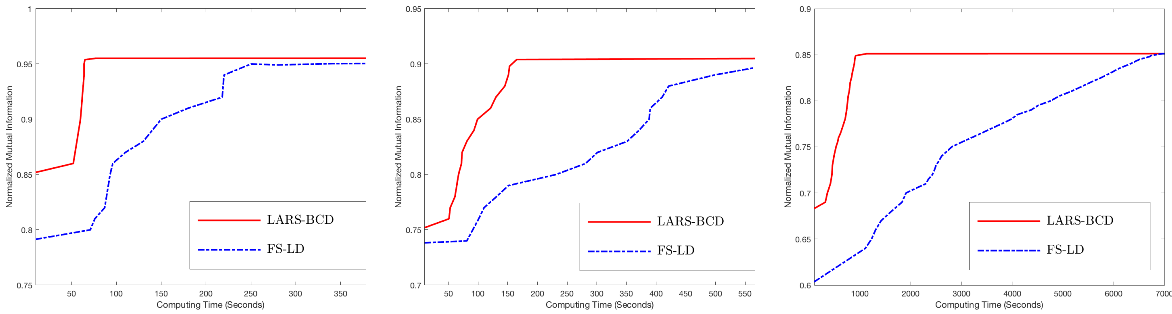
5-folds cross-validation on training dataset and test on the testing dataset. Table Table IV shows the test results on original image, coefficients from sparse coding (SC), coefficients from GrapgSC [35] and coefficinet from our DGSC algorithm. As can be seen, our algorithm outperforms the rest 3 cases. This assures that data graph and feature graph are both important in capturing discriminative features of the images.

2.8.2.3 Summary of DGSC

Kmeans, Normalized Cut, Sparse coding, are 3 very popular and robust algorithm in data clustering. However, from the experiment results, even the graph regularized sparse coding method outperform those 3 algorithms in a relatively large margin. Our dual-graph regularized sparse coding method even outperforms the graph regularized sparse coding by adding one more graph constraint for dictionary. This constraint is useful in control of the variation of dictionary elements. The improvement on CMU-PIE dataset is relatively small, however, the advantage really shines in COIL data set. The reason behind it is: COIL datasets contain objects with 72 orientations which is hard to capture the similarity between two images using Euclidean distance. In our dual graph-regularized sparse coding method, we add a feature constraint to control the variation of dictionary matrix, which can result in more robust mapping from high-dimension image data to coefficients. Our dual graph regularized sparse coding method can also be used to enhance the image classification results, as shown in table Table IV.

2.9 Conclusion

In this paper, we present a generalized graph Laplacian regularized sparse coding framework for both image restoration and image representation problems. We come up with a relatively novel image decomposition restoration algorithm by specializing our generalized framework. With the help of MCA



(CMU-PIE) (COIL20) (COIL100)

Figure 14. Computation time vs NMI [4]
Run time comparison of LARS-BCD and FS-LD optimization algorithms on three datasets, evaluated by NMI

algorithm, we are able to decompose natural image into cartoon layer (piece-wise-smooth) and textur layer (texture and sharp edges), we then restore each layer with optimal graph Laplacian regularizer and sparse coding algorithm respectively. Our proposed algorithm outperforms state-of-the-art image denoising algorithms in PSNR, also competitive in image deblurring. However, compared with state-of-the-art image denoising algorithms such as BM3D and SSC-GSM, our algorithm is relatively poor in SSIM index evaluation. We also developed a novel dual graph regularized sparse coding (DGSC) algorithm as another special case application (image representation) of our generalized framework. By considering both data graph and feature graph as regularizers in sparse representation, our learned co-efficient vector is both efficient and effective in image clustering and classification tasks. Experimental results in image clustering and classification on 4 popular bench mark datasets show that our proposed algorithm outperforms 4 popular algorithms in image clustering, also enhances the linear SVM classification results. We also proposed two efficient numerical optimization algorithm for application of our DGSC algorithm.

TABLE I

NATURAL IMAGE DENOISING WITH MCA-GSC

In each cell, the PSNR and SSIM index [70] evaluation results of the four denoising methods are reported in the following order: TOP LEFT:OGLR [19]. TOP RIGHT: BM3D [69]. BOTTOM-LEFT:SSC-GSM [2]. BOTTOM RIGHT: (**Proposed**) MCA-GSC

Image	OGLR / SSC-GSM		BM3D / MCA-GSC		**Standard Deviation $\sigma_{\mathcal{I}}$**									
	10		20		30		40		50		60		70	
Lena	35.62	35.89	32.93	33.02	31.22	31.23	30.06	29.82	28.86	29.00	28.19	28.20	27.46	27.50
	0.912	**0.915**	0.874	0.876	0.842	0.843	0.821	0.813	0.785	0.796	0.768	0.776	0.742	0.756
	35.96	**37.96**	32.58	**35.66**	31.40	**34.45**	30.32	**33.68**	29.05	**33.12**	28.34	**32.70**	27.63	**32.46**
	0.915	0.893	**0.879**	0.843	**0.848**	0.812	**0.823**	0.783	**0.797**	0.762	**0.788**	0.745	**0.764**	0.730
Barbara	34.46	34.96	31.45	31.75	29.63	29.79	28.31	28.00	27.36	27.23	26.42	26.30	25.62	25.51
	0.937	**0.942**	0.902	0.905	0.867	0.867	**0.838**	0.822	**0.801**	0.794	0.768	0.759	0.734	0.727
	35.27	**37.00**	32.06	**34.06**	29.69	**32.66**	28.37	**31.86**	27.45	**31.37**	26.56	**31.10**	25.59	30.95
	0.941	0.925	**0.910**	0.860	**0.873**	0.801	0.835	0.746	0.799	0.700	**0.793**	0.660	**0.763**	0.625
House	36.57	36.71	33.80	33.77	32.05	32.09	30.68	30.65	29.20	29.69	28.42	28.74	27.57	27.91
	0.923	0.907	**0.877**	0.869	0.845	0.850	0.822	0.813	0.784	0.797	0.766	0.779	0.738	0.763
	36.70	**38.28**	34.08	**35.95**	32.44	**35.06**	31.15	**34.39**	30.02	**33.76**	0.0	**32.97**	0.0	**32.40**
	0.912	0.883	0.871	0.846	**0.851**	0.825	**0.829**	0.805	**0.801**	0.785	**0.784**	0.761	**0.768**	0.736
Boat	32.83	33.92	29.45	30.88	27.55	29.12	26.25	27.74	25.25	26.78	24.50	26.02	23.85	25.4
	0.914	0.895	**0.840**	0.803	**0.780**	0.771	**0.733**	0.717	0.685	0.674	0.651	0.634	0.618	0.625
	33.89	**35.67**	30.87	**33.46**	28.88	**32.65**	27.73	**32.19**	26.75	**31.89**	25.98	**31.66**	25.52	**31.48**
	0.882	0.843	0.818	0.756	0.757	0.700	0.724	0.651	**0.689**	0.618	0.665	0.593	0.645	0.572
Peppers	34.91	35.02	32.67	32.75	31.23	31.23	30.10	29.93	28.83	29.09	28.20	28.26	27.42	27.54
	0.879	**0.879**	0.842	0.845	0.818	0.820	0.798	0.795	0.762	0.782	0.751	0.763	0.729	0.746
	34.83	**37.22**	31.41	**35.46**	31.57	**34.60**	30.22	**34.02**	26.82	**33.58**	28.31	**33.26**	27.49	**33.01**
	0.873	0.858	**0.846**	0.825	**0.825**	0.802	**0.807**	0.781	**0.785**	0.761	**0.778**	0.744	**0.765**	0.728
Mandrill	33.84	33.58	31.35	31.60	30.56	30.56	27.40	27.09	26.59	26.35	25.99	25.74	25.47	25.28
	0.883	**0.897**	0.786	**0.792**	**0.706**	0.702	**0.650**	0.617	**0.595**	0.549	**0.546**	0.498	**0.500**	0.459
	34.06	33.27	**31.99**	30.74	**30.64**	29.94	27.43	**29.62**	26.76	**29.48**	25.89	**29.41**	25.91	**29.36**
	0.886	0.851	0.784	0.677	0.673	0.520	0.602	0.410	0.520	0.361	0.484	0.339	0.439	0.324

TABLE II

DESCRIPTION OF THE CLUSTERING DATASETS

Data Set	# images	# pixels	# classes
CMU-PIE	1428	1024	68
COIL20	1440	1024	20
COIL100	7200	1024	100

TABLE III

NATURAL IMAGE DEBLUR WITH MCA-GSC

Performance comparison of our proposed MCA-GSC algorithm against 3 popular image deblurring algorithms: FISTA, IDD-BM3D, and SSC-GSM.

Images / Algorithms	Gaussian blur with standard deviation 1.6, $\sigma_{\mathcal{I}} = \sqrt{2}$									
	Butterfly	*Boats*	*C. Man*	*Starfish*	*Parrot*	*Lena*	Barbara	Peppers	Leaves	House
FISTA [72]	30.36 0.937	29.36 0.851	26.80 0.824	29.65 0.888	31.23 0.907	29.47 0.854	25.03 0.738	29.42 0.835	29.33 0.948	31.50 0.825
IDD-BM3D [73]	30.73 0.947	31.68 0.904	28.17 0.871	31.66 0.916	32.89 0.932	31.45 0.910	27.19 0.823	29.99 0.881	31.40 0.964	34.08 0.882
SSC-GSM [2]	31.12 **0.952**	31.78 **0.905**	**28.40** **0.872**	32.26 **0.925**	**33.30** **0.938**	31.52 **0.911**	28.42 **0.846**	30.18 **0.877**	32.02 **0.969**	**34.65** **0.883**
MCA-GSC	**31.26** 0.932	**31.91** 0.878	28.31 0.839	**32.35** 0.903	33.18 0.911	**31.58** 0.896	**28.80** 0.822	**30.36** 0.852	**32.26** 0.950	34.41 0.871

TABLE IV

CLASSIFICATION ACCURACY ON USPS

N is the size of training set, M is methods

N / M	Classification Accuracy (%)					
	100	500	1000	2000	5000	7291
Orignal	75.1	87.2	89.6	90.3	91.5	93.0
SC	80.3	87.8	88.2	89.5	91.7	93.5
GraphSC	82.8	88.6	90.3	91.3	93.9	94.1
DGSC	**82.6**	**89.9**	**90.5**	**92.0**	**94.4**	**95.8**

TABLE V. CLUSTERING RESULTS ON CMU-PIE DATASET

#C IS # OF CLUSTER, M IS METHOD

M / **#C**	Accuracy (%)					Normalized Mutual Information (%)				
	Kmeans	NCut	SC	GraphSC	**DGSC**	Kmeans	NCut	SC	GraphSC	**DGSC**
4	48.6	99.1	100	100	**100**	33 ± 5.6	98.6	100	100	**100**
20	38.4	78.3	88.4	94.2	**95.1**	55 ± 3.3	88.6	91 ± 1.1	96 ± 1.3	$\mathbf{97 \pm 1.1}$
36	34.9	75.6	77.9	88.5	**90.5**	60 ± 3.9	88.9	88 ± 2.3	95 ± 1.2	$\mathbf{95 \pm 1.5}$
52	33.2	72.6	69.6	84.6	**86.7**	61 ± 2.2	89.1	83 ± 2.1	93 ± 1.2	**95.6**
68	31.7	70.8	60.4	82.1	**84.3**	55	88.3	77.6	93.4	**95.2**

TABLE VI. CLUSTERING RESULTS ON COIL20 DATASET

#C IS # OF CLUSTER, M IS METHOD

M / **#C**	Accuracy (%)					Normalized Mutual Information (%)				
	Kmeans	NCut	SC	GraphSC	**DGSC**	Kmeans	NCut	SC	GraphSC	**DGSC**
4	80.1	79.4	90.2	96.8	**96.5**	73 ± 3.3	75	84 ± 6.3	89 ± 2.8	$\mathbf{97 \pm 1.1}$
8	72.4	70.9	84.6	89.2	**91.7**	64 ± 4.6	73.4	82 ± 5.2	85 ± 1.7	$\mathbf{94 \pm 1.1}$
12	68.1	74.1	81.9	90.1	**91.1**	68 ± 3.9	79.1	82 ± 2.4	84 ± 1.4	$\mathbf{92.3 \pm 1.2}$
16	72.1	70.9	74.7	83	**89**	73 ± 2.5	77.7	81 ± 3.3	83 ± 2.1	$\mathbf{91 \pm 0.6}$
20	62.2	67.1	73.8	78.6	**85.5**	72.4	77.3	78.5	83.3	**90.2**

TABLE VII. CLUSTERING RESULTS ON COIL100 DATASET
#C IS # OF CLUSTER, M IS METHOD

M / **#C**	Accuracy (%)					Normalized Mutual Information (%)				
	Kmeans	NCut	SC	GraphSC	**DGSC**	Kmeans	NCut	SC	GraphSC	**DGSC**
20	63.4	67.3	68.9	78.4	**85 ± 1.1**	71.2 ± 5.6	77.3	78 ± 0.5	82 ± 1.5	**89 ± 1.5**
40	51.4	64.3	61.4	74.3	**80 ± 1.1**	66 ± 3.3	78.4	76 ± 1.1	80 ± 1.3	**88 ± 0.4**
60	49.4	62.1	65.9	71.3	**77 ± 0.9**	69 ± 3.9	75.4	75 ± 1.3	80 ± 1.5	**86 ± 0.8**
80	53.9	59.9	62.0	69.6	**75 ± 1.1**	75 ± 1.2	73.7	72 ± 0.8	79 ± 1.2	**86 ± 0.3**
100	50.5	62.3	60.4	69.3	**74.7**	74.3	75.1	72.1	78.6	**85.5**

CHAPTER 3

KRONECKER LEAST ANGLE REGRESSION

3.1 ABSTRACT

Sparse representation of signals has draw great attention recently with the finding that we can well approximate a signal by linear combination of dictionary elements. Enormous research has been devoted to find the sparsest solution of a singal by solving a underdetermined linear system of algebraic equations. Least Angle Regression (LARS) is one of the most important algorithms, which can be used to solve least square problem with both ℓ_0 norm and ℓ_1 norm constrain. However, most modern signals are two dimensional or even higher, current state-of-the-art algorithms are relying on converting high dimensional data into vectors which are not considering structure information within data.

Research has shown that signals with multidimensional structure can be converted from vectors to tenors (mutliway arrays) with the utilization of Tucker model. Thus, the problem is converted to solving an underdetermined linear system with Kronecker structure. In this paper, we proposed Kronecker Least Angle Regression (Kron-LARS) algorithm as a generalization of the classic vector version (LARS) algorithm for tensors. We demonstrate that by utilizing the multidimensional structure of signal, Kron-LARS as a equivalent conversion of LARS can reduce the recovery complexity and memory usage. Additionally, by exploiting not only the Kronecker structure but also block sparisty of signals, our Kron-LARS can be easily extened as N-dimensional block sparse LARS (NBS-LARS), which is dramatically

fast and memory efficient. We demonstrate that NBS-LARS algorithm not only has considerably lower recovery complexity but also has better precision under same percentage of sampling.

We compared the recovery results and efficiency of our Kron-LARS and NBS-LARS algorithms with classical LARS algorithm on 1D, 2D and 3D signals. We also demonstrate that compared with exiting greedy algorithms, our algorithms preserve the good property of LARS algorithm which can be used to solve ℓ_1 norm (better precision and recovery success rate – convex optimization) with the efficiency of greedy algorithms. The NBS-LARS algorithm is a very fast solution for compressed sensing (CS) problem with large-scale data sets, such as 2D compressive imaging (CI), 3D MRI and hyperspectral CI, we show examples with real-world signals.

3.2 Introduction

Sparsity is one the most remarkable concept underlines recent developments of image and signal processing. Compression protocols such are JPEG, MPEG and JPEG2000 that are utilizing sparsity of signals have had big success delivering high compression ratios with minimal loss of information. The reason behind is by nature, signals living in a vector space that do not uniformly covering the entire space [75]. Specificly, most applications in getting signals (images) of interest can be well approximated by a linear combination of few active elements of a dictionary [76]. Those dictionaries or bases is a super set that can be used to represent the internal structure of sigals. Moreover, dictionaries could be overcomplete where the number of atoms is large than signal size. Most well know dictionaries are, for example, those generated from Discrete Cosine Transform (DCT) [77], the Wavelet Transform (WT) [78], etc.

Central to much of recent work about sparsity representation is the paradigm of compressed sensing (CS), also know under the concept of compressed sensing or compressive sampling [79–81]. Compared with traditional signal acquisition process which are based on sampling of analog singals at a high Nyquist rate and only storing the most significant coefficients, CS theory is a more compelling idea that the sampling process can be greatly simplified by fetching relatively few linear measurements of the signal while still allowing almost perfect reconstruction via nonlinear recovery process. The first intuitive approach to a reconstruction algorithm consists in searching for the sparsest vector that is consistent with the linear measurements. However, this is a ℓ_0-minimization problem which is NP-hard in general and mostly computationally infeasible. There are essentially two approaches for tractable alternative algorithms. The first is convex relaxation, leading to ℓ_1-minimization [82], also known as basis pursuit [83], whereas the second constructs greedy algorithms such as MP [84] and OMP [85]. Both convex relaxation and greedy algorithms have their advantages and disadvantages. Thoery for sparse recovery by ℓ_1-minimization is well established known as convex optimization, and basic properties of the measurement matrix are also well known: the null space property (NSP) [86] and the restricted isometry property (RIP) [87]. Also, there exists a wide range of cases where ℓ_1-minimization is known to find the sparsest solution while greedy algorithms such as OMP failed [83, 85, 88]. However, greedy algorithms are computational more efficient, hence applicable to larger data set. To this end, HOMOTOPY [89] gives in some sense the best of both worlds by a single algorithm: speed where possible, sparsity where possible. One of the most important algorithm to solve underdetermined linear equations is LARS [65], which can be obtained from HOMOTOPY by simply removing teh sign constraint check. Also, LARS is greedy in nature which solves a linearly penalized least-square probem while less greedy compared with OMP. So same as HOMOTOPY, LARS algorithm is a good combination of speed and sparsity.

An intrinsic limitation in conventional CS theory is that it relies on data representation in vector forms. However, in modern applications, signals tends to have multidimensional structure. To list a few, 2D images, 3D images such as computed tomography (CT), magnetic resonance imaging (MRI) and hyperspectral images. In those cases, multidimensional signal or image is stored in memory as a tensor $\underline{\mathbf{Y}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, where tensor is the generalization of vectors and matrices to higher dimensions and can also be referred as N-way arrays or Multiway arrays. Due to curse of dimensionality problem, finding sparse representations of N-way arrays can be very expensive in memory storage and computation load due to the number of entries grows exponentially with the number of dimensions.

Therefore, the higher-order extension of CS theory for multidimensional data has become an e-merging topic. There are mainly two directions. One direction attempts to find the best rank-R tensor approximation as a recovery of the original data tensor such as [90], where the authers proved the ex-istence and uniqueness of the best rank-R tensor approximation in the case of 3rd order tensors under appropriate assumptions. In [91], the authors proposed multi-way CS (MWCS) for sparse low-rank tensors by a two-step recovery process: fitting a low-rank model in compression domain, followed by per-mode decompression, which is appealing in terms of computational complexity and memory capac-ity. However, the performance of MWCS is relied on the estimation of tensor rank, which is a NP-hard problem. The other direction [92] uses Kronecker product matrices in CS to act as sparsifying bases that jointly model the structure present in all of the signal dimensions as well as to represent the measurement protocols. However, the recovery procedure, due to the vectorization of multidimensional signals, more efficient algorithms are needed.

In [93], a generalized tensor compressed sensing model based on Tucker [94] and CANDECOMP/PARAFAC (CP) decomposition [95] is proposed as a unified framework of low-rank approximation and kronecker product for compressed sensing. The author demonstrated reduced computational complexity at reconstruction, however, compared with [92], the compression ratios are worse. In [96] [97] and [5], the authors proposed a generalized OMP [85] algorithm called Kronecker-OMP and Tensor-OMP to compute sparse and block-sparse of a tensor with respect to a Kronecker basis and demonstrated faster and more precise sparse representations of tensors compared with classical OMP. Although speedy, OMP failed in [83, 85, 88] due to its local optimal nature, where ℓ_1-minimization is more robust in finding the sparsest solution as it's convex and global optimal solutions are secured. Inspired by [5, 96, 97] and relationship between OMP, LARS, and HOMOTOPY algorithms [89], we generalized the classical LARS algorithm to efficiently compute both sparse and block-sparse of a tensor with respect to Kronecker basis in both ℓ_0-norm and ℓ_1-norm minimization. We demonstrate that the new algorithms – Kron-LARS and NBS-LARS – share the good properties of HOMOTOPY algorithm with fast speed and sparsest solution, we also show that those new algorithms outperform classical LARS algorithm in both memory storage and computational complexity in multidimensional signals.

This paper is organized as follows. In section II, we instroduce the basic notation, concepts and some important previous results. In section III, we describe our new Kron-LARS and NBS-LARS algorithms for the cases of multidimensional sparsity and multiway block sparsity. In section IV, we show the performance guarantee, memory consumption as well as computation complexity of Kron-

LARS and NBS-LARS. In section V, we present extensive simulations using synthetic data as well real-world multidimensional signals to demonstrate the extrodenary performance of proposed algorithms. In section VI, we conclude this work with main contributions and discussions.

3.3 Notation and Preliminaries

Throughout this discussion, we represent vectors by boldface lowercase letters, matrices (two-dimensional array) by bold uppercase letters and tensor (N-dimensional array) by underlined boldface capital letters. For example, $\mathbf{a} \in \mathbb{R}^{I}$, $\mathbf{A} \in \mathbb{R}^{I \times M}$ and $\underline{\mathbf{A}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ are vector, matrix and tensor respectively. The i-th entry of a vector is denoted by a_i, the element (i, j) of a matrix $\mathbf{A}$ is represented by $\mathbf{A}(i, j) = a_{i,j}$. We use same notation for N-dimensional array by $\|\underline{\mathbf{A}}\|_F = \sqrt{\sum_{i_1} \sum_{i_2} \cdots \sum_{i_N} a^2_{i_1 i_2 \cdots i_N}}$. For block sparsity representation, following [5] we use block (sub-array) notation by restricting the indices to belonging to certain subsets of indices. For example, given subsets of $\mathcal{S}_n$ indices $\mathcal{I}_n = \{i^1_n, i^2_n, ..., i^{S_n}_n\}$ in each mode $n = 1, 2, ..., N$, and the subarray $\underline{\mathbf{A}}(\mathcal{I}_1, \mathcal{I}_2, ..., \mathcal{I}_N) \in \mathbb{R}^{S_1 \times S_2 \times \cdots \times S_N}$ is obtained by keeping the entries of the original tensor $\underline{\mathbf{Y}}$ at the selected subsets of indices $\mathcal{I}_n (n = 1, 2, ..., N)$. The cardinality of a subset of indices $\mathcal{I}$ is denoted by $|\mathcal{I}|$.

3.3.1 Multilinear Algebra

Definition 1. (Kronecker product). Given two matrices $\mathbf{A} \in \mathbb{R}^{I\times M}$ and $\mathbf{B} \in \mathbb{R}^{J\times N}$, their Kronecker product $\mathbf{A} \otimes \mathbf{B} \in \mathbb{R}^{IJ\times MN}$ is defined by:

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & ... & a_{1M}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & ... & a_{2M}\mathbf{B} \\ . & . & . & . \\ a_{I1}\mathbf{B} & a_{I2}\mathbf{B} & ... & a_{IM}\mathbf{B} \end{pmatrix} \tag{3.1}$$

Properties 1. (Properties of Kronecker product)

$$\begin{aligned} (\mathbf{A} \otimes \mathbf{B})^T &= (\mathbf{A}^T \otimes \mathbf{B}^T) \\ (\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) &= (\mathbf{AC}) \otimes (\mathbf{BD}) \end{aligned} \tag{3.2}$$

Definition 2. (Khatri-Rao product). Given two matrices $\mathbf{A} \in \mathbb{R}^{I\times K}$ and $\mathbf{B} \in \mathbb{R}^{J\times K}$, their Khatri-Rao product $\mathbf{A} \odot \mathbf{B} \in \mathbb{R}^{IJ\times k}$ is defined by:

$$\mathbf{A} \odot \mathbf{B} = \begin{pmatrix} \mathbf{a}_1 \otimes \mathbf{b}_1 & \mathbf{a}_2 \otimes \mathbf{b}_2 & ... & \mathbf{a}_K \otimes \mathbf{b}_K \end{pmatrix} \tag{3.3}$$

Properties 2. (Properties of Khatri-Rao product)

$$(\mathbf{A} \odot \mathbf{B})^\dagger = ((\mathbf{A}^T\mathbf{A}) * (\mathbf{B}^T\mathbf{B}))^\dagger (\mathbf{A} \odot \mathbf{B})^T \tag{3.4}$$

Definition 3. (Tucker decomposition [98])

$$\underline{\mathbf{Y}} = \underline{\mathbf{G}} \times_1 \mathbf{A}_1 \times_2 \mathbf{A}_2 \cdots \times_N \mathbf{A}_N, \tag{3.5}$$

where $\underline{\mathbf{G}} \in \mathbb{R}^{R_1 \times R_2 \times \cdots \times R_n}$ is core tensor and $\mathbf{A}_n \in \mathbb{R}^{I_n \times R_n}$ are factor matrices.

Properties 3. (relationship between the Tucker model and a Kronecker representation of an N-way array) [5]. Given $\underline{\mathbf{Y}} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$, $\underline{\mathbf{X}} \in \mathbb{R}^{M_1 \times M_2 \cdots \times M_N}$, $\mathbf{B}_n \in \mathbb{R}^{I_n \times M_n}(n = 1, 2, ..., N)$, $\mathbf{x} = vec(\underline{\mathbf{X}})$ and $\mathbf{y} = vec(\underline{\mathbf{Y}})$, the following two representations are equivalent:

$$\underline{\mathbf{Y}} = \underline{\mathbf{X}} \times_1 \mathbf{B}_1 \times_2 \cdots \times_N \mathbf{B}_N, \tag{3.6}$$

$$\mathbf{y} = (\mathbf{B}_N \otimes \mathbf{B}_{N-1} \otimes \cdots \otimes \mathbf{B}_1)\mathbf{x}. \tag{3.7}$$

Definition 4. (CANDECOMP/PARAFAC (CP) Decomposition [95]). For a tensor $\underline{\mathbf{Y}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, the CP decomposition is to find R components that best approximates $\underline{\mathbf{Y}}$, i.e., to find

$$\min \|\underline{\mathbf{Y}} - \underline{\hat{\mathbf{Y}}}\|_2 \tag{3.8}$$

with $\underline{\hat{\mathbf{Y}}} = \sum_{r=1}^{R} \lambda_r \mathbf{A}_1(:, r) \circ \mathbf{A}_2(:, r) \circ \cdots \circ \mathbf{A}_N(:, r)$. where λ_r is a scalar, $\mathbf{A}_n(:, r)$ is the r column vector of matrix $\mathbf{A}_n$, $(n = 1, 2, ..., N)$.

Definition 5. (Multiway block-sparsity) [5]. A multidimensional signal (N-way array) $\underline{\mathbf{Y}} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$ is $(S_1, S_2, ..., S_N)$-block sparse with respect to the factors $\mathbf{B}_n \in \mathbb{R}^{I_n \times M_n}(n = 1, 2, ..., N)$ if it admits a Tucker representation based only on few S_n selected columns of each factor $S_n \leq M_n$, i.e. if $\mathcal{I}_n = [i_n^1, i_n^2, ..., i_n^{S_n}]$ denote a subset of indices for mode $n (n = 1, 2, ..., N)$, then

$$\underline{\mathbf{Y}} = \underline{\mathbf{X}} \times_1 \mathbf{B}_1 \times_2 \mathbf{B}_2 \times_3 \cdots \times_N \mathbf{B}_N, \tag{3.9}$$

with $x_{i_1 i_2 \cdots i_N} = 0 \, \forall (i_1, i_2, ..., i_N) \notin \mathcal{I}_1 \times \mathcal{I}_2 \times \cdots \times \mathbf{B}_N$,

Typically, we assume that $S_n \ll M_n$ and $M_n \geq I_n$. In other words, multiway block-sparsity assumes that the non-zero entries of the core tensor $\underline{\mathbf{X}}$ are located within a subarray (block) defined by $\underline{\mathbf{X}}(\mathcal{I}_1, \mathcal{I}_2, ..., \mathcal{I}_N)$.

Properties 4. (Multiway Block Sparsity implies Sparsity of the vectorized version of the signal) [5]. If an N-way array $\underline{\mathbf{Y}} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$ is $(S_1, S_2, ..., S_N)$-block sparse with respect to factor matrices $\mathbf{B}_n \in \mathbb{R}^{I_n \times M_n}$ $(n = 1, 2, ..., N)$ then its vectorized version $\mathbf{y} = vec(\underline{\mathbf{Y}}) \in \mathbb{R}^{I_1 I_2 \cdots I_N}$ is K-sparse $(K = S_1 S_2 \cdots S_N)$ with respect to the Kronecker dictionary $\mathbf{B} = \mathbf{B}_N \otimes \mathbf{B}_{N-1} \otimes \cdots \otimes \mathbf{B}_1$.

Proof 1. If we use the equivalence of equation and the definition of Multiway Block Sparsity we conclude that the vector of coefficients $\mathbf{x} = vec(\underline{\mathbf{X}})$ has at most $K = S_1 S_2 \cdots S_N$ none-zero entries which means that $\mathbf{y}$ has a K-sparse representation on the dictionary $\mathbf{B}_N \otimes \mathbf{B}_{N-1} \otimes \cdots \otimes \mathbf{B}_1$.

3.3.2 Compressed Sensing and Sparse Solutions of Underdetermined Linear Systems.

Compressed sensing is a framework for reconstruction of signals that have sparse representations. A vector $\mathbf{x} \in \mathbb{R}^M$ is called K-sparse if it has at most K nonzero entries. The CS measurement protocol measures the signal $\mathbf{x}$ with the measurement matrix $\mathbf{B} \in \mathbb{R}^{I \times M}$ where $I < M$ and information is encoded as $\mathbf{y} \in \mathbb{R}^I$ where $\mathbf{y} = \mathbf{Bx}$. The core problem becomes recovery of $\mathbf{x}$ from $\mathbf{y}$ given measurement matrix $\mathbf{B}$. Since $I < M$, this is a underdetermined linear system which has infinitely many solutions.

However, if $\mathbf{x}$ is known to be sparse enough, almost perfect recovery of $\mathbf{x}$ is possible both theoretically and practically, which establishes the fundamental tenet of CS thoery.

The recovery is achieved by finding a solution $\hat{\mathbf{u}}$ satisfying either ℓ_0 minimization or ℓ_1 minimizatoin as Eq. (Equation 3.10) and (Equation 3.11) respectively.

$$\hat{\mathbf{u}} = \arg\min\{\|\mathbf{u}\|_0,\ \mathbf{Bu} = \mathbf{y}\} \tag{3.10}$$

$$\hat{\mathbf{u}} = \arg\min\{\|\mathbf{u}\|_1,\ \mathbf{Bu} = \mathbf{y}\} \tag{3.11}$$

Such $\hat{\mathbf{u}}$ coincides with original signal $\mathbf{x}$ under certain condition. To be able to recover the $\mathbf{x}$ uniquely, dictionary matrix $\mathbf{B}$ needs to satisfy Eq. (Equation 3.12) [85]

$$K < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{B})}\right) \tag{3.12}$$

where $2K$ is the maximum number of columns of $\mathbf{B}$ that are linearly dependent, $\mu(\mathbf{B}) = \max_{i \neq j} |\mathbf{b}_i^T \mathbf{b}_j|$ is the *coherence* of two columns of dictionary, and dictionary matrix $\mathbf{B}$ has unit-norm columns ($\|\mathbf{b}_i\|_2^2 = 1,\ i = 1, 2,\ ..., M$).

Typically, when Eq. (Equation 3.10) is satisfied, we can find a unique sparsest solution with ℓ_0 minimization. However, solving ℓ_0 minimization such as Eq. (Equation 3.10) is NP-hard (requires combinatorial optimzation) and generally computational infeasible. Most research work are focusing on its convex relaxation i.e. ℓ_1 minimization, which is more tractable. Additionally, when the answer to Eq. (Equation 3.10) is sparse, it can be the same answer to Eq. (Equation 3.11). One well known result states that each K-sparse signal can be recovered uniquely if $\mathbf{B}$ satisfies the null space property of order K [81], denoted as NSP_K. That is, if $\mathbf{Bw} = \mathbf{0}$, $\mathbf{w} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$, then for any subset $S \subset \{1, 2, ..., N\}$ with cardinality $|S| = K$ holds that $\|\mathbf{v}_S\|_1 < \|\mathbf{v}_{S^c}\|_1$, where $\mathbf{v}_S$ represents the vector that coincides with $\mathbf{v}$ on the index set S and is set to zero on S^c. Another result states that if the observation $\mathbf{y}$ is noisy, for a given integer K, the matrix $\mathbf{B} \in \mathbb{R}^{I\times N}$ satisfies the restricted isometry property (RIP_K) [87] is if

$$(1-\delta_K)\|\mathbf{x}\|_2 \leq \|\mathbf{Bx}\|_2 \leq (1+\delta_K)\|\mathbf{x}\|_2 \tag{3.13}$$

for all K-sparse $\mathbf{x} \in \mathbb{R}^M$ and for some $\delta_K \in (0, 1)$.

3.3.3 Multidimensional and Block Sparsity Compressed Sensing

In this paper, the generalization of CS theory to higher dimensions is based on assumption of Kronecker structure of the dictionary and sensing matrix which was proposed in [79] for 2D signals and [92] for the general N-dimensional signal cases. The vectorization of a N-dimension tensor $\underline{\mathbf{U}} \in \mathbb{R}^{M_1\times M_2\times\cdots\times M_N}$ with K-sparse representation on a Kronecker basis can be expressed as Eq. (Equation 3.14).

$$\mathbf{u} = (\mathbf{D}_N \otimes \mathbf{D}_{N-1} \cdots \otimes \mathbf{D}_1)\mathbf{x}, \text{ with } \|\mathbf{x}\|_0 \leq K. \tag{3.14}$$

Then, with Kronecker structure sensing matrix $\mathbf{\Psi} = \mathbf{\Psi}_N \otimes \mathbf{\Psi}_{N-1} \otimes \cdots \otimes \mathbf{\Psi}_1$, we get a large-scale underdetermined linear system of equations as Eq. (Equation 3.7), where $\mathbf{B}_n = \mathbf{\Psi}_n\mathbf{D}_n$ $(n = 1, 2, ..., N)$.

Another useful finding of real world signals is that the nonzero coefficients are not evenly distributed which can be well modeled by block sparsity. In [99], algorithms and theories are provided for efficient recovery of vector signal with block sparse assumption. In [5], Caiafa *et al.* proposed algorithms and performance guarantees by further assuming the N-way block sparsity of multidimensional signals, which results in much less memory storage and computation consumption. Based on this work, we generalize the classical vector LARS as Kron-LARS and NBS-LARS to fit the multidimensional sparsity and block sparisty structure with respect to Kronecker bases. We show that the generalized LARS agorithms outperforms classical LARS from both memory usage and computational complexity perspective. Additionally, they inherit the good properties of HOMOTOPY algorithm which is a combination of speed and sparsity.

3.4 Kronecker LARS and N-way Block Sparse LARS

In this section, we first present the classical LARS algorithm, and its relation to HOMOTOPY [89] algorithm and greedy algorithm such as OMP [85], then we present our novel Kronecker LARS and N-way Block Sparse LARS algorithm by exploiting signal's multidimensional and block sparse structure with Kronecker dictionary.

In following subsections, we compute sparse representation of vector $\mathbf{y}$ with fixed dictionary $\mathbf{B}$ where $\mathbf{B} \in \mathbb{R}^{I \times M}$. For a multidimensional signal (tensor) $\underline{\mathbf{Y}} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$ we compute its sparse representation with a fixed dictionary $\mathbf{B} = \mathbf{B}_N \otimes \mathbf{B}_{N-1} \otimes \cdots \otimes \mathbf{B}_1$, where $\mathbf{B}_n \in \mathbb{R}^{I_n \times M_n}$.

3.4.1 Least Angle Regression (LARS)

LARS [100] is a regression method that has a more gentle version of forward selection and is originally proposed to estimate a sparse coefficient vector in a noisy over-determined linear system. LARS

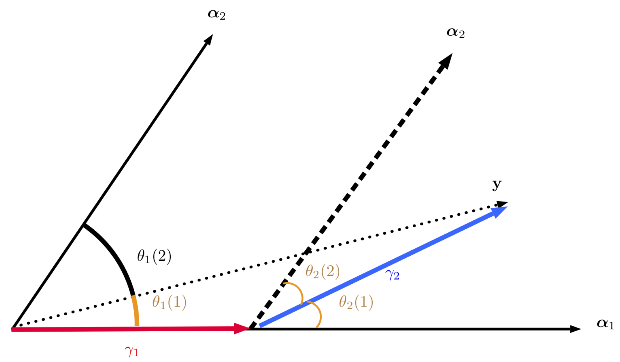


Figure 15. LARS ILLUSTRATION
vector $\boldsymbol{\alpha}_1$ has the largest correlation with **y** ($\theta_1(1) < \theta_1(2)$), instead of using the least squares solution (project **y** on to $\boldsymbol{\alpha}_1$), LARS algorithm move along the direction $\boldsymbol{\alpha}_1$ to a point where $\boldsymbol{\alpha}_2$ has same correlation with residual vector as $\boldsymbol{\alpha}_1$, i.e. $\theta_2(1) = \theta_2(2)$.

outputs estimates for all shrinkage and constraint parameters (HOMOTOPY) and is well suited for Basis Pursuit (BP) [83] in real case.

3.4.1.1 Relation to HOMOTOPY algorithm for solving ℓ_1 minimization

Generally, to solve ℓ_1 minimization such as Eq. (Equation 3.11), one could use simplex algorithm, basis pursuit [83] or interior-point method [101], which starting with a dense solution and converges to solutions of Eq. (Equation 3.11) in a sequence of iterations, each iteration with the solution of entire linear system. Where, in contrast, the HOMOTOPY method starts at 0 element, and building up a sparse solution by adding or removing elements from an active set. It's easy to see that with a sparse assumption, HOMOTOPY method is more efficient by reaching solution in few steps.

LARS is an alternative version of the HOMOTOPY algorithm from geometric perspective. It's derived from stagewise regression methods in statistics. Similar to HOMOTOPY algorithm, the LARS

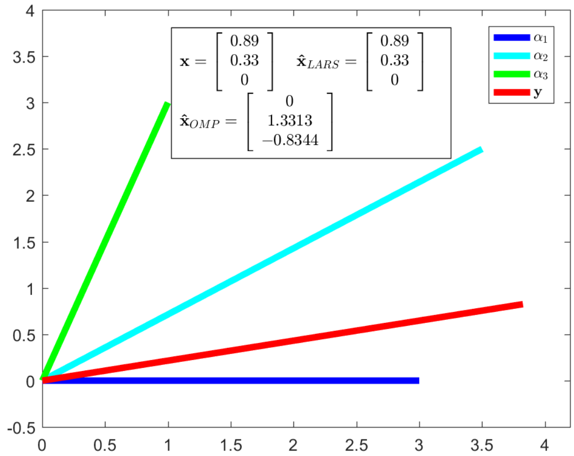


Figure 16. LARS VS OMP ON HIGHLY CORRELATED COLUMNS
LARS and OMP recovery of vector $\mathbf{y}$ with highly correlated column vectors $\{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\alpha}_3\}$

algorithm keeps an active set of nonzero elements composing the current solution. Then, for each step, the algorithm adds a new element to active set by taking a step along an equiangular direction, which is, a direction that has equal angles with the new element and vectors in the active set, such as in Fig. (Figure 15) .

Specifically, HOMOTOPY algorithm is very similar to LARS except that when adding a new element, HOMOTOPY algorithm may remove existing elements from active set. Thus, LARS with LASSO [60] modification as a variant of LARS is proposed, which is identical to the HOMOTOPY algorithm.

3.4.1.2 Relation to OMP algorithm

Conceptually, LARS algorithm is very similar to OMP as greedy algorithms. The difference can be easily summarized as follows: instead of taking a step size that yields least squares solution in each step, we shorten the step length and stop when a inacitve element has same correlation with the residual vector as the active element. We added the corresponding element from inactive set to active set and calculated a new direction. As all active elements are uncorrelated with the residual vector at least squares solution, we will therefore always get shorter step length for next active candidate compared with least squares solution.

In [100], experimental results show that the solution from LARS algorithm is often identical to the LASSO [60] solution. This equality is very interesting as LARS is so similar to OMP as a greedy algorithm. Both algorithm starts with an empty active set, each step adding a new element to the active set and ensuring that new element is most important among other candidates. However, at each step, LARS is finding the updating direction which has equal angles between new candidate and active set. This makes LARS less greedy and more global. Figure 16 is an example of vector $\mathbf{y}$ represented by 3 highly correlated vectors, where $\mathbf{x}$ is the mixing vector, $\hat{\mathbf{x}}_{\mathsf{LARS}}$ and $\hat{\mathbf{x}}_{\mathsf{OMP}}$ are recovered mixing vectors by LARS and OMP algorithms. In this case, OMP failed because of its greedy steps – residual is orthogonal to selected vector.

LARS algorithm's property can be summarized as follows: an algorithm for ℓ_0 minimization runs just as fast as OMP; an algorithm conceptually very similar to OMP but can be as effective as ℓ_1 minimization.

3.4.1.3 LARS algorithm basic derivation

LARS starts with an empty set of active variables, variable with the highest correlation with response is added to the model as first variable. We then use the least squares solution with this active variable as the first direction. Moving along this direction, we measure the angles between the ative variables and the residual vectors. The angles between them will chnage as it moves along the direction, and the correlation between the the active variable and residual vectors will shrink linearly towards 0. Before getting to the stage of 0 correlation, there may another variable that has the same correlation with respect to the residual vector as with the active variable. The new variable from inactive vector set is added to the active set and the move stops. We then get a new direction of moving towards the least squares solution of those two active variables. The move and new direction go on as above steps. After p steps, the full least squares solution will be reached.

The LARS algorithm is very efficient as we can get closed form solution for the step length at each iteration step. Denoting the model estimate of $\mathbf{y}$ at iteration k by $\hat{\mathbf{y}}^{(k)}$ and the least squares solution including the newly added active variable $\mathbf{y}_{OLS}^{\widehat{k+1}}$, the walk from $\hat{\mathbf{y}}^{(k)}$ towards $\hat{\mathbf{y}}_{OLS}^{(k+1)}$ can be formulated $(1-\gamma)\hat{\mathbf{y}}^{(k)}+\gamma\hat{\mathbf{y}}_{OLS}^{(k+1)}$ where $0 \leq \gamma \leq 1$. Estimating $\hat{\mathbf{y}}^{(k+1)}$, the position where the next active variable is to be added, then amounts to estimating γ. We seek the smallest positive γ where correlations become equal, that is

$$\mathbf{b}_{i\in\mathcal{I}}^{T}(\mathbf{y}-(1-\gamma)\hat{\mathbf{y}}^{(k)}-\gamma\hat{\mathbf{y}}_{OLS}^{(k+1)}) = \mathbf{b}_{j\in\mathcal{A}}^{T}(\mathbf{y}-(1-\gamma)\hat{\mathbf{y}}^{(k)}-\gamma\hat{\mathbf{y}}_{OLS}^{(k+1)}) \tag{3.15}$$

Solving this expression for γ, we get

$$\gamma_{i\in\mathcal{I}} = \frac{(\mathbf{b}_i - \mathbf{b}_j)^T(\mathbf{y} - \hat{\mathbf{y}}^{(k)})}{(\mathbf{b}_i - \mathbf{b}_j)^T(\hat{\mathbf{y}}_{OLS}^{(k+1)} - \hat{\mathbf{y}}^{(k)})} = \frac{(\mathbf{b}_i - \mathbf{b}_j)^T\mathbf{r}}{(\mathbf{b}_i - \mathbf{b}_j)^T\mathbf{d}} \tag{3.16}$$

where $\mathbf{d} = \hat{\mathbf{y}}_{OLS}^{(k+1)} - \hat{\mathbf{y}}^{(k)}$ is the direction of the walk, and $j \in \mathcal{A}$. Now, $\mathbf{b}$ is the orthogonal projection of $\mathbf{r}$ onto the plane spanned by the variables in $\mathcal{A}$, therefore we have $\mathbf{b}_j^T\mathbf{r} = \mathbf{b}_j^T\mathbf{b} = c$, representing the angle at the current breakpoint $\hat{\mathbf{y}}^{(k)}$. Furthermore, the sign of the correlation between variables is irrelevant. Therefore, we have

$$\gamma = \min_{i\in\mathcal{I}} \left\{ \frac{\mathbf{b}_i^T\mathbf{r} - c}{\mathbf{b}_i^T\mathbf{d} - c}, \frac{\mathbf{b}_i^T\mathbf{r} + c}{\mathbf{b}_i^T\mathbf{d} + c} \right\}, \quad 0 < \gamma \leq 1, \tag{3.17}$$

where the two terms are for correlations/angles of equal and opposite sign respectively. The coefficients at this next step are given by

$$\alpha^{(k+1)} = (1-\gamma)\alpha^{(k)} + \gamma\alpha_{OLS}^{(k+1)}. \tag{3.18}$$

Given these key pieces of the LARS algorithm, we state the entire procedure in Alg. (2)

black

3.4.2 Kronecker LARS (Kron-LARS) Algorithm : generalized LARS algorithm with Kronecker dictionary

Popular algorithms such as Basis Pursuit and OMP with Kronecker dictionary have been exploited in [102] and [96], however, they are either very slow as general convex optimization algorithm or greedy algorithm that will not always finds the sparsest solution. Here, we proposed Kron-LARS algorithm

Algorithm 2 Least Angle Regression [100]

Require: Given over-complete dictionary with $\mathbf{B} \in \mathbb{R}^{I \times M}$, signal $\mathbf{y} \in \mathbb{R}^{I}$, tolerance ϵ

Ensure: Sparse representation $\mathbf{y} = \mathbf{B}\mathbf{x}$ with $||\mathbf{x}||_0 \leq K$ ($||\mathbf{x}||_0$ represent number of nonzero elements.)

1: Initialize the coefficient vector $\boldsymbol{\alpha}^{(0)} = \mathbf{0}$ and the fitted vector $vec(\hat{\mathbf{y}}^{(0)}) = \mathbf{0}$,
2: Initialize the active set $\mathcal{A} = \emptyset$ and inactive set $\mathcal{I} = \{1...p\}$
3: **for** $k = 0$ **to** $p - 2$ **do**
4: Update the residual $\mathbf{r} = \mathbf{y} - \hat{\mathbf{y}}^{(k)}$
5: Find the maximal correlation $c = \max_{i \in \mathcal{I}} |\mathbf{b}_i^T \mathbf{r}|$
6: Move variable i corresponding to c from $\mathcal{I}$ to $\mathcal{A}$, $\mathcal{A} = \mathcal{A} \bigcup \{i\}$
7: Calculate the least square solution $\alpha_{OLS}^{(k+1)} = (\mathbf{B}_{\mathcal{A}}^T \mathbf{B}_{\mathcal{A}})^{-1} \mathbf{B}_{\mathcal{A}}^T \mathbf{y}$
8: Calculate the current direction $\mathbf{d} = \mathbf{B}_{\mathcal{A}} \alpha_{OLS}^{(k+1)} - \hat{\mathbf{y}}^{(k)}$
9: Calculate the step length $\gamma = \min^{+}_{i \in \mathcal{I}} \left\{ \dfrac{\mathbf{b}_i^T \mathbf{r} - c}{\mathbf{b}_i^T \mathbf{d} - c}, \dfrac{\mathbf{b}_i^T \mathbf{r} + c}{\mathbf{b}_i^T \mathbf{d} + c} \right\}, 0 < \gamma \leq 1$
10: Update regression coefficients $\alpha^{(k+1)} = (1 - \gamma)\alpha^{(k)} + \gamma \alpha_{OLS}^{(k+1)}$
11: Update the fitted vector $\hat{\mathbf{y}}^{(k+1)} = \hat{\mathbf{y}}^{(k)} + \gamma \mathbf{d}$
12: **end for**
13: Let $\alpha^{(p)}$ be the full least square solution $\alpha^{(p)} = (\mathbf{B}^T \mathbf{B})^{-1} \mathbf{B}^T \mathbf{y}$
14: Output the series of coefficients $\mathbf{A} = [\alpha^{(0)} ... \alpha^{(p)}]$

(Alg. 3) as a generalization of classical LARS or HOMOTOPY algorithm to keep the speed and sparsity properties at the same time.

Here we exploit the Kronecker structure of the dictionary to avoid storage of extra large dictionary and saves memory. To be specific, instead of caculating vector residual and correlation between vectors as *Step 4, 5* in LARS (Alg. 2), we calculate tensor residual and multiway product as *Step 4, 5* shown in Kron-LARS (Alg. 3). Additionally, in *Step 5* of Kron-LARS (Alg. 3), we also get much less computation complexity compared with same *Step 5* of LARS (Alg. 2). Assuming that $I_1 = I_2 = \cdots = I_N = I$ and $M_1 = M_2 = \cdots = M_N = M$, note that, we are solving underdetermined system of equations, so the number of dictionary elements M is usually large than number of rows (I) at a scale of $M = nI$, $n = 2 \sim 10$. We achieve a complexity of order $\mathcal{O}((nI)^{N+1})$, which is smaller than the computational complexity of the same *Step 5* in LARS (Alg. 2) to vectorized signal (about $\mathcal{O}(n^{N+1}I^{nN})$ where usually $n = 2 \sim 10$).

Another memory and computation intensive task in LARS (Alg. 2) is *Step 7*, i.e. least squares (LS) problem, which will need to inverse a very large dictionary.

For a N-dimensional tenosr that has K-sparse representation with Kronecker dictionary can have an equivalent Tucker representation as shown in Property 3 with a sparse core tensor $\underline{\mathbf{X}}$ that has only K nonzero elements. If we define the location of the nonzero entries in $\underline{\mathbf{X}}$ by $(i_1^k, i_2^k, ..., i_N^k)$ with $k = 1, 2, ..., K$, then we can express the N-way array $\underline{\mathbf{Y}}$ as a weighted sum of K rank-1 N-way arrays as

$$\underline{\mathbf{Y}} = \sum_{k=1}^{K} x_{i_1^k i_2^k \cdots i_N^k} \mathbf{B}_1(:, i_1^k) \circ \mathbf{B}_2(:, i_2^k) \circ \cdots \circ \mathbf{B}_N(:, i_N^k) \tag{3.19}$$

where $\circ$ represent outer product between vectors. From CP decomposition in Definition 4 and Kruskal operator defined in [91] *chapter 5*, we can interpret Eq. (Equation 3.19) as the following expression:

$$\mathbf{y} = vec(\underline{\mathbf{Y}}) = (\mathbf{C}_N \odot \mathbf{C}_{N-1} \odot \cdots \odot \mathbf{C}_1)\mathbf{x} \tag{3.20}$$

where $\mathbf{C}_n \in \mathbb{R}^{I_n \times K}$, $\mathbf{C}_n(:, k) = \mathbf{B}_n(:, i_n^k)$, $n = 1, 2, ..., N$ and $k = 1, 2, ..., K$; $\odot$ represent Khatri-Rao product as defined in Definition 2.

Thus, in multidimensional signal, the least square solution, i.e. *Step 7* in LARS (Alg. 2) can be represented as

$$\hat{\mathbf{x}} = (\mathbf{C}_N \odot \mathbf{C}_{N-1} \odot \cdots \odot \mathbf{C}_1)^{\dagger}\mathbf{y}, \tag{3.21}$$

Based on properties of Khatri-Rao product as in Property 2, the large matrix inverse problem can be converted to as follows:

$$\hat{\mathbf{x}} = \mathbf{F}^{\dagger}(\mathbf{C}_N \odot \mathbf{C}_{N-1} \odot \cdots \odot \mathbf{C}_1)^T \mathbf{y}, \tag{3.22}$$

where $\mathbf{F} = (\mathbf{C}_N^T \mathbf{C}_N) * (\mathbf{C}_{N-1}^T \mathbf{C}_{N-1}) * \cdots * (\mathbf{C}_1^T \mathbf{C}_1)$, $\dagger$ represents the sudo inverse and $*$ represents the elementwise product.

As $\mathbf{C}_n^T \mathbf{C}_n$, $n = 1, 2, ..., N$ is sysmetric and positive definite, based on Schur complement inversion formula in [103], each step of the calculation complexity is only $\mathcal{O}(I^N)$, where $I = I_1 = I_2 \cdots = I_{N-1} = I_N$ and N represents number of dimensionals. Compared with LARS, solving least square with Kronecker dictionary save significant amount of memory and computation complexity. The implementation of LARS for N-dimensional tenosr using a Kronecker dictionary (Kron-LARS) is given in algorithm 3.

black

3.4.3 NBS-LARS Algorithm: an Algorithm to Find Multiway Block-Sparse Representations

In this subsection we exploit signal with not only multidimensional structure but also block sparsity (similar to the work in [5]) assumption and come up with a very efficient algorithm called N-dimensional block sparse LARS (NBS-LARS) (as present in Alg. 4). The goal is to find a $(S_1, S_2, ..., S_N)$-block sparse representation of an N-dimensional tensor with respect to the factors $\mathbf{B}_n \in \mathbb{R}^{I_n \times M_n}(n = 1, 2, ..., N)$. We further show that, as nonzero elements are located within a subarray of size $S_1 \times S_2 \times \cdots \times S_N$, they can be quickly identified with much fewer iterations as compared to both LARS and Kron-LARS presented.

Algorithm 3 Kron-LARS Algorithm

Require: mode-n dictionaries $\{\mathbf{B}_1, \mathbf{B}_2, \cdots, \mathbf{B}_N\}$ with $\mathbf{B}_n \in \mathbb{R}^{I_n \times M_n}(n = 1, 2, ..., N)$, signal $\underline{\mathbf{Y}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, tolerance ϵ

Ensure: Sparse representation $vec(\underline{\mathbf{Y}}) \approx (\mathbf{B}_N \otimes \mathbf{B}_{N-1} \otimes \cdots \otimes \mathbf{B}_1)vec(\underline{\mathbf{X}})$ with $\|vec(\underline{\mathbf{X}})\|_0 \leq K$ (with nonzero entries given by $a_k = x_{i_1^k, i_2^k, ..., i_N^k}$, $k = 1, 2, ..., K$)

Initialize the coefficient vector $\boldsymbol{\alpha}^{(0)} = \mathbf{0}$ and the fitted vector $vec(\hat{\underline{\mathbf{Y}}}^{(0)}) = \mathbf{0}$
Initialize the active set $\mathcal{A} = \emptyset$ and the inactive set $\mathcal{I} = \{1, ..., K\}$, $k = 0$.
while $k \leq K$ and $\|\mathbf{R}\|_F > \epsilon$ **do**
    Update the residual tensor $\underline{\mathbf{R}} = \underline{\mathbf{Y}} - \hat{\underline{\mathbf{Y}}}^{(k)}$
    Find the maximal correlation $c_{max} = \max_{[i_1 i_2 \ldots i_N]} |\underline{\mathbf{R}} \times_1 \mathbf{B}_1^T(:, i_1) \times_2 \cdots \times_N \mathbf{B}_N^T(:, i_N)|$; $c = \underline{\mathbf{R}} \times_1 \mathbf{B}_1^T(:, i_1) \times_2 \cdots \times_N \mathbf{B}_N^T(:, i_N)$;
    Move variable corresponding to c_{max} from $\mathcal{I}$ to $\mathcal{A}$. $\mathcal{A} = [\mathcal{A}, i_n^k](n = 1, 2, ..., N)$; $\mathbf{C}_n(:, k) = \mathbf{B}_n(:, i_n^k)$, $(n = 1, 2, ..., N)$
    Calculate the least square solution $\boldsymbol{\alpha}_{OLS}^{(k+1)} = \arg\min_{\boldsymbol{\alpha}} \|(\mathbf{C}_N \odot \mathbf{C}_{N-1} \odot \cdots \odot \mathbf{C}_1)\boldsymbol{\alpha} - vec(\underline{\mathbf{Y}})\|_F^2$;
    Calculate the current direction $\mathbf{d} = (\mathbf{C}_N \odot \mathbf{C}_{N-1} \odot \cdots \odot \mathbf{C}_1)\boldsymbol{\alpha}_{OLS}^{(k+1)} - vec(\hat{\underline{\mathbf{Y}}}^{(k)})$
    Calculate the step length $\gamma = \min_{i \in \mathcal{I}}^{+} \left\{ \frac{c - c_{max}}{c_d - c_{max}}, \frac{c + c_{max}}{c_d + c_{max}} \right\}, 0 < \gamma \leq 1$, where $\underline{\mathbf{D}_d} = reshape\{\mathbf{d}, (I_1, I_2, ..., I_N)\}$, $c_d = (\mathbf{C}_N(:, N) \odot \mathbf{C}_{N-1}(:, N-1) \cdots \odot \mathbf{C}_1(:, 1))^T\mathbf{d}$.
    Update the regression coefficients $\boldsymbol{\alpha}^{(k+1)} = (1 - \gamma)\boldsymbol{\alpha}^{(k)} + \gamma\boldsymbol{\alpha}_{OLS}^{(k+1)}$
    Update the fitted vector $vec(\hat{\underline{\mathbf{Y}}}^{(k+1)}) = vec(\hat{\underline{\mathbf{Y}}}^{(k)}) + \gamma\mathbf{d}$
    $k = k + 1$
end while
Output coefficient tensor $\underline{\mathbf{A}} = reshape\{\boldsymbol{\alpha}, (M_1, M_2, ..., M_N)\}$

Algorithm 4 NBS-LARS Algorithm

Require: mode-n dictionaries $\{\mathbf{B}_1, \mathbf{B}_2, \cdots, \mathbf{B}_N\}$ with $\mathbf{B}_n \in \mathbb{R}^{I_n \times M_n}$ $(n = 1, 2, ..., N)$, signal $\underline{\mathbf{Y}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, maximum number of non-zero entries K_{max}, tolerance ϵ

Ensure: Sparse representation $\underline{\mathbf{Y}} \approx \underline{\mathbf{X}} \times_1 \mathbf{B}_1 \times_2 \mathbf{B}_2 \times_3 \cdots \times_N \mathbf{B}_N$ with $x_{i_1 i_2 \cdots i_N} = 0, \forall (i_1, i_2, ..., i_N) \notin \mathcal{A}_1 \times \mathcal{A}_2 \times \cdots \times \mathcal{A}_N$ (with non-zero entries given by $\underline{\mathbf{X}}(\mathcal{A}_1, \mathcal{A}_2, ..., \mathcal{A}_N) = \underline{\mathbf{A}}$).

1: Initialize the coefficient vector $\boldsymbol{\alpha}^{(0)} = \mathbf{0}$ and the fitted tensor $\hat{\underline{\mathbf{Y}}}^{(0)} = \underline{\mathbf{0}}$
2: Initialize the active set $\mathcal{A}_n = [\emptyset](n = 1, 2, ..., N)$ and the inactive set $\mathcal{I} = [I_1, I_2, ..., I_N]$, $k = 0$.
3: **while** $|\mathcal{I}_1||\mathcal{I}_2| \cdots |\mathcal{I}_N| < K_{max}$ and $\|\mathbf{R}\|_F > \epsilon$ **do**
4: Update the residual $\underline{\mathbf{R}} = \underline{\mathbf{Y}} - \hat{\underline{\mathbf{Y}}}^{(k)}$
5: Find the maximal correlation $c_{max} = \max_{[i_1 i_2 ... i_N]} |\underline{\mathbf{R}} \times_1 \mathbf{B}_1^T(:, i_1) \times_2 \cdots \times_N \mathbf{B}_N^T(:, i_N)|$; $c = \underline{\mathbf{R}} \times_1 \mathbf{B}_1^T(:, i_1) \times_2 \cdots \times_N \mathbf{B}_N^T(:, i_N)$;
6: Move variable corresponding to c from $\mathcal{I}$ to $\mathcal{A}$. $\mathcal{A}_n = \mathcal{A}_n \cup [i_n^k]$; $\mathbf{H}_n(:, k) = \mathbf{B}_n(:, i_n^k)$; $(n = 1, 2, ..., N)$
7: Calculate the least square solution by Cholesky factorization of the Hermitian matrix: $\boldsymbol{\alpha}_{OLS}^{(k+1)} = \arg\min_{\alpha} \|(\mathbf{H}_N \otimes \mathbf{H}_{N-1} \otimes \cdots \otimes \mathbf{H}_1)\boldsymbol{\alpha} - vec(\underline{\mathbf{Y}})\|_F^2$;
8: Calculate the current direction $\underline{\mathbf{D}_d} = \underline{\mathbf{A}}_{OLS}^{(k+1)} \times_1 \mathbf{H}_1 \times_2 \cdots \times_N \mathbf{H}_N - \hat{\underline{\mathbf{Y}}}^{(k)}$
9: Calculate the step length $\gamma = \min^{+}_{i_1 i_2 \cdots i_N \in \mathcal{I}_1 \times \mathcal{I}_2 \times \cdots \times \mathcal{I}_N} \left\{ \frac{c - c_{max}}{c_d - c_{max}}, \frac{c + c_{max}}{c_d + c_{max}} \right\}, 0 < \gamma \leq 1$, where $c_d = \underline{\mathbf{D}_d} \times_1 \mathbf{H}_1^T(:, 1) \times_2 \cdots \times_N \mathbf{H}_N^T(:, N), i_1 i_2 \cdots i_N \in \mathcal{I}_1 \times \mathcal{I}_2 \times \cdots \times \mathcal{I}_N$
10: Update the regression coefficients $\underline{\mathbf{A}}^{(k+1)} = (1 - \gamma)\underline{\mathbf{A}}^{(k)} + \gamma \underline{\mathbf{A}}_{OLS}^{(k+1)}$
11: Update the fitted tensor $\hat{\underline{\mathbf{Y}}}^{(k+1)} = \hat{\underline{\mathbf{Y}}}^{(k)} + \gamma \mathbf{D}_d$
12: $k = k + 1$;
13: **end while**
14: Output coefficient tensor $\underline{\mathbf{A}}$

If we represent submatrices as $\mathbf{H}_n \in \mathbb{R}^{I_n \times S_n}$ by restricting the mode-n dictionary to columns indicated by indices $\mathcal{I}_n$, i.e. $\mathbf{H}_n = \mathbf{B}_n(:, \mathcal{I}_n)$, the approximation of singal by Tucker model can be represented as follows using Property 3.

$$\mathbf{y} \approx \hat{\mathbf{y}} = (\mathbf{H}_N \otimes \mathbf{H}_{N-1} \otimes \cdots \otimes \mathbf{H}_1)\mathbf{u} \tag{3.23}$$

where $\mathbf{u} \in \mathbb{R}^K$ ($K = \prod_{n=1}^{N} S_n$) is the vectorized version of the N-dimensional tensor consisting of only nonzero elements. Thus, we get the following least square problems:

$$\boldsymbol{\alpha} = \arg\min_{\mathbf{u}} \|(\mathbf{H}_N \otimes \mathbf{H}_{N-1} \otimes \cdots \otimes \mathbf{H}_1)\mathbf{u} - \mathbf{y}\|_2^2, \tag{3.24}$$

where $\mathbf{y} \in \mathbb{R}^{I_1 I_2 \cdots I_N}$ is the vectorized version of the N-dimensional tensor $\underline{\mathbf{Y}}$. It is also shown as *Step 7* in Alg. 4.

By defining $\mathbf{H} = \mathbf{H}_N \otimes \mathbf{H}_{N-1} \otimes \cdots \otimes \mathbf{H}_1$, we see that the solution of this problem is give by $\mathbf{a} = [\mathbf{H}^T\mathbf{H}]^{-1}\mathbf{H}\mathbf{y}$, which means that $[\mathbf{H}^T\mathbf{H}]\mathbf{a} = \mathbf{H}\mathbf{y}$. This allows us to write

$$\mathbf{H}_1^T\mathbf{H}_1\mathbf{A}_{(1)}\left(\mathbf{H}_N^T\mathbf{H}_N \otimes \cdots \otimes \mathbf{H}_2^T\mathbf{H}_2\right) = \mathbf{H}_1^T\mathbf{Y}_{(1)}(\mathbf{H}_N^T \otimes \cdots \otimes \mathbf{H}_2^T). \tag{3.25}$$

By denoting $\underline{\mathbf{Z}}^{(1)} = \underline{\mathbf{A}} \times_1 \mathbf{I} \times_2 \mathbf{H}_2^T\mathbf{H}_2 \cdots \times_N \mathbf{H}_N^T\mathbf{H}_N$ and $\underline{\mathbf{P}} = \underline{\mathbf{Y}} \times_1 \mathbf{H}_1^T \times_2 \cdots \times_N \mathbf{H}_N^T$, we have

$$\mathbf{H}_1^T\mathbf{H}_1(\mathbf{Z}^{(1)})_{(1)} = \mathbf{P}_{(1)}, \tag{3.26}$$

which can be solved for $(\mathsf{Z}^{(1)})_{(1)}$ efficiently by using a Cholesky factorization of the Hermitian matrix $\mathbf{H}_1^T\mathbf{H}_1$. As $\mathbf{H}_1^T\mathbf{H}_1$ is only $|\mathcal{I}_1| \times |\mathcal{I}_1|$, this is a small computation problem. Then, by using the solution $\underline{\mathbf{Z}}^{(1)}$ of the subproblem, we can write its mode-2 unfolded version as

$$\mathbf{H}_2^T\mathbf{H}_2\mathbf{A}_{(2)}(\mathbf{H}_N^T\mathbf{H}_N \otimes \cdots \otimes \mathbf{H}_3^T\mathbf{H}_3 \otimes \mathbf{I}) = (\mathbf{Z}^{(1)})_{(2)}, \tag{3.27}$$

where with $\underline{\mathbf{Z}}^{(2)} = \underline{\mathbf{A}} \times_1 \mathbf{I} \times_2 \mathbf{I} \times_3 \mathbf{H}_3^T\mathbf{H}_3 \cdots \times_N \mathbf{H}_N^T\mathbf{H}_N$, we can solve the following simple subproblem efficiently also using the Cholesky factorization of the Hermitian matrix $\mathbf{H}_2^T\mathbf{H}_2$:

$$\mathbf{H}_2^T\mathbf{H}_2(\mathbf{Z}^{(2)})_{(2)} = (\mathbf{Z}^{(1)})_{(2)}. \tag{3.28}$$

After N steps of iteratively applying this procedure, we finally get the desired matrix $\mathbf{A}_{(N)}$, which maps to the coefficients in a mode-N matrix format for selected indices in the current iteration. The NBS-LARS algorithm optimizes the not only memory storage but also iterations compared to the classisc LARS algorithm as the maximum number of iterations is $k_{max} \ll K = S_1S_2 \cdots S_N$, with K being the number of nonzero entries within the core tensor $\underline{\mathbf{X}}$.

3.5 Performance Guarantee, Memory Consumption and Computational Complexity Analysis

In this section, we first analyze the performance guarantee of classical LARS and our proposed Kron-LARS and NBS-LARS algorithms. Then we demonstrate the memory storage and computation efficiency of our Kron-LARS and NBS-LARS algorithms.

3.5.1 Algorithm performance guarantee

3.5.1.1 Performance guarantee of LARS

In [104], the authors addressed two question for a given dictionary $\mathbf{B} \in \mathbb{R}^{I \times M}$ and signal $\mathbf{y} \in \mathbb{R}^{I}$:

- Uniqueness: Under which conditions is a highly sparse representation necessarily the sparsest possible representation?
- Equivalence: Under which conditions is a highly sparse solution to the ℓ_0 minimization problem also necessarily the solution to the ℓ_1 minimization problem?

To met those conditions, matrix **B** needs to satisfy following constrains [104] [91]:

- Large spark: $\mathtt{spark}\ (\mathbf{B} > 2K)$, where the $\mathtt{spark}$ of a given matrix is the smallest number of columns that are linearly dependent.
- Low coherence: $\mu(\mathbf{B}) < 1/(2K-1)$, where the coherence is defined as the largest normalized absolute inner product between any two columns, i.e. $\mu(\mathbf{B}) = \max_{i \neq j} |\mathbf{b}_i^T \mathbf{b}_j|$

where K is the number of nonzero coefficients with respect to known dictionary **B**.

3.5.1.2 Performance guarantee of Kron-LARS

As for Kron-LARS, we can use Property 3 to build relation between tensor and vector format easily. Thus, we have

$$vec(\underline{\mathbf{Y}}) = \mathbf{B}\mathbf{x} \tag{3.29}$$

where $vec(\underline{\mathbf{Y}}) = \mathbf{y} \in \mathbb{R}^{I^N}$ (with $I = I_1 = \cdots = I_N$) and $\mathbf{B} = \mathbf{B}_N \otimes \mathbf{B}_{N-1} \otimes \cdots \otimes \mathbf{B}_1$. We prove that the sparsity guarantee of Kron-LARS is

$$K < \frac{1}{2}(1 + \frac{1}{\mu}) \tag{3.30}$$

where K is the number of nonzero coefficients with respect to dictionary $\mathbf{B}$ and $\mu = max\{\mu(\mathbf{B}_1), \mu(\mathbf{B}_2), ..., \mu(\mathbf{B}_N)\}$.

Proof 2. It's easy to see that, from performance guarantee of LARS and Eq. (Equation 3.12), we can get

$$K < \frac{1}{2}\big(1 + \frac{1}{\mu(\mathbf{B})}\big) \tag{3.31}$$

where 2K is the maximum number of columns of $\mathbf{B}$ that are linearly dependent, $\mu(\mathbf{B}) = \max_{i \neq j} |\mathbf{b}_i^T \mathbf{b}_j|$ is the *coherence* of two columns of dictionary, and dictionary matrix $\mathbf{B}$ has unit-norm columns ($\|\mathbf{b}_i\|_2^2 = 1$, $i = 1, 2, ..., M$). As signal is K sparse, from Eq. (Equation 3.19) and Eq. (Equation 3.20), we can get $\mathbf{b}_i = \mathbf{C}_N(:, i) \otimes \mathbf{C}_{N-1}(:, i) \otimes \cdots \otimes \mathbf{C}_1(:, i)$ and $\mathbf{b}_j = \mathbf{C}_N(:, j) \otimes \mathbf{C}_{N-1}(:, j) \otimes \cdots \otimes \mathbf{C}_1(:, j)$ Using Property 1 of Kronecker product, we can get

$$\mathbf{b}_i^T \mathbf{b}_j = (\mathbf{C}_N^T(:, i)\mathbf{C}_N(:, j)) \otimes \cdots \otimes (\mathbf{C}_1^T(:, i)\mathbf{C}_1(:, j)) \tag{3.32}$$

According **Theorem 3.5.** and **Corollary 3.6.** in [105], we can get

$$\max_{i \neq j} |\mathbf{b}_i^T \mathbf{b}_j| = \max_{1 \leq n \leq N} \max_{i \neq j} |\mathbf{C}_n^T(:, i)\mathbf{C}_n(:, j)| = \mu \tag{3.33}$$

Hence, $\mu(\mathbf{B}) = \mu$. And we can get the following sparsity guarantee

$$K < \frac{1}{2}(1 + \frac{1}{\mu}) \tag{3.34}$$

3.5.1.3 Performance guarantee of NBS-LARS

From the performance guarantee of N-way block sparse OMP algorithm in [5], the performance guarantee is same and summarized as follows:

Theorem 1. (NBS-LARS performance guarantee). Given the multiway decomposition $\underline{\mathbf{Y}} = \underline{\mathbf{X}} \times_1 \mathbf{B}_1 \times_2 \mathbf{B}_2 \cdots \times_N \mathbf{B}_N$, with a fixed N-way array $\underline{\mathbf{Y}} \in \mathbb{R}^{I \times I \times \cdots \times I}$ and known dictionaries $\mathbf{B}_n \in \mathbb{R}^{I \times M}$ with coherences $\mu_n (n = 1, 2, ..., N)$, if a $(S, S, ..., S)$-block sparse solution exists satisfying

$$(S\mu)^N < 2 - (1 + (S - 1)\mu)^N, \tag{3.35}$$

with $\mu = \max\{\mu_1, \mu_2, ..., \mu_N\}$. Then NBS-LARS (Alg. 4) is guaranteed to find this sparse representation in K iterations with $S \leq K \leq NS$, see proof in [5].

3.5.2 Memory usage and computational complexity

3.5.2.1 Memory usage

In Figure 17, the memory required to store the resulting explicit matrix $\mathbf{B}$ for 1D, 2D and 3D signals (tensors) for the case of $M_n = I_n/2 (N = 1, 2, 3)$ are shown. If we use 16GB as the limit memory size, the dictionary for a 2D signal with a size of 304×304 can be store and for the 3D case it corresponds to a tensor with size $51 \times 51 \times 51$.

TABLE VIII

COMPLEXITY ANALYSIS

Iterations and operations required at iteration k for standard LARS, Kronecker-LARS and NBS-LARS algorithms for an N-way array $\underline{\mathbf{Y}} \in \mathbb{R}^{I\times I \cdots \times I}$ with dictionaries $\mathbf{B}_n \in \mathbf{R}^{I\times M}$ and the block sparsity parameter S (usually $S \ll I$, $M \geq I$)

Complexity Analysis			
	LARS	Kronecker-LARS	NBS-LARS
# Iter.	S^N	S^N	$\leq NS$
STEP **5**	$2(MI)^N + 2M^N$	$2M^N I\left(\frac{1-(I/M)^N}{1-I/M}\right) + 2M^N$	$2M^N I\left(\frac{1-(I/M)^N}{1-I/M}\right) + 2M^N$
STEP **7**	$(kI)^N + 3k^{(2N)}$	$2I^N + 2Nk^N I + (N+4)k^N + 7k^{2N}$	$\leq 2Nk(I+1) + Nk^2 + 2Nk^{N+1}$
STEP **8**	$2(Ik)^N + I^N$	$N(N-1)I^N + I^N$	$2kI^N + I^N$
STEP **9**	$2(Ik)^N + 2I^N$	$N(N-1)I^N + 2I^N$	$2kI^N + 2I^N$
Asymptotical cost per iter.	$(IM)^N$	$I(M)^N$	$I(M)^N$
Total cost	$(SIM)^N$	$I(SM)^N$	$INS(M)^N$

Run-time memory usage analysis between LARS, Kronecker-LARS and NBS-LARS are listed in Table Table IX.

3.5.2.2 Computational cost

As LARS algorithm is greedy in nature, it starts from empty active set and added one variable from inactive set to active set at each iteration/step. As number of active variables grow from 0 to K gradually, the computation cost of each iteration/step is low, compared with classical Basis Pursuit (BP) algorithm which updates all coefficients in every iteration/step. As a result, LARS is preferred over BP specially when the number of nonzero coefficeints is small. Let us consider the case of the recovery of N-dimensional tensors from the measurements given by $\underline{\mathbf{W}} \in \mathbb{R}^{M\times M \cdots \times M}$ having a block sparse representation $\underline{\mathbf{W}} = \underline{\mathbf{X}} \times_1 \mathbf{B}_1 \times \mathbf{B}_2 \cdots \times_N \mathbf{B}_N$ with factors matrices $\mathbf{B}_n \in \mathbb{R}^{M\times I}(M < I)$ and a

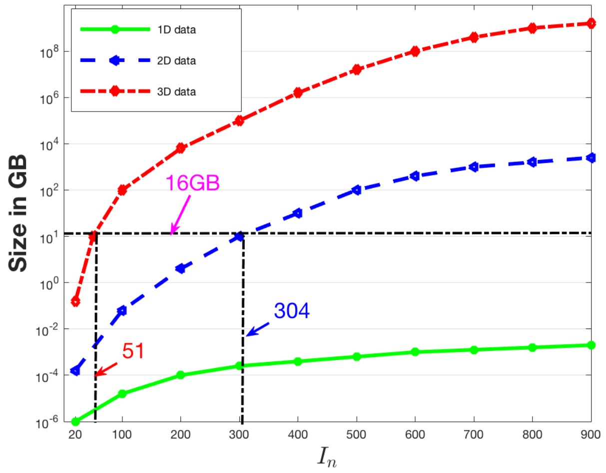


Figure 17. MEMORY USAGE

Memory usage on 1D, 2D and 3D data with mode size I_n

$(S, S. ..., S)$ block -sparse core tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I \times I \cdots \times I}$, i.e. with non-zero coefficients concentrated in a $S \times S \times \cdots S$ subtensor.

3.5.2.3 Complexity Analysis

Here we analyze the computational complexity of LARS, Kronecker-LARS and NBS-LARS. We assume an N-way array $\underline{\mathbf{Y}} \in \mathbb{R}^{I \times I \times \cdots \times I}$ and mode-n dictionaries given by matrices $\mathbf{B}_n \in \mathbb{R}^{I \times M}$. We also assume that an $(S, S, ..., S)$-block sparse representation of $\underline{\mathbf{Y}}$ with factors $\mathbf{B}_n (n = 1, 2, ..., N)$, which means that there are S^N nonzero coefficients. We consider the arithmetic operations required by step 5(maximum correlated atom detection), step 7 (least squares estimation of nonzero coefficients), and step 4 (residual update) in all algorithms at iteration number k in terms of I (mode size), M (number of atoms per mode), and N (number of dimensions). The comparative results are summarized in Table

TABLE IX

RUN-TIME MEMORY ANALYSIS

Iterations and operations required at iteration k for standard LARS, Kronecker-LARS and NBS-LARS algorithms for an N-way array $\underline{\mathbf{Y}} \in \mathbb{R}^{I\times I\times I\cdots\times I}$ with dictionaries $\mathbf{D}_n \in \mathbf{R}^{I\times M}$ and the block sparsity parameter S (usually $S \ll I$)

Run-time Memory Analysis			
	LARS	Kronecker-LARS	NBS-LARS
# Iter.	S^N	S^N	$\leq NS$
STEP k	$(IM)^N$	$I^N + NIM$	$I^N + NIM$

3.5.1.3 (see details in the appendix) where the advantage of NBS-LARS over Kronecker-LARS and LARS is evident. From this table, we observe that for very sparse representations with $S \ll I < M$, the complexity is dominated by step 5, which is exactly the same for Kronecker-LARS and NBS-LARS. The key advantage of NBS-LARS over the other algorithms is that it requires many fewer iterations ($\mathcal{O}(S)$ against $\mathcal{O}(S^N)$ iterations in LAR/Kronecker-LAR). In addition, in step 5, the NBS-LAR algorithm complexity in terms of the number of entries I^N is sublinear compared to a linear dependence of the standard LAR and the Kronecker-LAR algorithms. In section V, we show several numerical results with comparisons of the computation times required by different algorithms applied to multidimensional signals.

3.6 Exerimental Results

In this section, we present several simulation results on both synthetically generated signals and real-world signals in order to compare the performance of our proposed Kron-LARS and NBS-LARS

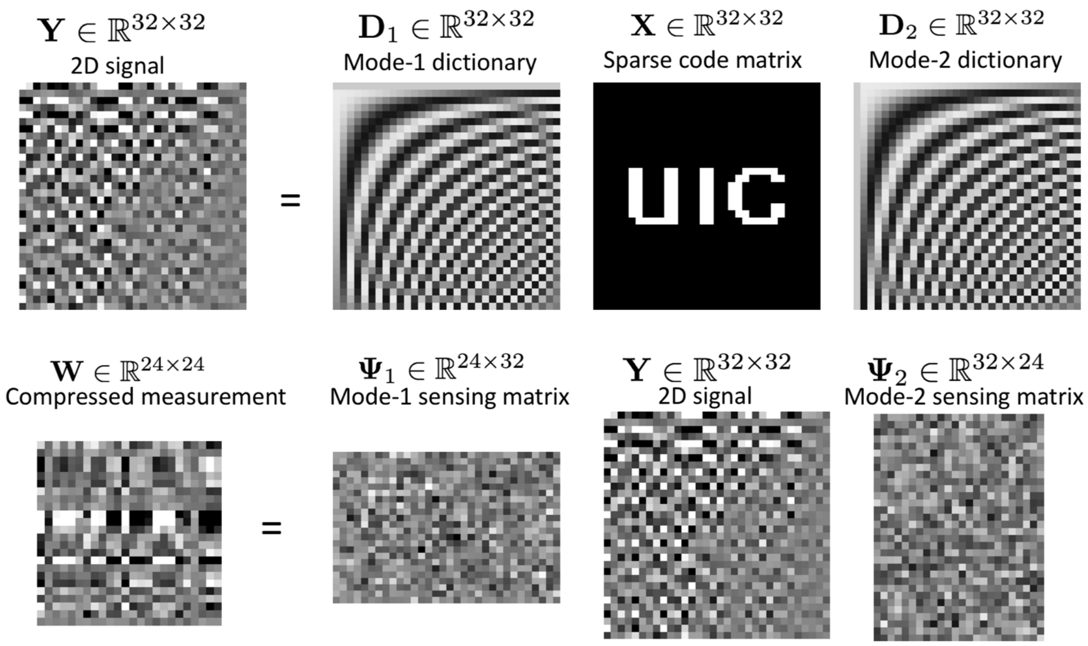


Figure 18. 2D COMPRESSED SENSING ON SYNTHETIC DATA

Sparse core matrix is $\mathbf{X} \in \mathbb{R}^{32\times32}$, DCT dictionaries of mode-1 and mode-2 are $\mathbf{D}_1 \in \mathbb{R}^{32\times32}$ and $\mathbf{D}_2 \in \mathbb{R}^{32\times32}$, sensing matrices of mode-1 and mode-2 are $\mathbf{\Psi}_1 \in \mathbb{R}^{32\times24}$ and $\mathbf{\Psi}_2 \in \mathbb{R}^{32\times24}$.

algorithms against similar greedy algorithms Kron-OMP and NBOMP proposed in [5] as well as classic algorithms such as LARS, OMP. All those algorithms and their best fitted data sparsity are listed in Table Table X.

We first apply proposed Kron-LARS and NBS-LARS algorithms to 2D synthetic data to show that our algorithms can recover the 2D signal exactly. Then, we compare our algorithm with classic OMP and LARS in real-world 2D images. Next, we apply algorithms on 3D signals, such as MRI and hyperspectral imaging. Finally, we run our algorithms on video clips and show its efficient and accurate performance.

3.6.1 2D synthetic data compressed sensing

3.6.1.1 2D unstructured synthetic data

we start with 2D unstructured synthetic data. Given 2D signal $\mathbf{Y}$, by using Tucker decomposition, we can get

$$\mathbf{Y} = \mathbf{X} \times_1 \mathbf{D}_1 \times_2 \mathbf{D}_2 \tag{3.36}$$

where $\mathbf{X}$ is sparse core matrix, $\mathbf{D}_1$ and $\mathbf{D}_2$ are mode 1 and mode 2 dictionary from DCT respectively. With compressed measurement matrix $\mathbf{W}$, mode 1 sensing matrix $\mathbf{\Psi}_1$ and mode 2 sensing matrix $\mathbf{\Psi}_2$, we have

$$\mathbf{W} = \mathbf{Y} \times_1 \mathbf{\Psi}_1 \times_2 \mathbf{\Psi}_2 \tag{3.37}$$

TABLE X

COMPUTING SPARSE REPRESENTATIONS OF MULTIDIMENSIONAL SIGNALS

Dictionary D	**Sparsity type**	**Algorithms**
Non-structured	Non-structured	classical OMP / LARS
Kronecker	Non-structured	Kron-LARS / Kron-OMP
Kronecker	N-dim Block-Sparsity	NBOMP, NBS-LARS

Combining Eq. (Equation 3.36) and Eq. (Equation 3.37), we get

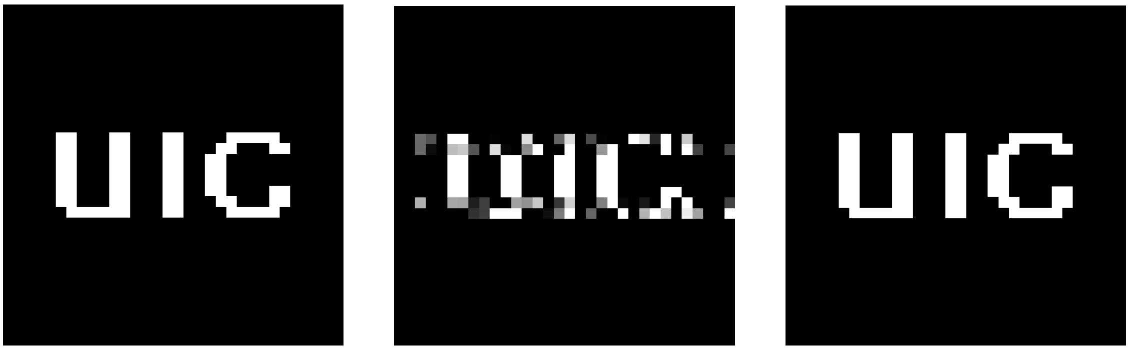
$$\mathbf{W} = \mathbf{X} \times_1 \mathbf{\Psi}_1\mathbf{D}_1 \times_2 \mathbf{\Psi}_2\mathbf{D}_2 \tag{3.38}$$

With compressed measurement signal $\mathbf{W}$, dictionary matrix $\mathbf{D}_n$ and sensing matrix $\mathbf{\Psi}_n$ $(n = 1, 2)$ given, we can restore the core sparse matrix $\mathbf{X}$ as shown in Figure 18 by applying Kron-LARS algorithm.

As Kron-OMP is the most similar algorithm to our proposed Kron-LARS algorithm in terms of both memory usage and computational complexity. We apply both algorithms to the generated synthetic data with 105 nonzero points out of 1024. This is far beyong the theoretical limit of sparsity for a perfect restoration, but it turns out both algorithms can restore the core matrix exactly at certain sampling ratio. However, it turns out Kron-LARS can restore the core matrix at lower sampling ratio which makes it more appealing. We also find to it may take more steps for Kron-LARS, but it's actually faster than Kron-OMP in terms of computation time. Figure 19 shows the core matrix restoration by both Kron-OMP and Kron-LARS at sampling ratio of 56%. Kron-LARS can restore the core matrix exactly well Kron-OMP restored in low PSNR. Table Table XI is a list of comparison results in terms of PSNR (dB), STEPS taken, and rum time (S) between Kron-OMP and Kron-LARS on different sampling ratios.

3.6.1.2 2D structured synthetic data

structured data is also happening very often in real life, such as circles, squares, rectangles, symmetric objects, etc. Here we created a 25 nonzero dots square matrix (out of 1024 dots) as in Figure 20 (a).



(a) (b) (c)

Figure 19. Kron-LARS vs Kron-OMP compressed sensing on synthetic data

(a) Sparse core matrix; (b) Kron-OMP algorithm [5] recovered core matrix; (c) **Proposed Kron-LARS** algorithm recovered core matrix. At sampling ratio of 56%, Kron-OMP recovered core matrix PSNR is 13.34 dB, proposed Kron-LARS recovered core matrix PSNR is 267.89 dB (almost exact).

TABLE XI

KRON-OMP VS KRON-LARS ON SYNTHETIC DATA RESTORATION

(PSNR (dB), STEPS and Run-Time vs Sampling Ratio (%))

%	Kron-OMP			Kron-LARS		
	PSNR	STEPS	Time (s)	PSNR	STEPS	Time (s)
14	9.78	144	6.11	9.40	142	0.20
19	9.83	196	10.52	10.82	194	0.38
25	9.47	256	17.74	11.79	254	1.03
32	11.61	324	27.60	13.66	253	0.88
39	11.04	400	43.82	15.15	279	1.29
47	11.28	484	66.08	23.69	293	1.71
56	13.34	145	6.86	Inf	184	0.72
66	15.10	134	6.27	Inf	166	0.63
77	Inf	84	3.14	Inf	119	0.36
87	Inf	97	3.99	Inf	132	0.48
100	Inf	102	4.39	Inf	142	0.75

Although Kron-OMP and Kron-LARS can restore the core matrix efficiently, but they all need at least 25 steps. Structure is the key here to reduce the computation complexity i.e. iteration steps. Our proposed NBS-LARS algorithm is an extension of the existing NBOMP algorithm where the core algorithm is inherited from LARS instead of OMP. Both NBOMP and NBS-LARS algorithms set number of nonzero constrain on each mode of the data matrix, in this case it's row and columns. Each step they will add active variable into both row active set and column active set. Thus, ideally, our 25 nonzero core matrix could be resored in 5 steps if the steps are going through the diagonal.

Figure 20 (b) and (c) are the core matrix restoration results and steps from NBOMP and NBS-LARS respectively. They take 8 and 9 steps respectively to restore the core matrix which are both less than 25 steps. It's easy to find out although NBS-LARS takes more steps in restoration, it's in a more tracktable pattern along the diagonal and potentially more stable and consistent in terms of high dimensional data compressed sensing.

Figure 21 shows tensor recovery percentage versus sample ratio over 50 simulations with ($32 \times 32 \times 32$) tenosr signals having ($5 \times 5 \times 5$) block sparse representation with a Kronecker DCT dictionary shown in Figure 22.

3.6.2 Real-world 2D image compressed sensing

Applying compressed sensing into real-world images is also called compressed imaging (CI) [106]. CI can save efforts in collecting large amount of pixel data and store them after compression by only collecting the non-redundant data in acquisition step.

To evaluate the performance of our algorithms in CI, we selected 3 widely used benchmark images as shown in Figure 23.

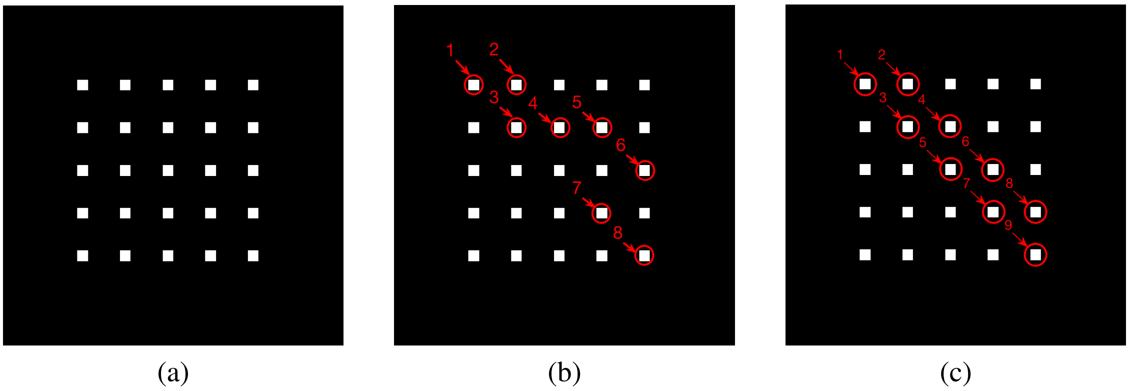


Figure 20. NBOMP [5] vs NBS-LARS in structured synthetic data restoration

(a) Sparse core matrix; (b) NBOMP algorithm [5] recovered core matrix; (c) **Proposed NBS-LARS** algorithm recovered core matrix. NBOMP and NBS-LARS only take 8 and 9 steps respectively to fully restore 25 nonzero points. Red circle with number indicate the step and activated nonzero point.

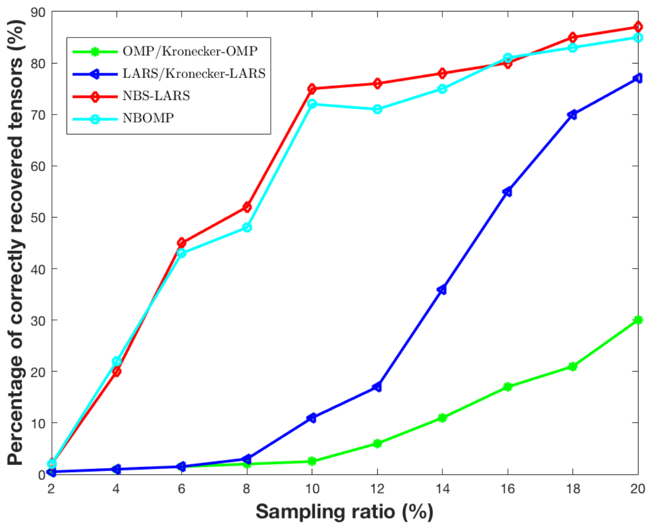


Figure 21. Recovery percentage versus sample ratio over 50 simulations with $(32 \times 32 \times 32)$ tensor signals having $(5 \times 5 \times 5)$ block sparse representation with a Kronecker DCT dictionary.

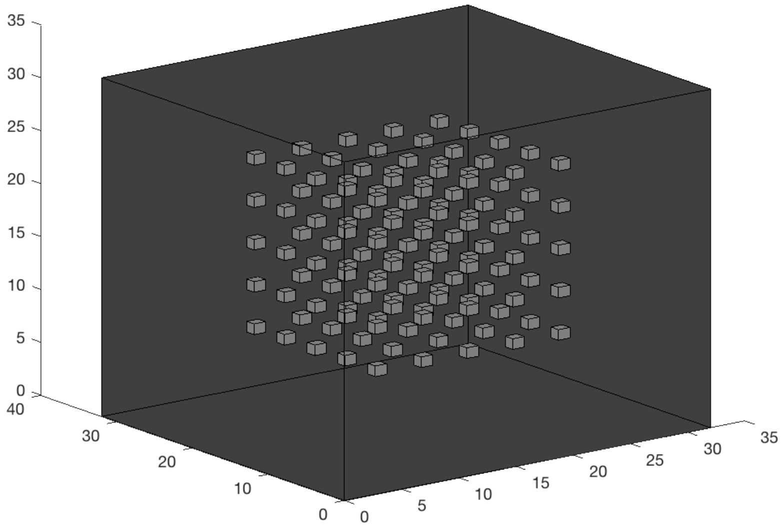


Figure 22. 3D tensor (32 × 32 × 32) having (5 × 5) block sparse representation.

Phantom Cameraman Lena

Figure 23. Three benchmark images for compressed sensing testing

Here, we compare the performance of our Kron-LARS and NBS-LARS against LARS, Kron-OMP and N-BOMP algorithms. We evaluate the average performance of each algorithm on 3 benchmark images with size from 8×8 to 512×512, by restricting the recovery PSNR to $\geq$ 50dB, we got computation time versus image size plot as in Figure 24. We can see that when image data size is relatively small, LARS algorithm is slightly faster than Kron-OMP and Kron-LARS as it stores the entire large dictionary in memory. However, when data size grow larger, the advantage of computation reduction of Kron-OMP and Kron-LARS is more significant than dictionary multiplication, thus, both of them outperform LARS. Another important thing to point out, as LARS stores the entire large dictionary in memory, it goes to more than 10 GB and freezes the program when image size is 128×128 and above.

Figure 25 are an example of restoration on phantom ($\mathbf{X} \in \mathbb{R}^{256 \times 256}$) by NBOMP and NBS-LARS algorithms. The recovery by NBS-LARS is almost exact while NBOMP is only at 20 dB in PSNR. In this example, we use sensing matrix without sampling.

3.6.3 3D Magnetic Resonance Imaging (MRI) and Hyperspectral image compressed sensing

Other important applications of compressed sensing in high dimensional signal are MRI and hyperspectral image compressed sensing, which is an extension of CI by stacking 2D image slices together as a 3D image. Similar to Eq. (Equation 3.38), by adding a mode 3 dictionary $\mathbf{D}_3 \in \mathbb{R}^{\mathrm{I}}$ and sensing matrix $\mathbf{\Psi}_3$, we can get the following expression

$$\mathbf{W} = \mathbf{X} \times_1 \mathbf{\Psi}_1\mathbf{D}_1 \times_2 \mathbf{\Psi}_2\mathbf{D}_2 \times_3 \mathbf{\Psi}_3\mathbf{D}_3 \tag{3.39}$$

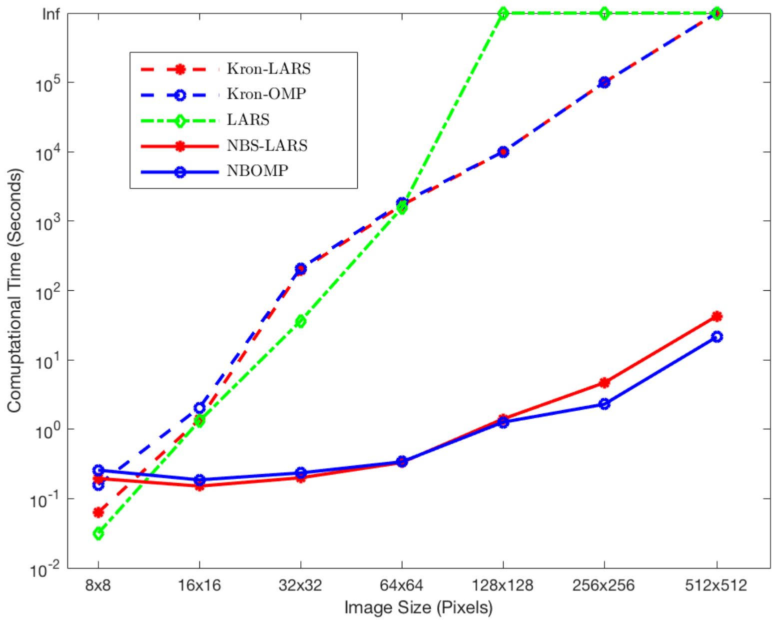


Figure 24. Computation Efficiency of Kron-LARS, NBS-LARS, Kron-OMP, N-BOMP, and LARS algorithms on 3 bench mark images. (PSNR$\geq$ 50dB).

3.6.3.1 CS Magnetic Resonance Imaging (MRI)

MRI and Computer Tomographic (CT) technologies have been major motivations of the development of CS theory since the publication of [79] that proves structured singals can be almost perfectly recovered from Fourier samples. CS MRI has becoming a mature technology during the last decade, which is a real application of CS theory that can reduce the data acquisition process significantly.

Despite the success of CS theory in MRI, computation efficient is still a major concern for high-dimensional and large dataset. To this end, Kronecker structure is a good fit of MRI to reduce memory storage and computation complexity.

NBOMP NBS-LARS

Figure 25. NBOMP and proposed NBS-LARS restoration of *phantom* ($\mathbf{X} \in \mathbb{R}^{256\times 256}$).

Figure 26 shows a 3D MRI brain image, 75% and 50% sampling ratio reconstruction by Kron-LARS. 75% sampling reconstruction PSNR is 44.65 dB and 50% sampling reconstruction PSNR is 37.24 dB.

3.6.3.2 CS hyper-spectral imaging

Another specific application of CS using Kronecker structure is Hyper-spectral compressed imaging (HCI) which stacks 2D images at several spectral bands to form a 3D signal. In real case, matrices $\mathbf{\Psi}_1$ and $\mathbf{\Psi}_2$ determine the separable sensing operator applied to each 2D slice and matrix $\mathbf{\Psi}_3$ is the identity matrix.

Figure 27 shows hyperspectral image compressed sensing at different slices (best and worst) by NBS-LARS algorithm. Hyperspectral image source is *sence 7* of data set in [6].

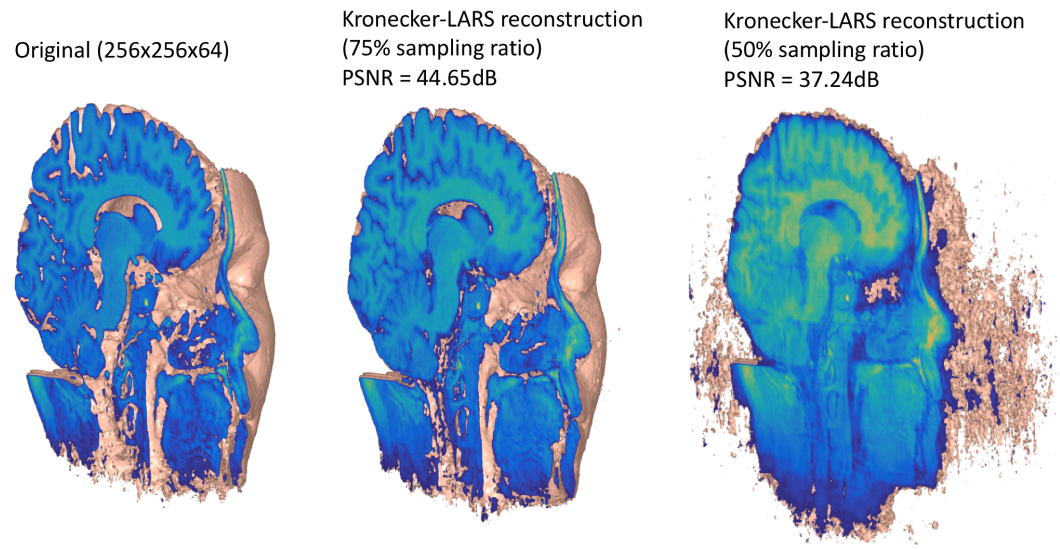


Figure 26. MRI reconstruction by Kron-LARS on 75% and 50% sampling.

3.7 Conclusion

Real world signals have intrinsic sparsity property, however, to inverse the original signal from a sparsity representation is a computational intensive process. More efficient algorithms and assumptions are needed for large datasets. As modern signals are often high-dimensional with sparisty structure, efficient algorithms that using signals' multidimensional structure become a emerging research area. In this paper, we extend the classical LARS algorithm to Kron-LARS and NBS-LARS by utilizing Kronecker structure as well as N-dimension block sparsity assumption. We demonstrate that the new algorithms have the good property of classical LARS – very fast and very sparse – and more efficient in memory usage and computational complexity when processing high-dimensional data. We apply

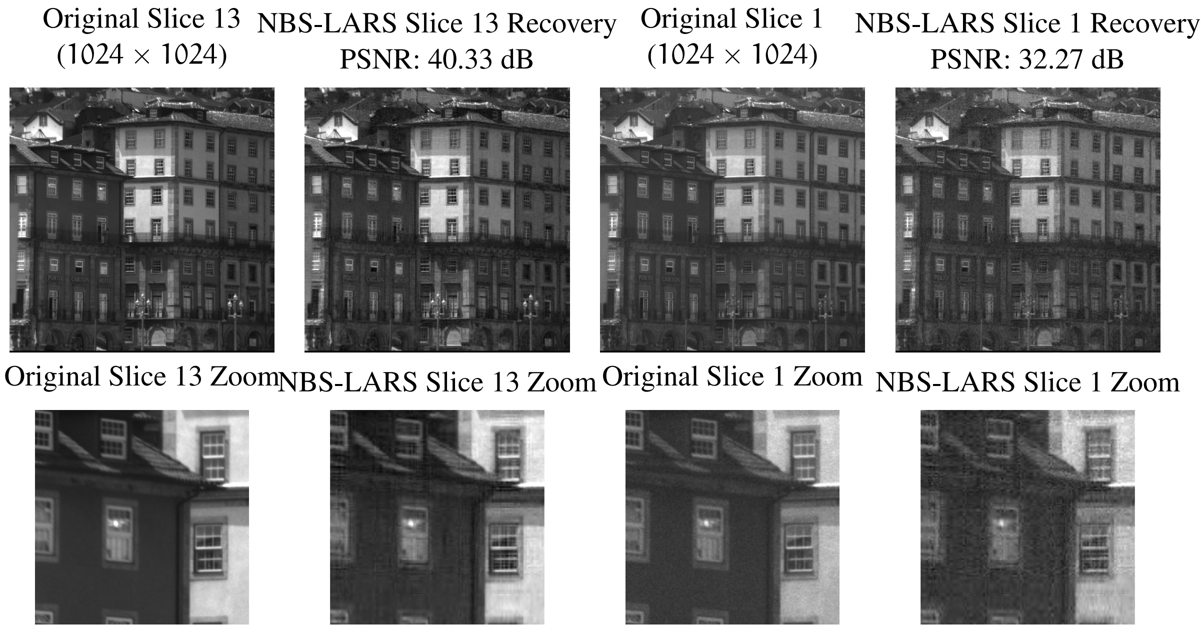


Figure 27. Hyperspectral image compressed sensing by NBS-LARS algorithm applied on *sence 7* of data set in [6].

our algorithms into 2D synthetic data, 2D real-world image, 3D MRI and 3D HSI compressed sensing. Simulation results show that our Kron-LARS performances better than LARS in terms of memory usage and computation time as the dimension increases. Additionally, when block-sparsity assumption is true, NBS-LARS algorithm can further reduce the memory and complexity dramatically. NBS-LARS can better recover signals based on sample ratio when block assumption is reasonable.

CHAPTER 4

COLOR NORMALIZATION OF HISTOLOGY SLIDES USING GRAPH REGULARIZED SPARSE NMF (PREVIOUSLY PUBLISHED AS LINGDAO SHA,DAN SCHONFELD,AMIT SETHI (2017) COLOR NORMALIZATION OF HISTOLOGY SLIDES USING GRAPH REGULARIZED SPARSE NMF, PROC.SPIE, 10140 - 10140 - 11.)

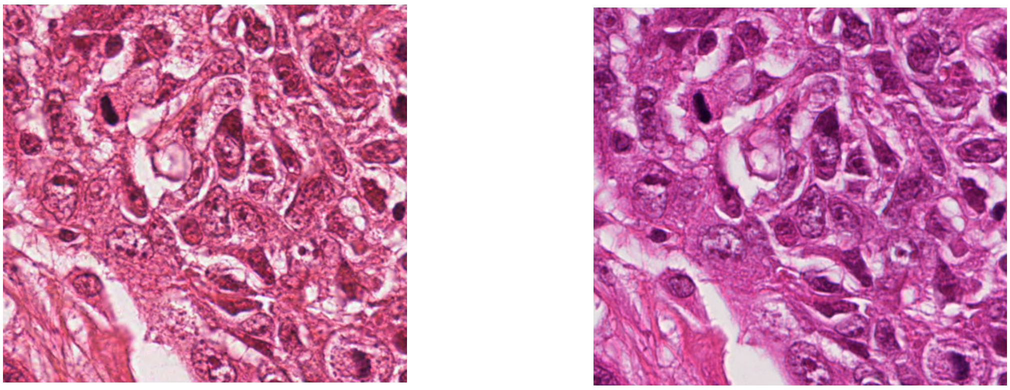
4.1 ABSTRACT

Computer based automatic medical image processing and quantification are becoming popular in digital pathology. However, preparation of histology slides can vary widely due to differences in staining equipment, procedures and reagents, which can reduce the accuracy of algorithms that analyze their color and texture information. To reduce the unwanted color variations, various supervised and unsupervised color normalization methods have been proposed. Compared with supervised color normalization methods, unsupervised color normalization methods have advantages of time and cost efficient and universal applicability. Most of the unsupervised color normalization methods for histology are based on stain separation. Based on the fact that stain concentration cannot be negative and different parts of the tissue absorb different stains, nonnegative matrix factorization (NMF), and particular its sparse version (SNMF), are good candidates for stain separation. However, most of the existing unsupervised color normalization method like PCA, ICA, NMF and SNMF fail to consider important information about sparse manifolds that its pixels occupy, which could potentially result in loss of texture information during

color normalization. Manifold learning methods like Graph Laplacian have proven to be very effective in interpreting high-dimensional data. In this paper, we propose a novel unsupervised stain separation method called graph regularized sparse nonnegative matrix factorization (GSNMF). By considering the sparse prior of stain concentration together with manifold information from high-dimensional image data, our method shows better performance in stain color deconvolution than existing unsupervised color deconvolution methods, especially in keeping connected texture information. To utilized the texture information, we construct a nearest neighbor graph between pixels within a spatial area of an image based on their distances using heat kernal in $l\alpha\beta$ space. The representation of a pixel in the stain density space is constrained to follow the feature distance of the pixel to pixels in the neighborhood graph. Utilizing color matrix transfer method with the stain concentrations found using our GSNMF method, the color normalization performance was also better than existing methods.

4.2 INTRODUCTION

Digital histopathology is a research field where image processing techniques and pattern recognition methods are exploited to enable computers to understand histopathology images and to aid diagnosis decisions. Recently, with fast development of machine learning in computerized artificial intelligence, automatic medical image processing has shown its impact in the field of digital pathology. It makes quantitative analysis of large number of histology slides both time and cost efficient. However, slide preparation can vary widely due to different stain manufacturers (as shown in Figure Figure 28, stain variation exists when scanned by two different scanners, even with the same tissue section), different staining procedures and different storage times. Unwanted stain color effects is less a problem for trained pathologists, but it can reduce the accuracy of computational methods.



Aperio Scanner Hamamatsu Scanner

Figure 28. Same tissue section under Aperio and Hamamatsu scanner

To overcome the color variation and distortion problem. One way is to only analyze gray level images, it will perform very well when gray level intensity is the primary cue - Basavanhally [107] uses the fact that nuclei is much darker than surrounding anatomy for lymphocytes detection. However, many important features are related to colors and wealth of information in the color representation are used routinely by pathologists. Another way is color normalization, which is a process proposed to reduce the adverse effects of stain variation. Sethi [108] made an empirical comparison of color normalization methods for epithelial-stromal classification in HE images in which they concluded that it helps the classification task, irrespective of the color normalization techniques tested. However, there are also concerns that it will add distortions or artifacts to normalized images. Hence, stable and high quality color normalization is long seeking in digital histopathology.

With strong desire of well performing color normalization methods, biologists and computer scientists are working together to come up with various categories of methods. The existing color normalization methods can be briefly separated into three categories. The first category is RGB histogram matching, which is done by matching the histograms of RGB channels of target and source images. Histogram matching is the basic method in digital image processing, it is well known and easy to implement. However, this is an ill-posed method, histogram matching often results color distortion and background staining. The second category is matching histogram statistics (mean and standard deviation). Representation of this method is Reinhard [109]'s color transfer method, which first convert image from RGB color space to $l\alpha\beta$ color space then match the histogram statistics (mean and variance) of each channel. This method also facing the problem of background staining. The third category is normalization after separating RGB image into channels of stain concentration or we say normalization after color deconvolution. Most of the existing state-of-the-art color normalization methods are in this category, our method mainly focus on making the color deconvolution process accurate, stable and automatic, hence it's also in category three.

Here we introduce several state-of-the-art color normalizations and their limitation. Magee [110] proposed a enhanced Reinhard's method by adding automatic image segmentation before statistics matching, and then use color deconvolution method for color normalization. He utilizes a supervised method to estimate the image specific color deconvolution vectors. The color normalization process can easily fail if this supervised estimation processed is biased or failed. Almost at the same time, Macenko [111] proposed a unsupervised color normalization method by first create plane from the SVD directions of optical density corresponding to the two largest singular values, then project data onto this plane. However, SVD modifies the color distribution of both source and target images, which is not desirable as we want reference image to stay unchanged for an automatic system. Recently, Khan [112] proposed a nonlinear mapping approach using image-specific color deconvolution, which is a supervised method. It showed great performance with standard HE stained images. However, it's not universally applicable to other stains such as DAB and Feulgen stains. Additionaly, its performance is not stable when deal with low-quality histology images. Vahadane [113] and Xu [114] both proposed a sparse non-negative matrix factorization method which preserves the structure of histological images by sparsity control. However, there is a potential of losing texture details with this method. Li [115] proposed a complete color normalization approach by using color cues computed from saturation-weighted statistics. However, it is sensitive to quality of histology images. With low contrast stain, the results are misleading.

Compared with supervised color normalization method, unsupervised color normalization methods are less expensive, less time consuming and easier to use. Various unsupervised color normalization methods based on ICA, PCA, NMF and SNMF have been proposed. They show great performance in general color normalization process, but do not preserve texture information properly.

In this paper, we propose a new unsupervised color deconvolution method called graph regularized sparse nonnegative matrix factorization (GSNMF). By interpreting high-dimensional texture information with graph Laplacian and stain concentration with sparsity constrain, GSNMF can better grasp detain texture information during the color deconvolution process.

4.3 Color Separation Methods

In 2001, Ruifrok and Johnston [116] proposed a CD framework with potential application in histopathology image analysis. The CD framework transforms the RGB color space to a new color space (optical density) defined by the stains used for staining the tissue section. For example, an 2D image **I** is transformed to $\mathbf{Y}$ in optical density space via Lambert-Beers law as follows:

$$\mathbf{I} = \mathrm{I}_0 \exp(-\mathbf{Y}), \ \mathbf{Y} = \log(\frac{\mathrm{I}_0}{\mathbf{I}}) \tag{4.1}$$

Where I_0 is the illuminating light on the sample.

Recall that NMF [117] tries to find a set of basis vectors that can be used to best approximate the data. One might further hope that the basis vectors can respect the intrinsic Riemannian structure of the image patches. A natural assumption here is that if two data points are close in the intrinsic geometry of the data distribution, then the representations of these two points with respect to the new basis, are also close to each other. This assumption is usually referred to as local invariance assumption, which plays an essential role in the development of various kinds of algorithms including dimensionality reduction algorithms and semi-supervised learning algorithms.

Recent studies in spectral graph theory and manifold learning have demonstrated that the local geometric structure can be effectively modeled through a nearest neighbor graph on a scatter of data points. Cai [33] proposed and demonstrated that a graph regularized NMF outperforms NMF in data representation. Zhu [118] proposed a graph regularized sparse NMF framework to solved the hyperspectral unmixing problem. Our graph regularized sparse NMF is another extension and application of Cai [33] and Zhu [118]'s work in histopathology.

Consider a graph with N vertex where each vertex corresponds to a data point. For each data point, we find its n nearest neighbors and construct edges between it and its neighbors. To do this, we first transfer image from RGB color space to $l\alpha\beta$ color space. Recall we have transformed image from RGB color space to optical density color space in equation (1), why we again transform image from RGB color space to $l\alpha\beta$? The reason is that these two transformation are for two different purposes. Matrix $\mathbf{Y}$ in equation (1) is for further stain separation, however, transformed image in $l\alpha\beta$ space is used for graph matrix or weight matrix calculation.

There are enormous ways of calculating weights between data points. Here, we adopt Heat Kernel (HK) method, cause it best represents differences and ranges from 0 to 1. Heat kernel for two data points in $l\alpha\beta$ color space is as follows:

$$W_{ij} = \begin{cases} \mathrm{HK}(\mathbf{l}_i, \mathbf{l}_j), \mathbf{l}_i \in \mathcal{N}(\mathbf{l}_j) \text{ and } \mathbf{l}_j \in \mathcal{N}(\mathbf{l}_i) \\ \\ 0, \text{ otherwise} \end{cases} \tag{4.2}$$

$$\text{Where } \mathrm{HK}(\mathbf{l}_i, \mathbf{l}_j) = e^{-\frac{\|\mathbf{l}_i - \mathbf{l}_j\|^2}{\sigma}}, \mathbf{l} = vector(\mathbf{L}), \mathbf{L} = \mathrm{RGB2}l\alpha\beta(\mathbf{I})$$

To meet the criterion of being point $\mathbf{l}_i$'s nearest neighbors, a point $\mathbf{l}_j$ needs to first be within a spatial distance. This is done by making a $n \times n(n = 5)$ local window which centers at point $\mathbf{l}_i$, if point $\mathbf{l}_j$ is in this local window, it is a local neighbor and will be used for weights calculation, otherwise, it's 0 in weight matrix. Second, among those n^2 neighbors, only Heat kernel weights within the top 50% will be kept (empirical threshold), the lower 50% will be force to 0. This is summarized as follows:

- Nearest spatial distance, i.e. $\mathbf{l}_j$ is in the $n \times n(n = 5)$ local window centers at $\mathbf{l}_i$.
- Nearest feature distance, i.e. calculate the $\mathrm{HK}(\mathbf{l}_i, \mathbf{l}_j)$, weights with top 50% values will be kept, otherwise, force to 0.

Figure Figure 29 explains the effects of local window and merging local windows with structure grouping.

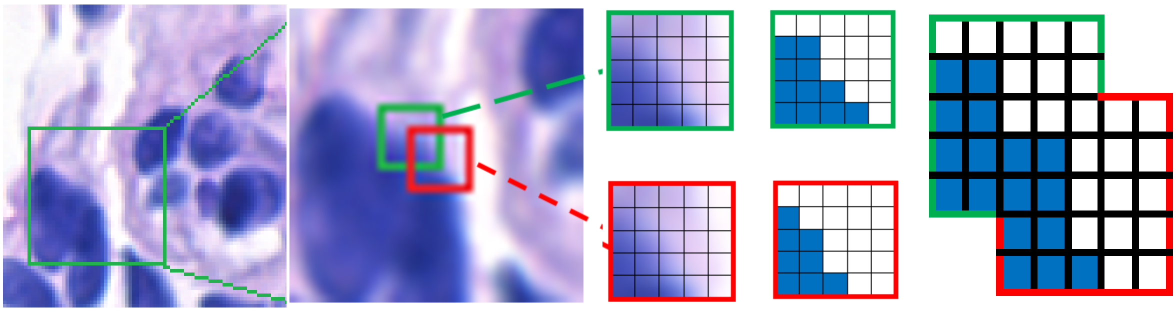


Figure 29. Highly similar neighboring pixels could be grouped by local structures. (a) H&E image patch, (b) Even smaller patch from (a) with complex texture, (c) local neighboring window examples, (d) highly similar pixels in local structure within a window are grouped together, (e) when the local structures are combined, the highly similar pixels in the two windows are grouped together.

4.3.1 Proposed color separation method

Assume the separated coefficient vector is $\mathbf{x}$, the coefficient distance correspond and regularization term is:

$$d(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|^2 \tag{4.3}$$

By multiplying the coefficient distance with Heat kernel weight matrix, it will help forcing the two coefficient vector to close when we minimize the cost function and their weight coefficient is nonzero.

$$\begin{aligned} R &= \frac{1}{2}\sum_{i,j=1}^{N} \|\mathbf{x}_i - \mathbf{x}_j\|^2 W_{ij} + \alpha \sum_{n=1}^{N} \|\mathbf{x}_n\|_1 = \sum_{i=1}^{N} \mathbf{x}_i^T \mathbf{x}_i E_{ii} - \sum_{i,j=1}^{N} \mathbf{x}_i^T \mathbf{x}_j W_{ij} + \alpha \sum_{n=1}^{N} \|\mathbf{x}_n\|_1 \\ &= \mathrm{Tr}(\mathbf{X}^T\mathbf{E}\mathbf{X}) - \mathrm{Tr}(\mathbf{X}^T\mathbf{W}\mathbf{X}) + \alpha \sum_{n=1}^{N} \|\mathbf{x}_n\|_1 = \mathrm{Tr}(\mathbf{X}^T\mathbf{L}\mathbf{X}) + \alpha\|\mathbf{X}\|_1 \end{aligned} \tag{4.4}$$

Where $\mathrm{Tr}(\cdot)$ denote the trace of a matrix and $\mathbf{E}$ is a diagonal matrix whose entries are column (or row, since $\mathbf{W}$ is symmetric) sums of $\mathbf{W}$, $E_{ii} = \sum_j W_{ij}$. $\mathbf{L} = \mathbf{E} - \mathbf{W}$, which is called graph Laplacian.

Objective function is defined as

$$\mathcal{O} = \frac{1}{2}\|\mathbf{Y} - \mathbf{DX}\|_F^2 + \frac{\lambda}{2}\mathrm{Tr}(\mathbf{X^T L X}) + \alpha\|\mathbf{X}\|_1 \tag{4.5}$$

Where $\lambda \geq 0, \alpha \geq 0, \mathbf{D}, \mathbf{X} \geq 0$, $\|\mathbf{D}(:, j)\|_2^2 = 1, j = 1, 2, ..., r$.

Optimization for GSNMF

$$\min_{D,X} \frac{1}{2}\|\mathbf{Y} - \mathbf{DX}\|_F^2 + \frac{\lambda}{2}\mathrm{Tr}(\mathbf{X^T L X}) + \alpha \sum_{k=1}^{K}\sum_{n=1}^{N} X_{kn}, \text{s.t.} D \geq 0, X \geq 0 \tag{4.6}$$

Let θ_{lk}, ϕ_{kn} be the Laplacian multipliers for constraint $D_{lk} \geq 0$ and $X_{kn} \geq 0$ respectively, and $\Theta = [\theta_{lk}] \in \mathbb{R}_+^{L \times K}, \Phi = [\phi_{kn}] \in \mathbb{R}_+^{K \times N}$. The Lagrange $\mathcal{L}$ is given by

$$\mathcal{L} = \frac{1}{2}\|\mathbf{Y} - \mathbf{DX}\|_F^2 + \frac{\lambda}{2}\mathrm{Tr}(\mathbf{X^T L X}) + \alpha \sum_{k=1}^{K}\sum_{n=1}^{N} x_{kn} + \mathrm{Tr}(\Theta \mathbf{D^T}) + \mathrm{Tr}(\Phi \mathbf{X^T}) \tag{4.7}$$

We can further obtain the partial derivative of $\mathcal{L}$ with respect to $\mathbf{D}$ and $\mathbf{X}$ as

$$\frac{\partial \mathcal{L}}{\partial \mathbf{D}} = \mathbf{DXX^T} - \mathbf{YX^T} + \Theta, \ \frac{\partial \mathcal{L}}{\partial \mathbf{X}} = \mathbf{D^T DX^T} - \mathbf{X^T Y} + \alpha + \Phi \tag{4.8}$$

Based on the Karush-Kuhn-Tucker conditions $\theta_{lk}D_{lk} = 0$ and $\phi_{kn}X_{kn} = 0$, we could obtain the following equations by letting the above partial derivatives equal to zero and multiplying both sides with D_{lk} and X_{kn} respectively,

$$\begin{aligned} (\mathbf{DX}\mathbf{X^T})_{lk}\mathbf{D}_{lk} - (\mathbf{YX^T})_{lk}\mathbf{D}_{lk} = 0 \\ (\mathbf{D^T}\mathbf{D}\mathbf{X^T})_{kn}\mathbf{X}_{kn} - (\mathbf{X^T}\mathbf{Y})_{kn}\mathbf{X}_{kn} + \lambda(\mathbf{XL})_{kn}\mathbf{X}_{kn} + \alpha\mathbf{X}_{kn} = 0 \end{aligned} \tag{4.9}$$

With equation $\mathbf{L} = \mathbf{E} - \mathbf{W}$, we get

$$[(\mathbf{D^T}\mathbf{DX})_{kn}\mathbf{X}_{kn} + \lambda(\mathbf{LE})_{kn}\mathbf{X}_{kn} + \alpha]\mathbf{X}_{kn} = [(\mathbf{D^T}\mathbf{Y})_{kn} + \lambda(\mathbf{LW})_{kn}]\mathbf{X}_{kn} \tag{4.10}$$

We can get the updating rules as

$$\mathbf{D}_{lk} \leftarrow \mathbf{D}_{lk}\frac{(\mathbf{YX^T})_{lk}}{(\mathbf{DXX^T})_{lk}},\ \mathbf{X}_{kn} \leftarrow \mathbf{X}_{kn}\frac{(\mathbf{D^T}\mathbf{Y} + \lambda\mathbf{XW})_{kn}}{(\mathbf{DXX^T} + \lambda\mathbf{XE} + \alpha)_{kn}} \tag{4.11}$$

4.3.2 System Illustration

Figure Figure 30 is a vivid illustration of our GSNMF system. Image **I** is transformed to $l\alpha\beta$ color space for calculating weight matrix, it's also transformed to optical density color space for stain separation. Normalization step is done by exchanging the color matrix.

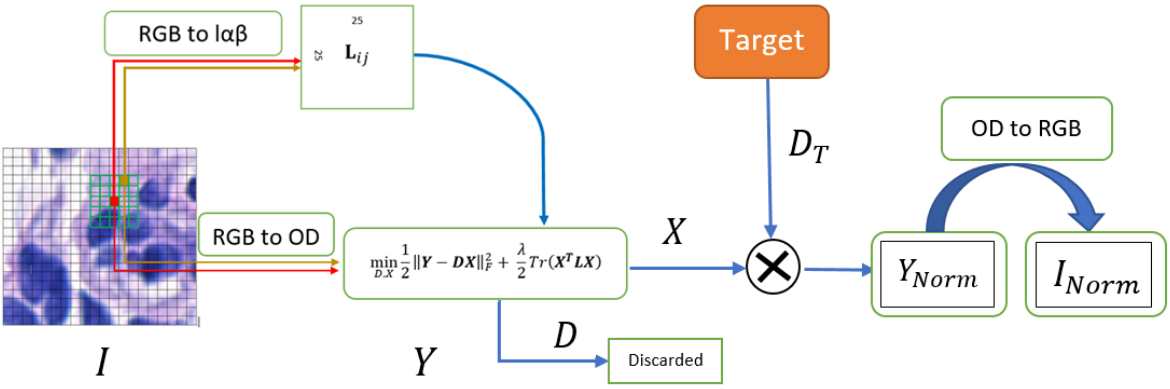


Figure 30. Illustration of GSNMF Color Normalization System.

4.4 Color Transfer

As our main focus of this paper is on color deconvolution, we utilize the similar color transfer method as Vahadane [113] and Macenko [111]. From equation (6), **D** can be considered as color matrix and **X** as coefficient matrix, we exchange the color matrix between source image and target image.

$$\mathbf{Y}_{scource} = \mathbf{D}_{source}\mathbf{X}_{source}, \ \mathbf{Y}_{target} = \mathbf{D}_{target}\mathbf{X}_{target} \tag{4.12}$$

We calculate 95 percentile of each row of $\mathbf{X}_{source}$ and $\mathbf{X}_{target}$ as $X_{S-extreme}(i)$ and $X_{T-extreme}(i)$ then come up with a scale matrix

$$H_{scale}(i, 1) = \frac{X_{T-extreme}(i)}{X_{S-extreme}(i)} \tag{4.13}$$

Where $i = 1, ..., r$ which is the row number in coefficient matrix $\mathbf{X}$. In our case of color normalization $r = 2$.

The scale row of $\mathbf{X}_{source}$ are represented as

$$\mathbf{X}_{s-scaled}(i,:) = \mathbf{X}_{source}(i,:)H_{scale}(i,1) \tag{4.14}$$

New transferred image in optical density space

$$\mathbf{Y}_{transferred} = \mathbf{D}_{target}\mathbf{X}_{s-scaled} \tag{4.15}$$

From optical density space back to RGB space

$$\mathbf{I}_{normalized} = \mathbf{I}_0 \exp(-\mathbf{Y}_{transferred}) \tag{4.16}$$

Where $\mathbf{I}_0 = 255$

4.5 Experiments and Results

4.5.1 Stain Separation

4.5.1.1 H&E stain

The quality of stain separation is essential cause it has the significant influence on color normalization result. Here we qualitatively compare our stain separation results with state-of-the-art unsupervised stain separation methods: NMF and SNMF. Figure Figure 31 shows stain separation experiments on two H&E stained breast tissue section. It's easy to find out NMF gives the worst separation result. With

SNMF, the separation of nucleus and epithelium is great, however, it tends to be clear cut and losing texture details especially in the area where nucleus resides. Our proposed GSNMF method, because of grouping effects, sparse separation and smooth texture are well reserved at the same time. The separation of nucleus and epithelium are smoothly and naturally, you can easily find from figure Figure 31, the separated epithelium area still has very light tissue there instead of white space which is usually true in nature - nuclei are surrounded by tissues. They are not 100% isolated, they are slightly connected.

4.5.1.2 H&DAB stain

Another very popular stain for breast histopathology research is H&DAB stain. This can show an important advantage of our method - unsupervised. It will automatically adapted to the case for stain separation instead of retrain or use predefined stain matrix for stain separation, which can increase the separation accuracy. In figure Figure 32, both SNMF and our GSNMF easily outperforms NMF method in H&DAB stain separation. Since the contrast between H stain and DAB stain is more obvious than most H stain and E stain, it's hard to tell the difference between SNMF and GSNMF separation results.

4.5.2 Color Normalized Image Comparison

To further show the outstanding performance of our color normalization method. We compare our color normalization results with normalized outputs from several other state-of-the-art color normalization methods: Khan [112]'s nonlinear mapping color normalization method, Li [115]'s color cues method and Vahadane [113]'s sparse NMF method.

Since both SNMF and our GSNMF methods are coming from same nonnegative assumption, we first demonstrate the advantage of our method against SNMF method. Figure Figure 33 shows the color normalization results of a breast tissue section by both SNMF method and our GSNMF method. It's easy

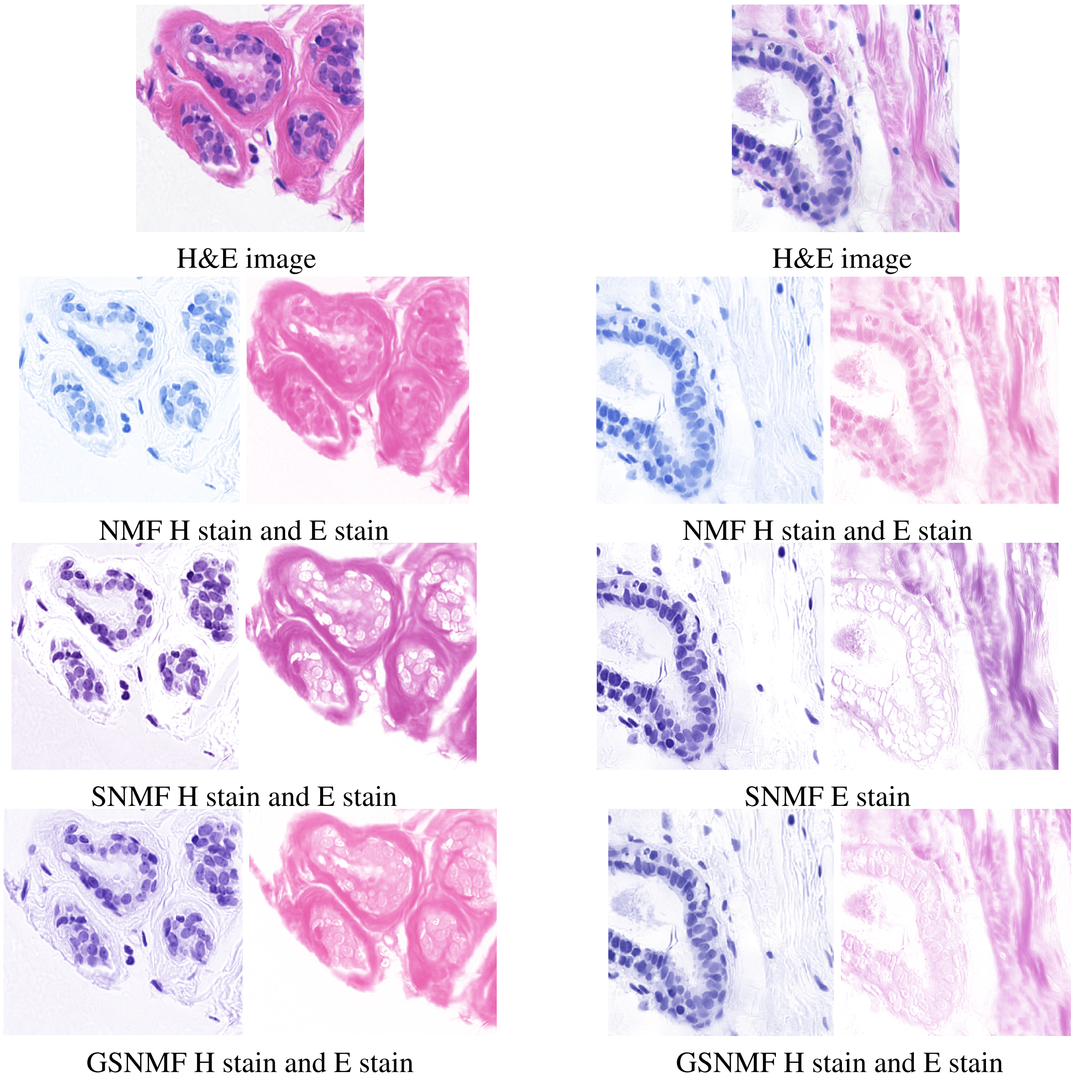
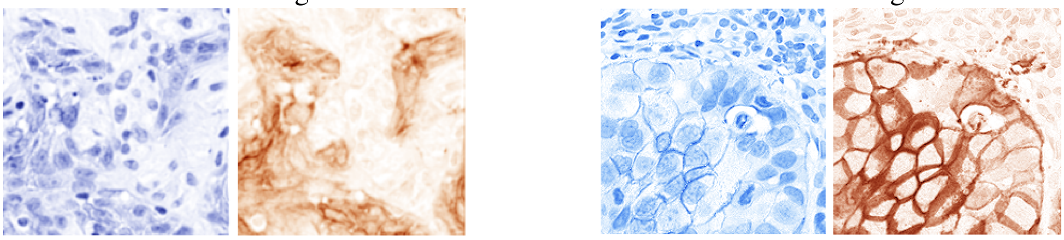
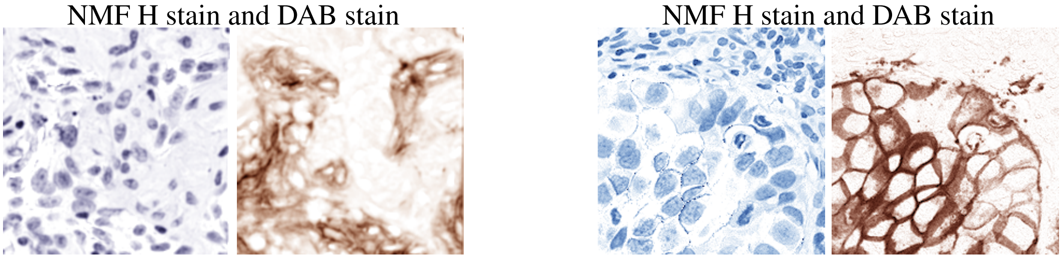


Figure 31. H&E images color deconvolution by NMF, SNMF and GSNMF

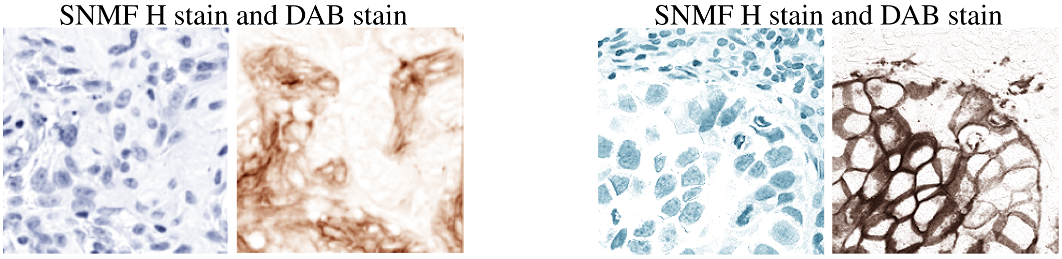
H&DAB image H&DAB image



NMF H stain and DAB stain NMF H stain and DAB stain

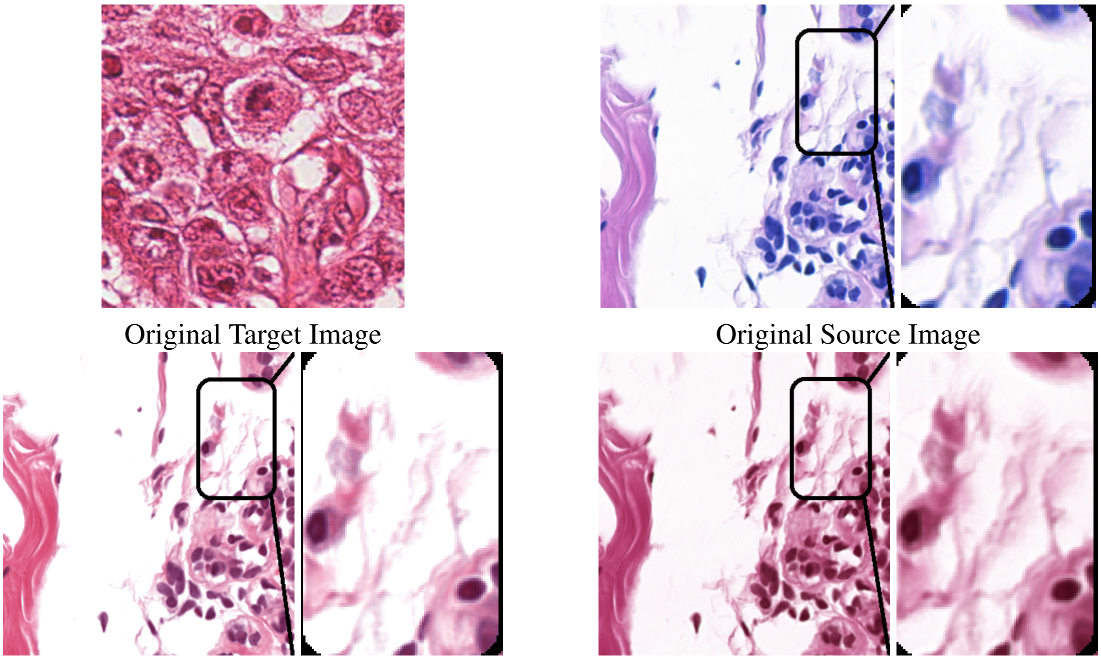


SNMF H stain and DAB stain SNMF H stain and DAB stain



GSNMF H stain and DAB stain GSNMF H stain and DAB stain

Figure 32. H&DAB images color deconvolution by NMF, SNMF and GSNMF



Original Target Image

Original Source Image

Normalized by SNMF method

Normalized by proposed GSNMF method

Figure 33. Slides detail capturing by SNMF and Propose GSNMF methods

to find from the magnified images that SNMF method is losing texture details where GSNMF well kept. In digital histopathology, one of the most important concern is image distortion. Our GSNMF method show excellent performance in keeping texture detail while color deconvolution and normalization.

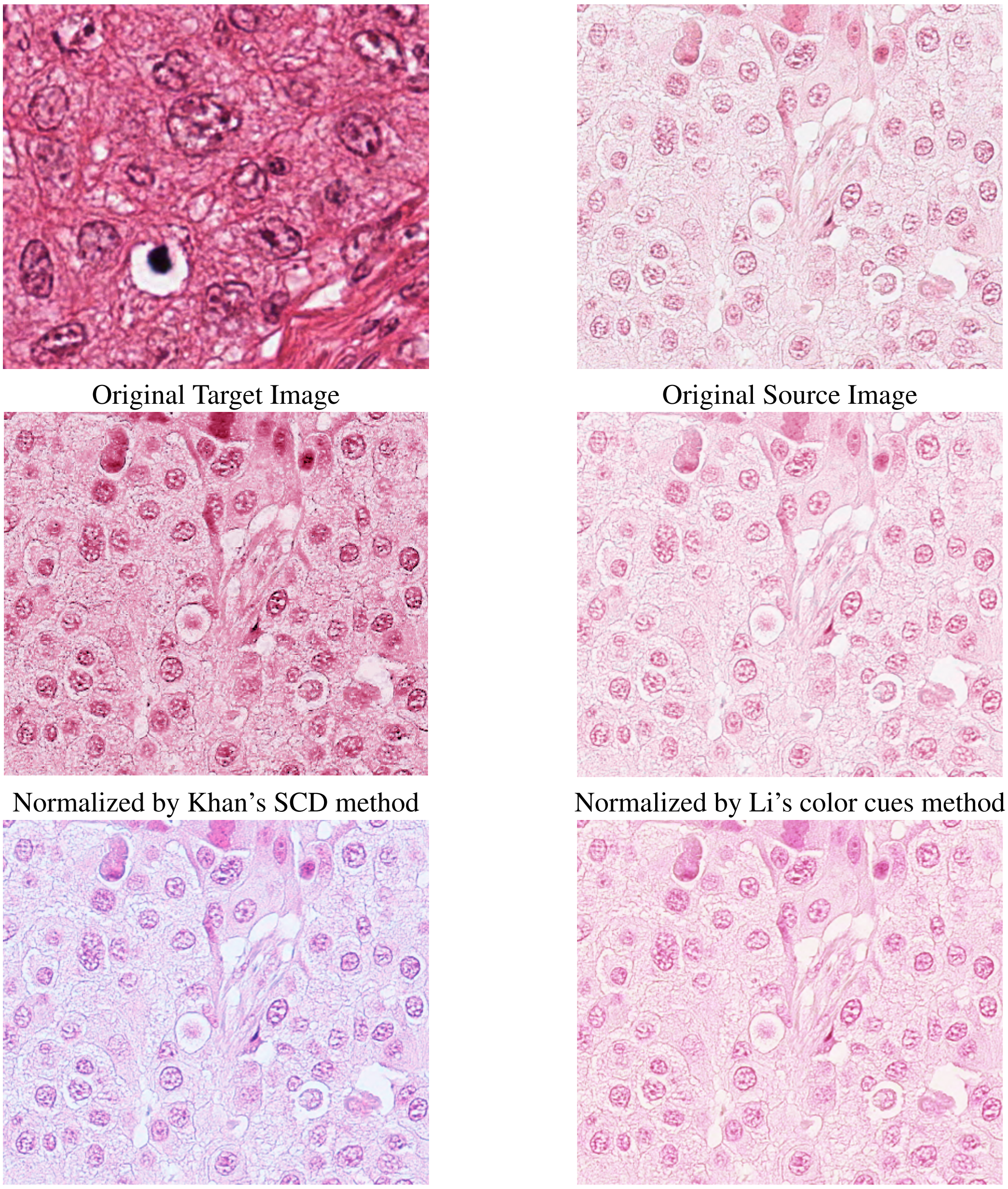
To compare all four methods simultaneously, we choose the low quality of contrast example which has been used by Vahadane [113]. From figure Figure 34, we can find image from Khan's method is over-saturated and distorted; image from Li's method is slightly different from original source image, which means color normalization is almost ineffective; SNMF color normalized image shows shift-ed color which is different from both source and target image. Our GSNMF method gives the most reasonable result by showing correctly transfer color and contrast.

4.6 Conclusion

In this paper, we proposed a novel graph regularized sparse NMF method for histology slides color normalization. This can reduce the slides' color variation, rescue faded or distorted slides. Compared with the state-of-the-art color normalization methods, our method has the advantage of being stable, highly accurate, responsive and self-adaptive. Highly accurate stain separation plus texture preserving ensures us not losing any information during the process of color normalization. Unsupervised nature of our method makes it easy to use without any training, labeling and annotation from biologist. Stability and speed of processing of GSNMF makes it practical to be used for color normalization on whole slide images.

4.7 ACKNOWLEDGMENTS

Lingdao Sha would like to thank Prof. Peter Gann's guidance and help during his research in digital histopathology.



Original Target Image

Original Source Image

Normalized by Khan's SCD method

Normalized by Li's color cues method

Normalized by SNMF method

Normalized by proposed GSNMF method

Figure 34. Low constrast H&E image color normalization by Khan's, Li's, SNMF and Proposed GSNMF methods

CHAPTER 5

LOCALLY LINEAR EMBEDDED SPARSE CODING FOR IMAGE REPRESENTATION (PREVIOUSLY PUBLISHED AS L. SHA AND D. SCHONFELD AND J. WANG (2017) LOCALLY LINEAR EMBEDDED SPARSE CODING FOR IMAGE REPRESENTATION, 2017 IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING (ICASSP), 2527-2531.)

5.1 ABSTRACT

Recently, sparse coding has been widely and successfully used in image classification, noise reduction, texture synthesis and audio processing. Although traditional sparse coding method with fixed dictionaries like wavelet and curvelet can produce promising results, unsupervised sparse coding has shown its advantage by optimizing the dictionary adaptively. However, existing unsupervised sparse coding failed to consider the high dimensional manifold information within data. Recently, a graph regularized sparse coding method has shown outstanding performance by incorporating graph laplacian manifold information. In this paper, we proposed a sparse coding method called locally linear embedded sparse coding, to consider the local manifold structure as well as learning the sparse representation. We also provided a novel modified online dictionary learning method which iteratively utilizes modified least angle regression and block coordinate descent method to solve the problem. Instead of getting entire coefficient matrix then update dictionary matrix, our method updates coefficient vector and dic-

tionary matrix in each inner iteration. Extensive experimental results have demonstrated the efficiency and accuracy of our method in image clustering.

5.2 Introduction

Sparse coding enables successful representation of stimuli with only a few active coefficients. It has shown state-of-art results in ordinary signal processing tasks like image denoising [119] and restoration [120], audio [121] and video processing [122], as well as more complicated tasks like image classification [123] and image clustering [35]. When applied to natural images, sparse coding produces learned bases that can resemble the receptive fields of neurons in the visual cortex [27], which is similar to the results of Independent Component Analysis (ICA) [124] and Gabor filter [125]. Compared with other unsupervised methods like PCA and ICA, sparse coding can learn overcomplete basis sets and doesn't require statistical-independence of the dictionary prototype signals. In machine learning and statistics, slightly different matrix factorization problems such as non-negative matrix factorization, its variants [117] [28] and sparse principal component analysis [126] have been successfully used to obtain interpretable basis elements.

When dealing with high dimensional feature space in image clustering and classification, sparse coding with dimensionality reduction becomes a reasonable thought. Cai [33] proposed a graph regularized nonnegative matrix factorization (NMF) method, inspired by his work, Gao [34] and Zheng [35] proposed graph regularized sparse coding (GraphSC), which explicitly considers the local geometrical structure of the data. In those epic work, graph regularized NMF and sparse coding show big improvement on image clustering compared with existing NMF and sparse coding. However, all of these graph regularized work are based on graph laplacian method, which is only one of the many manifold learning methods. In this paper, we proposed a locally linear embedded sparse coding method (LLESC) together with a novel modified online dictionary learning method (MODL) to solve the objective function efficiently.

The rest of this paper is organized as follows: In Section II, we give a brief description of sparse coding problem and popular methods to solve the sparse coding problem. Section III introduces the LLESC algorithm, as well as the MODL solution. Experimental results on image clustering are presented in Section IV.

5.2.1 Contribution

- Use K-SVD instead of PCA for preprocessing, less processing time for convergence.
- Locally linear embedding method (LLE) [30] compared with Graph Laplacian [31] for constrains.
- A novel modified online dictionary learning algorithm to solve the graph regularized sparse coding problem efficiently.
- Use SIFT compared with Euclidean distance for weight matrix calculation.

5.3 A brief review of sparse coding

Given a data matrix $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_m] \in \mathbb{R}^{n \times m}$, let $\mathbf{D} = [\mathbf{d}_1, ..., \mathbf{d}_k] \in \mathbb{R}^{n \times k}$, where each $\mathbf{d}_i$ represents a basis vector in the dictionary, and $\mathbf{A} = [\boldsymbol{\alpha}_1, ..., \boldsymbol{\alpha}_m] \in \mathbb{R}^{k \times m}$ be the coefficient matrix, where each column is a sparse representation for a data point. A good dictionary and coefficient pair should minimize the empirical loss function, which can be represented as $\sum_{i=1}^{m} \|\mathbf{x}_i - \mathbf{D}\alpha_i\|_p$. The typical norms used for measuring the loss function are the L_p norms where $p = 1, 2$ and ∞. Here we concentrate on least square loss problems when $p = 2$.

The objective function of sparse coding can be formulated as:

$$\min_{\mathbf{D},\mathbf{A}} \|\mathbf{X} - \mathbf{D}\mathbf{A}\|_F^2 + \beta \sum_{i=1}^{m} f(\alpha_i) \quad \text{s.t. } \|\mathbf{d}_i\|^2 \leqslant c,\ i = 1, ..., k \tag{5.1}$$

where f is a function to measure the sparseness of α_i and $\|\cdot\|_F$ denotes the matrix Frobenius norm.

Following [60] [37], we adopt the idea of L_1 norm instead of L_0, which can produce similar results with affordable computational cost. The objective function then becomes:

$$\min_{\mathbf{D},\mathbf{A}} \|\mathbf{X} - \mathbf{D}\mathbf{A}\|_F^2 + \beta \sum_{i=1}^{m} \|\alpha_i\|_1 \quad \text{s.t. } \|\mathbf{d}_i\|^2 \leqslant c,\ i = 1, ..., k \tag{5.2}$$

Although the objective function is not convex with **D** and **A** together, it is convex with either one fixed. We iteratively optimize the objective function by minimizing over one variable with the other one fixed. Thus, it becomes an L_1-regularized least squares problem with an L_2-constrained least square problem.

5.4 Locally linear embedded sparse coding (LLESC)

5.4.1 Algorithm Outline

Locally linear embedding (LLE) is an unsupervised learning algorithm that computes low dimensional, neighborhood preserving embedding of high dimensional data. LLE attempts to discover nonlinear structure in high dimensional data by exploiting the local symmetries of linear reconstruction [30]. Given a set of m dimensional data points x_1, ..., x_m, we can characterize the local geometry of these patches by linear coefficients that reconstruct each data point from its neighbors. Reconstruction errors are:

$$\frac{1}{2}\sum_{i=1}^{m}|\mathbf{x}_i - \sum_{j} W_{ij}\mathbf{x}_j|^2 = \mathrm{Tr}(\mathbf{X}\mathbf{L}\mathbf{X}^{\mathsf{T}}) \tag{5.3}$$

where $\mathbf{L} = (\mathbf{I} - \mathbf{W})^{\mathbf{T}}(\mathbf{I} - \mathbf{W})$, $\mathbf{I}$ is identity matrix, $\mathbf{W}$ is weight matrix.

Nearest neighbor is a necessary step to compute the weight matrix. Besides using Euclidean distance, we also utilizes scale-invariant feature transform (SIFT) [127] for nearest neighbor calculation, which shows better performance in situations with scaled and rotated image objects.

LLE constructs a neighborhood preserving mapping: $\mathbf{x}_i \mapsto \boldsymbol{\alpha}_i$. By incorporating the LLE regularizer into the original sparse coding, we can get the following objective function of LLESC:

$$\min_{\mathbf{D},\mathbf{A}} \|\mathbf{X} - \mathbf{D}\mathbf{A}\|_F^2 + \lambda\mathrm{Tr}(\mathbf{A}\mathbf{L}\mathbf{A}^{\mathsf{T}}) + \beta\sum_{i=1}^{m}\|\boldsymbol{\alpha}_i\|_1 \quad \text{s.t. } \|\mathbf{d}_i\|^2 \leqslant c,\ i = 1, ..., k \tag{5.4}$$

where $\lambda \geqslant 0$ is the regularization parameter.

5.4.2 Coefficients Learning and Dictionary Learning

In this section, we show how to solve Equation 5.4 with modified online dictionary learning algorithm.

Fixing dictionary $\mathbf{D}$, the objective function becomes:

$$\min_{\mathbf{A}} \|\mathbf{X} - \mathbf{D}\mathbf{A}\|_F^2 + \lambda\mathrm{Tr}(\mathbf{A}\mathbf{L}\mathbf{A}^{\mathsf{T}}) + \beta\sum_{i=1}^{m}\|\boldsymbol{\alpha}_i\|_1 \tag{5.5}$$

As Equation 5.5 is convex, global minimum can be achieved [61].

With modified online dictionary learning, we update each vector $\boldsymbol{\alpha}_i$ individually, while keeping all the other vectors constant. In order to solve the problem by optimizing over each $\boldsymbol{\alpha}_i$, we rewrite Equation 5.5 in vector form.

Reconstruction error $\|\mathbf{X} - \mathbf{D}\mathbf{A}\|_F^2$ can be written as:

$$\sum_{i=1}^{m} \|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|^2 \tag{5.6}$$

As matrix $\mathbf{L}$ is symmetric in LLE, the regularizer $\mathrm{Tr}(\mathbf{A}\mathbf{L}\mathbf{A}^T)$ can be rewritten as:

$$\mathrm{Tr}(\mathbf{A}\mathbf{L}\mathbf{A}^T) = \mathrm{Tr}(\sum_{i,j=1}^{m} L_{ij}\boldsymbol{\alpha}_i\boldsymbol{\alpha}_j^T) = \sum_{i,j=1}^{m} L_{ij}\boldsymbol{\alpha}_i^T\boldsymbol{\alpha}_j \tag{5.7}$$

We combine reconstruction error with LLE regularizer, add sparsity constrain to it, the objective function becomes:

$$\min_{\boldsymbol{\alpha}_i} \sum_{i=1}^{m} \|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|^2 + \lambda \sum_{i,j=1}^{m} L_{ij}\boldsymbol{\alpha}_i^T\boldsymbol{\alpha}_j + \beta \sum_{i=1}^{m} \|\boldsymbol{\alpha}_i\|_1 \tag{5.8}$$

When updating $\boldsymbol{\alpha}_i$, the other vectors $\{\boldsymbol{\alpha}_j\}_{j\neq i}$ are fixed [35] [68]. Thus, we get the following optimization problem:

$$\min_{\boldsymbol{\alpha}_i} \|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|^2 + \lambda L_{ii}\boldsymbol{\alpha}_i^T\boldsymbol{\alpha}_j + \boldsymbol{\alpha}_i^T\mathbf{h}_i + \beta \sum_{j=1}^{k} |\boldsymbol{\alpha}_i^{(j)}| \tag{5.9}$$

Where $\mathbf{h}_i = 2\lambda(\sum_{j\neq i} L_{ij}\boldsymbol{\alpha}_j)$ and $\boldsymbol{\alpha}_i^{(j)}$ is the j-th coefficient of $\boldsymbol{\alpha}_i$.

In algorithm 5 of modified online dictionary learning (MODL), we keep dictionary $\mathbf{D}$ fixed, optimizing each individual coefficient α_i with all other coefficients fixed for each input data $\mathbf{x}_i$. The method used is modified least angle regression which will be explained in 6. Dictionary update is by block coordinate descent method, please reference [68] for detail.

Algorithm 5 Modified Online Dictionary Learning

Require: $\mathbf{x} \in \mathbb{R}^m$ from $p(\mathbf{x})$ ($\mathbf{x}$ sequentially aligned in $p(\mathbf{x})$), $\beta \in \mathbb{R}$ (regularization parameter), $\mathbf{D}_0 \in \mathbb{R}^{m\times k}$ (initial dictionary), T (number of samples in data set $p(\mathbf{x})$).

1: $\mathbf{A}_0 \in \mathbb{R}^{k\times k} \leftarrow 0$, $\mathbf{B}_0 \in \mathbb{R}^{m\times k} \leftarrow 0$ (Reset the "past"information)
2: **for** $t = 1$ **to** T **do**
3: Draw $\mathbf{x}_t$ from $p(\mathbf{x})$ (sequentially drawn)
4: Sparse coding: compute using modified LARS (algorithm 6):

$$\boldsymbol{\alpha}_t \triangleq \arg\min_{\boldsymbol{\alpha}\in\mathcal{R}^k} \frac{1}{2}\|\mathbf{x}_t - \mathbf{D}_{t-1}\boldsymbol{\alpha}\|_2^2 + \lambda L_{tt}\boldsymbol{\alpha}^T\boldsymbol{\alpha} + \boldsymbol{\alpha}^T\mathbf{h}_t + \beta\|\boldsymbol{\alpha}\|_1 \tag{5.10}$$

5: $\mathbf{A}_t \leftarrow \mathbf{A}_{t-1} + \boldsymbol{\alpha}_t\boldsymbol{\alpha}_t^T$
6: $\mathbf{B}_t \leftarrow \mathbf{B}_{t-1} + \mathbf{x}_t\boldsymbol{\alpha}_t^T$
7: Compute $\mathbf{D}_t$ using block coordinate descent method [68], with $\mathbf{D}_{t-1}$ as warm restart, so that

$$\begin{aligned}\mathbf{D}_t &\triangleq \arg\min_{\mathbf{D}\in\mathcal{C}} \frac{1}{t}\sum_{i=1}^{t}\left(\frac{1}{2}\|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda\sum_{i,j=1}^{m} L_{ij}\boldsymbol{\alpha}_i^T\boldsymbol{\alpha}_j + \beta\sum_{i=1}^{m}\|\boldsymbol{\alpha}_i\|_1\right) \\ &= \arg\min_{\mathbf{D}\in\mathcal{C}} \frac{1}{t}\left(\frac{1}{2}\mathrm{Tr}(\mathbf{D}^T\mathbf{D}\mathbf{A}_t) - \mathrm{Tr}(\mathbf{D}^T\mathbf{B}_t)\right)\end{aligned} \tag{5.11}$$

where $\mathcal{C} \triangleq \{\mathbf{D} \in \mathbb{R}^{m\times k} \quad \text{s.t. } \forall j = 1, ..., k, \mathbf{d}_j^T\mathbf{d}_j \leqslant 1\}$.
8: Return $\mathbf{D}_T$, $\mathbf{A}$ for complete dictionary and coefficients learning.

5.4.3 Modified Least Angle Regression

Least Angle Regression (LARS) [65] is a regression method that provides a general version of forward selection, which is highly efficient in solving LASSO [66]. We follow the steps presented in [67]. In step 7 of algorithm 6, instead of calculating the ordinary least square solution (OLS) of Equation 5.12, we calculate the locally linear embedded least square solution (LLELS) of Equation 5.13 to incorporate structure information.

$$\boldsymbol{\alpha}_{OLS}^{(k+1)} = (\mathbf{D}_{\mathcal{A}}^{T}\mathbf{D}_{\mathcal{A}})^{-1}\mathbf{D}_{\mathcal{A}}^{T}\mathbf{y} \tag{5.12}$$

$$\boldsymbol{\alpha}_{LLELS}^{(k+1)} = (\mathbf{D}_{\mathcal{A}}^{T}\mathbf{D}_{\mathcal{A}} + \lambda L_{kk}\mathbf{I})^{-1}(\mathbf{D}_{\mathcal{A}}^{T}\mathbf{x} - \mathbf{h}_{k}/2) \tag{5.13}$$

where $\mathbf{I}$ is identity matrix and $\mathbf{h}_k = 2\lambda(\sum_{k \neq j} L_{kj}\boldsymbol{\alpha}_j)$.

5.5 Experimental results

In this section, we present image clustering experiments on CMU-PIE and COIL data set [1], data statistics are shown in table Table XII. We compared clustering accuracy of our method (LLESC) against several unsupervised methods. We also compared the computation efficiency between LLESC and GrapSC methods [34] [35].

All clustering tasks are based on a Windows 10 machine with Intel Core i7-2820M 2.3GHz CPU and 16GB RAM. Algorithms were implemented and executed in MATLAB environment. We used VLFeat toolbox [2]for SIFT calculation.

---

[1] http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html

[2] http://www.vlfeat.org/

Algorithm 6 Modified Least Angle Regression

1: Initialize the coefficient vector $\boldsymbol{\alpha}^{(0)} = 0$ and the fitted vector $\hat{\mathbf{x}}^{(0)} = 0$.
2: Initialize the active set $\mathcal{A} = \phi$ and the inactive set $\mathcal{I} = 1, ..., p$.
3: **for** $k = 0$ **to** $p - 2$ **do**
4: Update the residual $\varepsilon = \mathbf{x} - \hat{\mathbf{x}}^{(k)}$
5: Find the maximal correlation $c = max_{i \in \mathcal{I}} |\mathbf{d}_i^T \varepsilon|$
6: Move variable corresponding to c from $\mathcal{I}$ to $\mathcal{A}$
7: Calculate the graph constrained least square solution:

$$\boldsymbol{\alpha}_{LLELS}^{(k+1)} = (\mathbf{D}_{\mathcal{A}}^T \mathbf{D}_{\mathcal{A}} + \lambda L_{kk} I)^{-1} (\mathbf{D}_{\mathcal{A}}^T x - h_k/2) \tag{5.14}$$

where I is identity matrix and $h_k = 2\lambda(\sum_{k \neq j} L_{kj} \boldsymbol{\alpha}_j)$
8: Calculate the current direction: $d = \mathbf{D}_{\mathcal{A}} \boldsymbol{\alpha}_{LLELS}^{(k+1)}$ - $\hat{\mathbf{x}}^{(k)}$
9: Calculate the step length: $\gamma = \min_{i \in \mathcal{I}}^{+} \{ \frac{\mathbf{d}_i^T \varepsilon - c}{\mathbf{d}_i^T d - c}, \frac{\mathbf{d}_i^T \varepsilon + c}{\mathbf{d}_i^T d + c} \}, 0 \leq \gamma \leq 1$
10: Update regression coefficients: $\boldsymbol{\alpha}^{(k+1)} = (1 - \gamma) \boldsymbol{\alpha}^{(k)} + \gamma \boldsymbol{\alpha}_{LLELS}^{(k+1)}$
11: Update the fitted vector $\hat{\mathbf{x}}^{(k+1)} = \hat{\mathbf{x}}^{(k)} + \gamma d$
12: Let $\boldsymbol{\alpha}^{(p)}$ be the full graph constrained least square solution:

$$\boldsymbol{\alpha}^{(p)} = (\mathbf{D}_{\mathcal{A}}^T \mathbf{D}_{\mathcal{A}} + \lambda L_{(p-1)(p-1)} \mathbf{I})^{-1} (\mathbf{D}_{\mathcal{A}}^T \mathbf{x} - h_{p-1}/2) \tag{5.15}$$

where **I** is identity matrix and $h_{p-1} = 2\lambda(\sum_{p-1 \neq j} L_{(p-1)j} \boldsymbol{\alpha}_j)$
13: **Output**: the series of coefficients $\mathbf{A} = [\boldsymbol{\alpha}^{(0)}, ..., \boldsymbol{\alpha}^{(p)}]$
Note: $\mathbf{d}_i$ is column of Dictionary $\mathbf{D}$, d is direction.

We use both PCA and K-SVD for preprocessing (pick the best results), after getting the coefficient matrix ($\mathbf{A}$) by GraphSC and LLESC, K-means method will be used to cluster those coefficients. We use computation time from matlab as efficiency evaluation metric, normalized mutual information (NMI) [33] [35] as clustering accuracy evaluation metric.

Table Table XIII, figure Figure 35, table Table XIV and figure Figure 36 shows LLESC clustering results on CMU-PIE and COIL data set. Figure Figure 37 shows an example of SIFT matching of two images with different orientations. Figure Figure 38 and figure Figure 39 show LLESC and LLESCsift (LLESC with SIFT) clustering results with different regularization parameter λ and number of clusters k on CMU-PIE and COIL data set. We can easily find LLESCsift performances slightly better than LLESC on COIL, as COIL data set contains images with different orientations and SIFT is better than Euclidean in finding similar images in those data sets. Finally, figuer Figure 40 shows our LLESC with MODL algorithm is more efficient than GraphSC in clustering on CMU-PIE and COIL data set.

TABLE XII

STATISTICS OF THE DATA SET

Data set	Size(N)	Dimensionality (M)	# of class (K)
CMU-PIE	1428	1024	68
COIL20	1440	1024	20

TABLE XIII

CLUSETERING PERFORMANCE ON CMU-PIE (K IS NUMBER OF CLUSTERS)

K	Normalized Mutual Information (%)				
	Kmeans	**PCA**	**KSVD**	**SC**	**LLESC**
4	33 ± 5.6	44 ± 6.2	100	100	100
12	52 ± 4.8	55 ± 5.1	91 ± 2.2	95 ± 1.2	97 ± 1.1
20	55 ± 3.3	59 ± 4.5	75 ± 2.6	91 ± 1.1	96 ± 1.3
28	59 ± 3.7	60 ± 3.4	76 ± 2.8	90 ± 1.2	96 ± 1.1
36	60 ± 3.9	63 ± 1.6	77 ± 3.1	88 ± 2.3	95 ± 1.2
44	60 ± 2.4	65 ± 1.1	74 ± 2.7	85 ± 1.5	95 ± 1.1
52	61 ± 2.2	62 ± 1.9	76 ± 2.2	83 ± 2.1	94 ± 1.3
60	65 ± 3.5	66 ± 2.1	78 ± 1.9	80 ± 1.4	94 ± 1.0
68	63	66	75	77	93

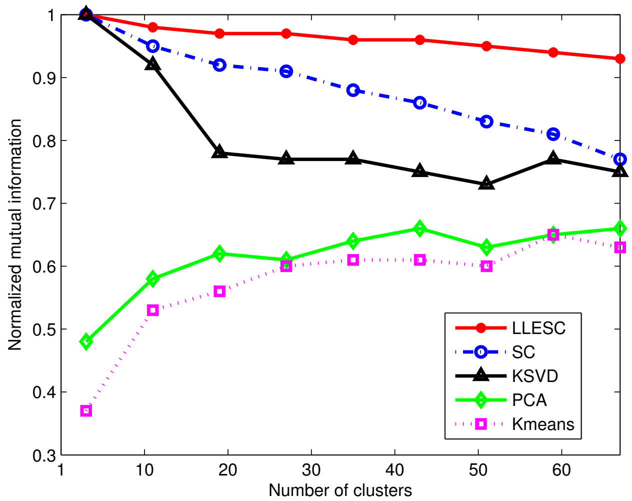


Figure 35. Normalized mutual information versus the number of clusters on CMU-PIE data set

TABLE XIV

CLUSETERING PERFORMANCE ON COIL20 (K IS NUMBER OF CLUSTERS)

	Normalized Mutual Information (%)				
K	**Kmeans**	**PCA**	**KSVD**	**SC**	**LLESC**
2	67 ± 8.5	54 ± 9.1	66 ± 8.8	81 ± 5.2	83 ± 9.6
4	65 ± 8.3	63 ± 9.2	64 ± 6.9	84 ± 6.3	84 ± 9.9
6	66 ± 9.4	59 ± 8.1	70 ± 8.1	78 ± 4.3	83 ± 9.2
8	61 ± 8.6	61 ± 9.7	79 ± 6.4	82 ± 5.2	79 ± 8.9
10	59 ± 9.6	60 ± 7.9	72 ± 5.5	84 ± 2.1	80 ± 8.6
12	62 ± 7.9	69 ± 6.5	70 ± 4.6	82 ± 2.4	81 ± 8.4
14	66 ± 7.7	65 ± 6.7	69 ± 5.1	76 ± 2.9	83 ± 6.3
16	71 ± 6.5	61 ± 5.5	72 ± 2.3	81 ± 3.3	82 ± 6.7
18	70 ± 4.4	60 ± 4.9	71 ± 1.4	76 ± 1.6	78 ± 5.9
20	72.4	66.7	74.1	77.3	80.3

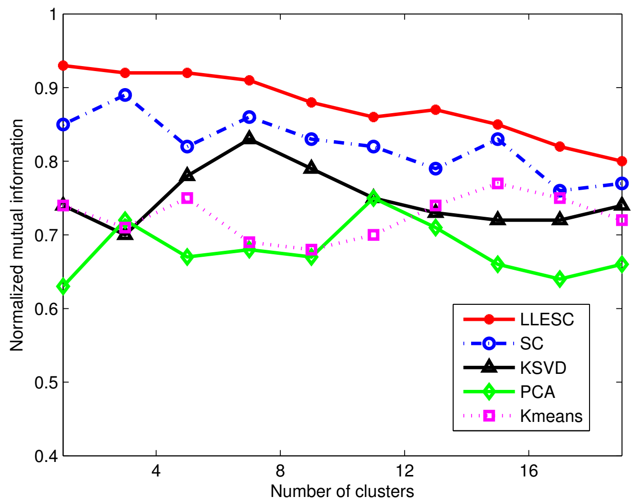


Figure 36. Normalized mutual information versus the number of clusters on COIL20 data set

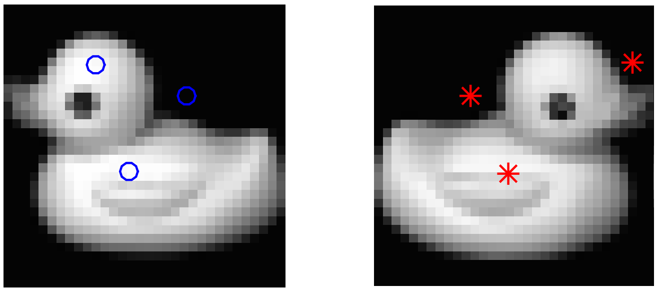
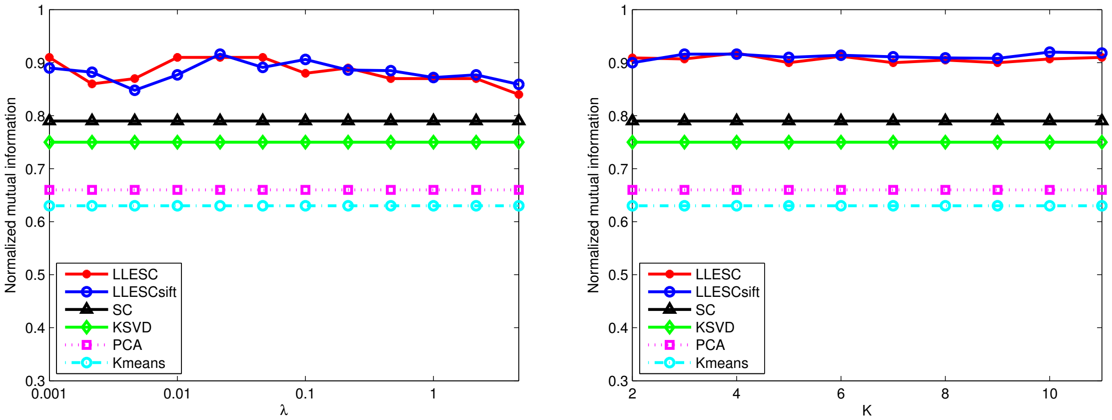


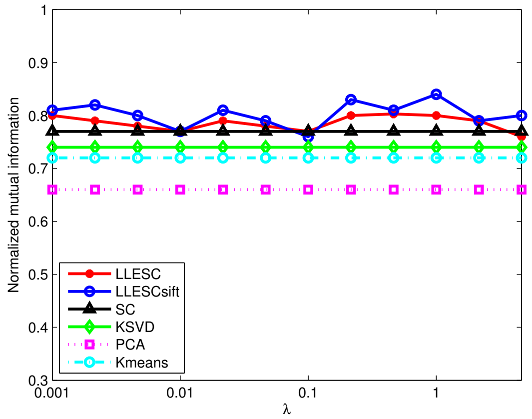
Figure 37. SIFT matching example



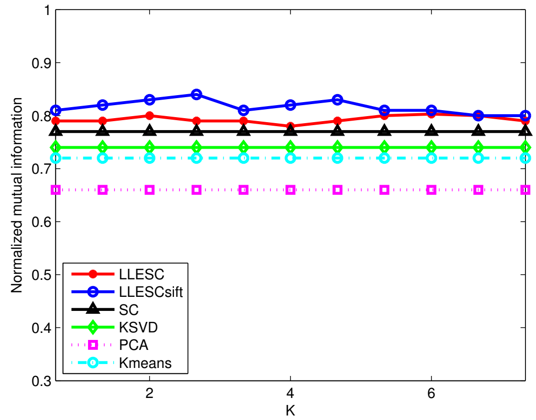
(a) NMI vs λ

(b) NMI vs k

Figure 38. Clustering performance with different values of regularization parameter (λ) and the number of nearest neighbors (k) on CMU-PIE face database.

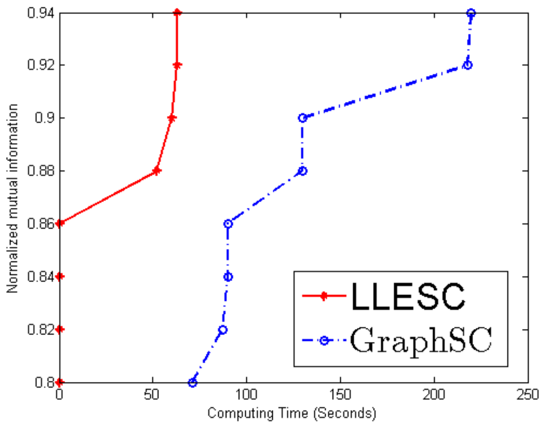


(a) NMI vs λ

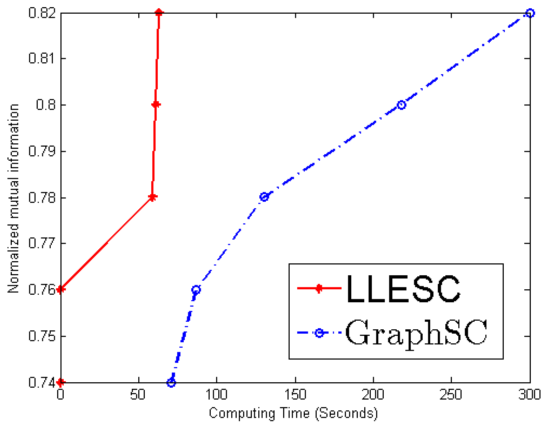


(b) NMI vs k

Figure 39. Clustering performance with different values of regularization parameter (λ) and the number of nearest neighbors (k) on COIL20 face database.



(a) On CMU-PIE data set



(b) On COIL20 data set

Figure 40. Clustering time between LLESC and GraphSC on CMU-PIE and COIL20 data set.

CHAPTER 6

CONCLUSION

This thesis summarized our research in generalized graph regularized sparse coding and fast tensor compressed sensing algorithms in image restoration, representation, medical image processing and tensor compressed sensing applications. It mainly consisted of a generalized graph regularized sparse coding method for image restoration and representation in Chpater 2, a Kronecker least angle regression algorithm for efficient tensor compressed sensing in Chapter 3, a graph regularized NMF algorithm for histology image color normalization in Chapter 4, and a locally linear embedded sparse coding algorithm for image representation in Chapter 5. We have discussed these methods and algorithms in detail and showed their performance with extensive computer simulation based experiments and compared with popular state-of-the-art algorithms.

Despite the good performance of proposed methods and algorithms. Many parts deserve further study to be more robust and mathematically proved. In Chapter 2, we proposed a generalized framework to combine graph Laplacian regularizer with sparse coding, which can be used for image restoration and representation. The proposed MCA-GSC algorithm outperforms state-of-the-art in terms of image denoising measured by PSNR, however, shy in SSIM evaluation. The deblurting performance is also comparable. In terms of image clustering and classification, the proposed dual graph regularized sparse coding method outperforms the popular clustering algorithm and slightly better in classification results. In Chapter 3, the proposed Kron-LARS and NBS-LARS algorithms are effective and efficient in 2D and 3D image data compressed sensing. They are efficient in both memory and computation, they outperforms the greedy algorithm Kron-OMP and NBOMP in terms of recovery PSNR versus sampling ratio. We still need to prove the robustness of the algorithms. In Chaper 4, our proposed GSNMF algorithm in histology image color normalization is effective and visually pleasing. It normalizes the image without destroying the texture and structure of the image. Besides qualitative results, quantative comparison is still pending. Last, in Chapter 5, we proposed locally linear embedded sparse coding algorithm for image representation, although efficient in testing dataset, we still lack of theratically guaranttee and extensive application in real-world datasets.

APPENDICES

Appendix A

LARS ALGORITHM (ALGORITHM 2)

We start with analysis of Alg. 2 with its recent optimized implementation presented in [128]. The LARS algorithm has input vector **y** $\in \mathbb{R}^{I^N}$ and an explicit dictionary **B** $\in \mathbb{R}^{I^N \times M^N}$ with $M > I$ (overcomplete dictionary).

Assuming that the algorithm takes K iterations/steps to converge:

Step 5: In this step, we calculate the correlation between dictionary columns and residual vector $\mathbf{B}^\mathsf{T}\mathbf{r}$, then take the maximum absolute value. It takes $(2(MI)^N + 2M^N)$ operations.

Step 7: By using the Cholesky factorization method, this step requires computing $(kI)^N + 3k^{2N}$ operations

Step 8: This step includes computing $\mathbf{B}(:, \mathcal{I})\mathbf{a}$ and subtracting it from **y** $(2(Ik)^N + I^N$ operations)

Appendix B

KRONECKER-LARS ALGORITHM (ALGORITHM 3)

Same as LARS case, we assume that the proposed Kron-LARS converges in K iterations/steps:

Step 5: In this step, we calculate correlation with $\underline{\mathbf{R}} \times_1 \mathbf{B}_1^T(:, i_1) \times_2 \mathbf{B}_2^T(:, i_2) \cdots \times_N \mathbf{B}_N^T(:, i_N)(2M^N + 2M^{N-1}I^2 + \cdots + 2MI^N = 2M^N I(\frac{1-(I/M)^N}{1-I/m}) operations)$ and take its maximum absolute-value. It takes $2M^N$ operations.

Step 7: This step computes $(\mathbf{c}_N \otimes \mathbf{c}_{N-1} \otimes \cdots \otimes \mathbf{c}_1)^T \mathbf{y}$ $(2I^N operations)$, $\mathbf{b} = (\overline{\mathbf{C}}_N^T \mathbf{c}_N) * (\overline{\mathbf{C}}_{N-1}^T \mathbf{c}_{N-1}) * \cdots * (\overline{\mathbf{C}}_1^T \mathbf{c}_1)$ $((2Nk^N I + Nk^N)$ operations), $\mathbf{d} = (\mathbf{Z}^{(k-1)})^{-1}\mathbf{b}$ ($2k^{2N}$ operations), $\mathbf{d}^T\mathbf{d}$ ($2k^N$ operations), updating the inverse matrix is $(2k^N + 3k^{2N})$, and computing the nonzero coefficients is $(2k^{2N})$. Thus, giving a total number of operations equals $2I^N + 2Nk^N I + (N+4)k^N + 7k^{2N}$.

Step 8: This step includes computing $(\mathbf{C}_N \odot \mathbf{C}_{N-1} \odot \cdots \odot \mathbf{C}_1)\mathbf{a}$ and subtracting it from $vec(\underline{\mathbf{Y}})$ $(N(N-1)I^N + I^N$ operations).

Appendix C

NBS-LARS ALGORITHM (ALGORITHM 4)

A distinctive characteristic of this algorithm compared to the previous ones is that, assuming that the algorithm is granted to obtain the true sparse representation, it will require much less iterations. More specifically, after the maximum correlated atom is detected in step 5, its position within the $(M \times M \times \cdots \times M)$ multiway array determines the indices $i_1^k, i_2^k, ..., i_N^k$ to be added to the current $\mathcal{I}_n$ $(n = 1, 2, ..., N)$ subsets. Some of these indices may already be included in the correspoding mode indices subsets. It is granted that at least one new index in one mode will be added. Thus, at every iteration, a situation between the following two extreme cases can happen: case 1: only one subset of indices $\mathcal{I}_n$ is incremented by 1 for some n; case 2: a new index is selected compexity for the worst case (case 2). We note that the minimum number of iterations is S (case 2) and the maximum number of iterations is NS (case 1):

Step 5: The same as for the Kronecker-LARS algorithm ($2M^N I + 2M^{N-1}I^2 + \cdots + 2MI^N = 2M^N I\left(\frac{1-(I/M)^N}{1-I/M}\right) + 2M^N$ operations).

Step 7: The update of the Cholesky facotrization for each mode, similar to the case of the algorithm 1, requires $2kI + k^2 + 2k$ operations and solving a set of N equations using again the Cholesky factorization, requires $2Nk^{N+1}$ operations. Thus, in the worst case (case 2) where the Cholesky factorizatio update is needed for every mode, the total number of operations is $2NkI + Nk^2 + 2Nk + 2Nk^{N+1}$.

Appendix C (Continued)

Step 8: This step includes computing $\underline{\mathbf{A}} \times_1 \mathbf{B}_1 \times_2 \mathbf{B}_2 \cdots \times_N \mathbf{B}_N$ ($2kI^N(\frac{1-(I/k)^N}{1-I/k}) \approx 2kI^N$ operations) and subtracting it from $\underline{\mathbf{Y}}$, giving us approximately $2kI^N + I^N$ operations where we assumed that $I \gg k$.

Appendix D

REPRINT PERMISSIONS

2017 IEEE. Reprinted, with permission, from [129] [Lingdao Sha, Dan Schonfeld : Dual graph regularized sparse coding for image representation. In 2017 IEEE Visual Communications and Image Processing (VCIP) , Dec 2017]. Permission shown as Fig. D.

2017 IEEE. Reprinted, with permission, from [130] [Lingdao Sha, Dan Schonfeld, Jing Wang: Locally linear embedded sparse coding for image representation. In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), March 2017]. Permission shown as Fig. D.

2017 SPIE. Reprinted, with permission, from [131] [Lingdao Sha, Dan Schonfeld, Amit Sethi: Color normalization of histology slides using graph regularized sparse nmf, Procceding SPIE, volume 10140, 2017]. Permission show as Fig. D.

Appendix D (Continued)

Copyright Clearance Center RightsLink® Home Create Account Help

IEEE Requesting permission to reuse content from an IEEE publication

Title: Dual graph regularized sparse coding for image representation

Conference Proceedings: 2017 IEEE Visual Communications and Image Processing (VCIP)

Author: Lingdao Sha

Publisher: IEEE

Date: Dec. 2017

LOGIN

If you're a copyright.com user, you can login to RightsLink using your copyright.com credentials.
Already **a RightsLink user** or want to learn more?

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

Perssion screenshot from IEEE VCIP

Appendix D (Continued)

Copyright Clearance Center RightsLink®

Home | Create Account | Help

IEEE
Requesting permission to reuse content from an IEEE publication

Title: Locally linear embedded sparse coding for image representation

Conference Proceedings: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)

Author: Lingdao Sha

Publisher: IEEE

Date: March 2017

Copyright © 2017, IEEE

LOGIN

If you're a copyright.com user, you can login to RightsLink using your copyright.com credentials.

Already **a RightsLink user** or want to learn more?

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

Perssion screenshot from IEEE ICASSP

Appendix D (Continued)

Paper reprint permission request (add to phd thesis) 收件箱 x

Lingdao Sha Hi, My name is Lingdao Sha, I want to request for permission of reprint my pu... 7月26日 (3天前)

Nicole Harris <nicoleh@spie.org> 7月27日 (2天前)
发送至 我

Dear Lingdao,

Thank you for seeking permission from SPIE to reprint material from our publications. As author, SPIE shares the copyright with you, so you retain the right to reproduce your paper in part or in whole.
Publisher's permission is hereby granted under the following conditions:
(1) the material to be used has appeared in our publication without credit or acknowledgment to another source; and
(2) you credit the original SPIE publication. Include the authors' names, title of paper, volume title, SPIE volume number, and year of publication in your credit statement.

Sincerely,

Nicole Harris
Administrative Editor, SPIE Publications
1000 20th St.
Bellingham, WA 98225
+1 360 685 5586 (office)
nicoleh@spie.org

SPIE is the international society for optics and photonics. http://SPIE.org

SPIE.

From: Lingdao Sha <shalingdao@gmail.com>
Sent: Thursday, July 26, 2018 8:36 PM
To: reprint_permission <reprint_permission@spie.org>
Subject: Paper reprint permission request (add to phd thesis)

...

Perssion screenshot from SPIE

CITED LITERATURE

1. Elad, M. and Aharon, M.: Image denoising via sparse and redundant representations over learned dictionaries. IEEE Transactions on Image Processing, 15(12):3736–3745, Dec 2006.

2. Dong, W., Shi, G., Ma, Y., Li, X., Mairal, C. J., Bach, F., Elad, M., Shi, G., Ma, Y., and Li, X.: Image restoration via simultaneous sparse coding: Where structured sparsity meets gaussian scale mixture, 2015.

3. Shan, Q., Jia, J., and Agarwala, A.: High-quality motion deblurring from a single image. ACM Trans. Graph., 27(3):73:1–73:10, August 2008.

4. Cai, D., He, X., and Han, J.: Document clustering using locality preserving indexing. IEEE Transactions on Knowledge and Data Engineering, 17(12):1624–1637, Dec 2005.

5. Caiafa, C. F. and Cichocki, A.: Computing sparse representations of multidimensional signals using kronecker bases. Neural Computation, 25(1):186–220, Jan 2013.

6. FOSTER, D. H., NASCIMENTO, S. M., and AMANO, K.: Information limits on neural identification of colored surfaces in natural scenes. Visual Neuroscience, 21(3):331336, 2004.

7. Hunt, B. R.: The application of constrained least squares estimation to image restoration by digital computer. IEEE Transactions on Computers, C-22(9):805–812, Sept 1973.

8. Lavin, P.: Restoration of a feature closed class of two-dimensional images. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-5(1):14–24, Jan 1983.

9. Cho, Z. H. and Burger, J. R.: Construction, restoration, and enhancement of 2 and 3-dimensional images. IEEE Transactions on Nuclear Science, 24(2):886–899, April 1977.

10. Youla, D. C. and Webb, H.: Image restoration by the method of convex projections: Part 1 #2014;theory. IEEE Transactions on Medical Imaging, 1(2):81–94, Oct 1982.

11. Sezan, M. I. and Stark, H.: Image restoration by the method of convex projections: Part 2-applications and numerical results. IEEE Transactions on Medical Imaging, 1(2):95–101, Oct 1982.

CITED LITERATURE (Continued)

12. Schutten, R. W. and Vermeij, G. F.: The approximation of image blur restoration filters by finite impulse responses. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-2(2):176–180, March 1980.

13. Philip, J.: Digital image and spectrum restoration by quadratic programming and by modified fourier transformation. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-1(4):385–399, Oct 1979.

14. Geman, S. and Geman, D.: Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-6(6):721–741, Nov 1984.

15. Abramatic, J. F. and Silverman, L. M.: Nonlinear restoration of noisy images. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-4(2):141–149, March 1982.

16. L. I. Rudin, S. O. and Fatemi, E.: Nonlinear total variation based noise removal algorithms. Phys. D, Nonlinear Phenomena, 60:259–268, Nov 1992.

17. Zhang, X. and Wu, X.: Image interpolation by adaptive 2-d autoregressive modeling and soft-decision estimation. IEEE Transactions on Image Processing, 17(6):887–896, June 2008.

18. Hu, W., Li, X., Cheung, G., and Au, O.: Depth map denoising using graph-based transform and group sparsity. In 2013 IEEE 15th International Workshop on Multimedia Signal Processing (MMSP), pages 001–006, Sept 2013.

19. Pang, J. and Cheung, G.: Graph laplacian regularization for image denoising: Analysis in the continuous domain. IEEE Transactions on Image Processing, 26(4):1770–1785, April 2017.

20. Mairal, J., Bach, F., Ponce, J., Sapiro, G., and Zisserman, A.: Non-local sparse models for image restoration. In 2009 IEEE 12th International Conference on Computer Vision, pages 2272–2279, Sept 2009.

21. Buades, A. and Coll, B.: A non-local algorithm for image denoising. In In CVPR, pages 60–65, 2005.

22. Hyvrinen, A.: Independent component analysis. Neural Computing Surveys, 2, 2001.

23. Aharon, M., Elad, M., and Bruckstein, A.: K-svd: Design of dictionaries for sparse representation. In IN: PROCEEDINGS OF SPARS05, pages 9–12, 2005.

CITED LITERATURE (Continued)

24. Meyer, Y.: Oscillating Patterns in Image Processing and Nonlinear Evolution Equations: The Fifteenth Dean Jacqueline B. Lewis Memorial Lectures. Boston, MA, USA, American Mathematical Society, 2001.

25. Balakrishnan, N., Hariharakrishnan, K., and Schonfeld, D.: A new image representation algorithm inspired by image submodality models, redundancy reduction, and learning in biological vision. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(9):1367–1378, Sept 2005.

26. Hyvrinen, A., Oja, E., Hoyer, P., and Hurri, J.: Image feature extraction by sparse coding and independent component analysis. In In Proc. Int. Conf. on Pattern Recognition (ICPR'98, pages 1268–1273, 1998.

27. B. A. Olshausen, D. J. F.: Sparse coding with an overcomplete basis set: a strategy employed by v1. Vision Research, 37:3311–3325, 1997.

28. Hoyer, P. O.: Non-negative matrix factorization with sparseness constraints. Jour. of, pages 1457–1469, 2004.

29. Joshua B. Tenenbaum, Vin de Silva, J. C. L.: A global geometric framework for nonlinear dimensionality reduction. Science, 290:2319–2323, 2000.

30. Saul, L. K. and Roweis, S. T.: An introduction to locally linear embedding. Technical report, 2000.

31. Belkin, M. and Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. In Advances in Neural Information Processing Systems 14, pages 585–591. MIT Press, 2001.

32. Mikhail, B. and Niyogi, P.: Towards a Theoretical Foundation for Laplacian-Based Manifold Methods, pages 486–500. Berlin, Heidelberg, Springer Berlin Heidelberg, 2005.

33. Cai, D., He, X., Han, J., and Huang, T. S.: Graph regularized non-negative matrix factorization for data representation. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 33(8):1548–1560, 2011.

34. Gao, S., hung Tsang, I. W., tien Chia, L., and Zhao, P.: Local features are not lonely laplacian sparse coding for image classification.

CITED LITERATURE (Continued)

35. Zheng, M., Bu, J., Chen, C., Wang, C., Zhang, L., Qiu, G., and Cai, D.: Graph regularized sparse coding for image representation. IEEE Transactions on Image Processing, pages 1327–1336, 2011.

36. Lee, D. D. and Seung, H. S.: Algorithms for non-negative matrix factorization. In In NIPS, pages 556–562. MIT Press, 2001.

37. Chen, S. S., Donoho, D. L., and Saunders, M. A.: Atomic decomposition by basis pursuit, 1995.

38. Gu, Q. and Zhou, J.: Co-clustering on manifolds. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '09, pages 359–368, New York, NY, USA, 2009. ACM.

39. Ding, C.: Orthogonal nonnegative matrix tri-factorizations for clustering. In In SIGKDD, pages 126–135. Press, 2006.

40. Sha, L., Schonfeld, D., and Wang, J.: Locally linear embedded sparse coding for image representation. In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 2527–2531, March 2017.

41. Sha, L., Schonfeld, D., and Wang, J.: Graph regularized sparse coding by modified online dictionary learning. IS and T International Symposium on Electronic Imaging Science and Technology, Part F130092:27–31, 1 2017.

42. Sha, L. and Schonfeld, D.: Dual graph regularized sparse coding for image representation. In 2017 IEEE Visual Communications and Image Processing (VCIP), pages 1–4, Dec 2017.

43. Starck, J. L., Elad, M., and Donoho, D. L.: Image decomposition via the combination of sparse representations and a variational approach. IEEE Transactions on Image Processing, 14(10):1570–1582, Oct 2005.

44. Do, M. N. and Vetterli, M.: The contourlet transform: an efficient directional multiresolution image representation. IEEE Transactions on Image Processing, 14(12):2091–2106, Dec 2005.

45. Starck, J.-L., Candes, E. J., and Donoho, D. L.: The curvelet transform for image denoising. IEEE TRANS. IMAGE PROCESS, 11(6):670–684, 2002.

46. Bandlet image estimation with model selection. Signal Processing, 91(12):2743 – 2753, 2011. Advances in Multirate Filter Bank Structures and Multiscale Representations.

CITED LITERATURE (Continued)

47. Donoho, D. L.: Wedgelets: nearly-minimax estimation of edges. Ann. Statist, pages 859–897, 1999.

48. Zhang, H., Wahba, G., Lin, Y., Voelker, M., Ferris, M., Klein, R., Klein, B., Zhang, H. H., Wahba, G., Lin, Y., Voelker, M., Ferris, M., Klein, R., and Klein, B.: Matching pursuit. Technical report, 1993.

49. Chen, S. S., Donoho, D. L., and Saunders, M. A.: Atomic decomposition by basis pursuit, 1995.

50. Andrews, D. F. and Mallows, C. L.: Scale mixtures of normal distributions. Journal of the Royal Statistical Society, 36(1):99–102, 1974.

51. Ting, D., Huang, L., and Jordan, M. I.: An analysis of the convergence of graph laplacians, 2011.

52. Hein, M.: Uniform convergence of adaptive graph-based regularization. In In Proceedings of the 19th Annual Conference on Learning Theory (COLT) (pp. 50 64. Springer, 2006.

53. Hein, M., yves Audibert, J., Luxburg, U. V., and Dasgupta, S.: Graph laplacians and their convergence on random neighborhood graphs. Journal of Machine Learning Research, page 2007, 2006.

54. Zhang, F. and Hancock, E. R.: Graph spectral image smoothing using the heat kernel. Pattern Recognition, 41(11):3328 – 3342, 2008.

55. Shi, J. and Malik, J.: Normalized cuts and image segmentation. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 22(8):888–905, 2000.

56. Weiss, Y.: Segmentation using eigenvectors: A unifying view. In In ICCV, 1999.

57. Belkin, M. and Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Comput., 15(6):1373–1396, June 2003.

58. Yin, M., Gao, J., and Lin, Z.: Laplacian regularized low-rank representation and its applications. IEEE Transactions on Pattern Analysis and Machine Intelligence, 38(3):504–517, March 2016.

59. Shuman, D. I., Narang, S. K., Frossard, P., Ortega, A., and V, P.: The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. IEEE Signal Process. Mag, pages 83–98, 2013.

CITED LITERATURE (Continued)

60. Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society, Series B, 58:267–288, 1994.

61. Boyd, S. and Vandenberghe, L.: Convex Optimization. New York, NY, USA, Cambridge University Press, 2004.

62. Takeda, H., Farsiu, S., and Milanfar, P.: Kernel regression for image processing and reconstruction. IEEE Transactions on Image Processing, 16(2):349–366, Feb 2007.

63. Jaynes, E. T.: Prior probabilities. IEEE Transactions on Systems Science and Cybernetics, 4:227–241, 1968.

64. Lee, H., Battle, A., Raina, R., and Ng, A. Y.: Efficient sparse coding algorithms. In In NIPS, pages 801–808. NIPS, 2007.

65. Bradley Efron, Trevor Hastie, I. J. R. T.: Least angle regression. Annals of Statistics, 32:407–451, june 2004.

66. Rosset, S. and Zhu, J.: Piecewise linear regularized solution paths. Ann. Statist, page 1030, 2007.

67. Karl Sjostrand, Line H. Clemmensen, R. L. B. E.: Spasm: A matlab toolbox for sparse statistical modeling. Journal of Statistical Software, 2010.

68. Mairal, J., Bach, F., Ponce, J., and Sapiro, G.: Online learning for matrix factorization and sparse coding. 2010.

69. Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K.: Image denoising by sparse 3-d transform-domain collaborative filtering. IEEE Transactions on Image Processing, 16(8):2080–2095, Aug 2007.

70. Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P.: Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing, 13(4):600–612, April 2004.

71. Sutour, C., Deledalle, C.-A., and Aujol, J.-F.: Estimation of the noise level function based on a nonparametric detection of homogeneous image regions. SIAM Journal on Imaging Sciences, 8(4):2622–2661, 2015.

72. Beck, A. and Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems, 2009.

CITED LITERATURE (Continued)

73. Danielyan, A., Katkovnik, V., and Egiazarian, K.: Bm3d frames and variational image deblurring. IEEE Transactions on Image Processing, 21(4):1715–1728, April 2012.

74. Chang, C.-C. and Lin, C.-J.: Libsvm: A library for support vector machines. ACM Trans. Intell. Syst. Technol., 2(3):27:1–27:27, May 2011.

75. Lu, Y. M. and Do, M. N.: A theory for sampling signals from a union of subspaces. IEEE Transactions on Signal Processing, 56(6):2334–2345, June 2008.

76. Elad, M., Figueiredo, M. A. T., and Ma, Y.: On the role of sparse and redundant representations in image processing. Proceedings of the IEEE, 98(6):972–982, June 2010.

77. Ahmed, N., Natarajan, T., and Rao, K. R.: Discrete cosine transform. IEEE Transactions on Computers, C-23(1):90–93, Jan 1974.

78. Bentley, P. M. and McDonnell, J. T. E.: Wavelet transforms: an introduction. Electronics Communication Engineering Journal, 6(4):175–186, Aug 1994.

79. Candes, E. J., Romberg, J., and Tao, T.: Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. IEEE Transactions on Information Theory, 52(2):489–509, Feb 2006.

80. Donoho, D. L.: Compressed sensing. IEEE Transactions on Information Theory, 52(4):1289–1306, April 2006.

81. Candes, E. J. and Tao, T.: Near-optimal signal recovery from random projections: Universal encoding strategies? IEEE Transactions on Information Theory, 52(12):5406–5425, Dec 2006.

82. Donoho, D. L. and Logan, B. F.: Signal recovery and the large sieve. SIAM J. Appl. Math., 52(2):577–591, April 1992.

83. Chen, S. S., Donoho, D. L., and Saunders, M. A.: Atomic decomposition by basis pursuit. SIAM Journal on Scientific Computing, 20(1):33–61, 1998.

84. Mallat, S. and Zhang, Z.: Adaptive time-frequency decomposition with matching pursuits. In [1992] Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis, pages 7–10, Oct 1992.

CITED LITERATURE (Continued)

85. Tropp, J. A.: Greed is good: algorithmic results for sparse approximation. IEEE Transactions on Information Theory, 50(10):2231–2242, Oct 2004.

86. Cohen, A., Dahmen, W., and Devore, R.: Compressed sensing and best k-term approximation. J. Amer. Math. Soc, pages 211–231, 2009.

87. Cands, E. J.: The restricted isometry property and its implications for compressed sensing. Comptes Rendus Mathematique, 346(9):589 – 592, 2008.

88. Donoho, D. L.: For most large underdetermined systems of linear equations the minimal $\ell_1$norm solution is also the sparsest solution. Communications on Pure and Applied Mathematics, 59(6):797–829.

89. Donoho, D. L. and Tsaig, Y.: Fast solution of ℓ_1 -norm minimization problems when the solution may be sparse. IEEE Transactions on Information Theory, 54(11):4789–4812, Nov 2008.

90. Multiarray signal processing: Tensor decomposition meets compressed sensing. Comptes Rendus Mcanique, 338(6):311 – 320, 2010.

91. Sidiropoulos, N. D. and Kyrillidis, A.: Multi-way compressed sensing for sparse low-rank tensors. IEEE Signal Processing Letters, 19(11):757–760, Nov 2012.

92. Duarte, M. F. and Baraniuk, R. G.: Kronecker compressive sensing. IEEE Transactions on Image Processing, 21(2):494–504, Feb 2012.

93. Friedland, S., Li, Q., and Schonfeld, D.: Compressive sensing of sparse tensors. IEEE Transactions on Image Processing, 23(10):4438–4447, Oct 2014.

94. Lathauwer, L. D., Moor, B. D., and Vandewalle, J.: A multilinear singular value decomposition. SIAM Journal on Matrix Analysis and Applications, 21(4):1253–1278, 2000.

95. Kolda, T. G. and Bader, B. W.: Tensor decompositions and applications. SIAM Rev., 51(3):455–500, August 2009.

96. Caiafa, C. F. and Cichocki, A.: Multidimensional compressed sensing and their applications. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 3(6):355–380.

97. Caiafa, C. F. and Cichocki, A.: Block sparse representations of tensors using kronecker bases. In 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 2709–2712, March 2012.

CITED LITERATURE (Continued)

98. Tucker, L. R.: Implications of factor analysis of three-way matrices for measurement of change. In Problems in measuring change., ed. C. W. Harris, pages 122–137. Madison WI, University of Wisconsin Press, 1963.

99. Eldar, Y. C., Kuppinger, P., and Bolcskei, H.: Block-sparse signals: Uncertainty relations and efficient recovery. IEEE Transactions on Signal Processing, 58(6):3042–3054, June 2010.

100. Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R.: Least angle regression. Annals of Statistics, 32:407–499, 2004.

101. Boyd, S. and Vandenberghe, L.: Convex Optimization. New York, NY, USA, Cambridge University Press, 2004.

102. Rivenson, Y. and Stern, A.: Practical compressive sensing of large images. In 2009 16th International Conference on Digital Signal Processing, pages 1–8, July 2009.

103. Ouellette, D. V.: Schur complements and statistics. Linear Algebra and its Applications, 36:187 – 295, 1981.

104. Donoho, D. L. and Elad, M.: Optimally sparse representation in general (non-orthogonal) dictionaries via minimization. In PROC. NATL ACAD. SCI. USA 100 2197202, 2002.

105. Jokar, S. and Mehrmann, V.: Sparse solutions to underdetermined kronecker product systems. Linear Algebra and its Applications, 431(12):2437 – 2447, 2009. Special Issue in honor of Shmuel Friedland.

106. Rivenson, Y. and Stern, A.: Compressed imaging with a separable sensing operator. IEEE Signal Processing Letters, 16(6):449–452, June 2009.

107. Basavanhally, A. N., Ganesan, S., Agner, S., Monaco, J. P., Feldman, M. D., Tomaszewski, J. E., Bhanot, G., and Madabhushi, A.: Computerized image-based detection and grading of lymphocytic infiltration in her2+ breast cancer histopathology. IEEE Transactions on Biomedical Engineering, 57(3):642–653, March 2010.

108. Sethi, A., Sha, L., Vahadane, A., Deaton, R., Kumar, N., Macias, V., and Gann, P.: Empirical comparison of color normalization methods for epithelial-stromal classification in h and e images. Journal of Pathology Informatics, 7(1):17, 2016.

109. Reinhard, E., Ashikhmin, M., Gooch, B., and Shirley, P.: Color transfer between images. IEEE Comput. Graph. Appl., (5):34–41, September 2001.

CITED LITERATURE (Continued)

110. Magee, D., Treanor, D., Crellin, D., Shires, M., Mohee, K., and Quirke, P.: Colour normalisation in digital histopathology images, 2009.

111. Macenko, M., Niethammer, M., Marron, J. S., Borland, D., Woosley, J. T., Guan, X., Schmitt, C., and Thomas, N. E.: A method for normalizing histology slides for quantitative analysis. ISBI'09, pages 1107–1110, Piscataway, NJ, USA, 2009. IEEE Press.

112. Khan, A. M., Rajpoot, N., Treanor, D., and Magee, D.: A nonlinear mapping approach to stain normalization in digital histopathology images using image-specific color deconvolution. IEEE Transactions on Biomedical Engineering, 61(6):1729–1738, June 2014.

113. Vahadane, A., Peng, T., Sethi, A., Albarqouni, S., Wang, L., Baust, M., Steiger, K., Schlitter, A. M., Esposito, I., and Navab, N.: Structure-preserving color normalization and s-parse stain separation for histological images. IEEE Transactions on Medical Imaging, 35(8):1962–1971, 2016.

114. Xu, J., Xiang, L., Wang, G., Ganesan, S., Feldman, M., Shih, N. N., Gilmore, H., and Madabhushi, A.: Sparse non-negative matrix factorization (snmf) based color unmixing for breast histopathological image analysis. Computerized Medical Imaging and Graphics, 46, Part 1:20 – 29, 2015. Sparsity Techniques in Medical Imaging.

115. Li, X. and Plataniotis, K. N.: A complete color normalization approach to histopathology images using color cues computed from saturation-weighted statistics. IEEE Transactions on Biomedical Engineering, 62(7):1862–1873, July 2015.

116. Ruifrok, A., J. D.: Quantication of histochemical staining by color deconvolution. Analytical & Quantitative Cytology & Histology, 2001.

117. Lee, D. D. and Seung, H. S.: Algorithms for non-negative matrix factorization. In In NIPS, pages 556–562. MIT Press, 2001.

118. Zhu, F., Wang, Y., Xiang, S., Fan, B., and Pan, C.: Structured sparse method for hyperspectral unmixing. Journal of Photogrammetry and Remote Sensing, 88:101 – 118, 2014.

119. Elad, M. and Aharon, M.: Image denoising via sparse and redundant representations over learned dictionaries. 15(12):3736–3745, 2006.

120. Mairal, J., Mairal, J., Elad, M., Elad, M., Sapiro, G., and Sapiro, G.: Sparse representation for color image restoration. In the IEEE Trans. on Image Processing, pages 53–69. ITIP, 2007.

CITED LITERATURE (Continued)

121. Grosse, R., Raina, R., Kwong, H., and Ng, A. Y.: Grosse et al. 149 shift-invariant sparse coding for audio classification.

122. Olshausen, B. A.: Sparse coding of time-varying natural images. In IN PROC. OF THE INT. CONF. ON INDEPENDENT COMPONENT ANALYSIS AND BLIND SOURCE SEPARATION, pages 603–608, 2000.

123. Yang, J., Yu, K., Gong, Y., and Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In in IEEE Conference on Computer Vision and Pattern Recognition(CVPR, 2009.

124. Bell, A. J. and Sejnowski, T. J.: The ”independent components” of natural scenes are edge filters, 1997.

125. Marelja, S.: Mathematical description of the responses of simple cortical cells. Journal of the Optical Society of America, 70:1297–1300, 1980.

126. Zou, H., Hastie, T., and Tibshirani, R.: Sparse principal component analysis. Journal of Computational and Graphical Statistics, 15:2006, 2004.

127. Lowe, D. G.: Distinctive image features from scale-invariant keypoints, 2003.

128. Sjstrand, K., Clemmensen, L., Larsen, R., Ersbll, B., and Einarsson, G.: Spasm-a matlab toolbox for sparse statistical modeling. 84, 01 2012.

129. Sha, L. and Schonfeld, D.: Dual graph regularized sparse coding for image representation. In 2017 IEEE Visual Communications and Image Processing (VCIP), pages 1–4, Dec 2017.

130. Sha, L., Schonfeld, D., and Wang, J.: Locally linear embedded sparse coding for image representation. In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 2527–2531, March 2017.

131. Lingdao Sha, Dan Schonfeld, A. S.: Color normalization of histology slides using graph regularized sparse nmf, 2017.

132. Chen, S. S., Donoho, D. L., and Saunders, M. A.: Atomic decomposition by basis pursuit. SIAM Rev, 43:129–159, jan 2001.

CITED LITERATURE (Continued)

133. Zisserman, J. M. F. B. . J. P. . G. S. . A.: Non-local sparse models for image restoration. In 2009 IEEE 12th International Conference on Computer Vision, pages 2272 – 2279, Kyoto, Sept 2009.

134. Shervashidze, N. and Bach, F.: Learning the structure for structured sparsity. IEEE Transactions on Signal Processing, 63:4894–4902, Sept 2015.

135. Yuan, M. and Lin, Y.: Model selection and estimation in regression with grouped variables. JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B, 68:49–67, Sept 2016.

136. Feiyun Zhu, Ying Wang, S. X. B. F. C. P.: Structured sparse method for hyperspectral unmixing. Computer Vision and Pattern Recognition, 2014.

137. Zou, H. and Hastie, T.: Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society, Series B, 67:301–320, 2005.

138. Mairal, J., Bach, F., Ponce, J., Sapiro, G., and Zisserman, A.: Discriminative learned dictionaries for local image analysis. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pages 1–8, June 2008.

139. Olshausen, B. A. and Fieldt, D. J.: Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature, 381:607609, 1996.

140. Ponce, J., Mairal, J., Bach, F., Ponce, J., Image, S. M., Processing, V., Hal, H. I., Mairal, J., Bach, F., and Suprieure, E. N.: Sparse modeling for image and vision processing, 2014.

141. Jenatton, R., yves Audibert, J., and Bach, F.: Structured variable selection with sparsity-inducing norms, 2011.

142. Gu, Q. and Zhou, J.: Gu, zhou: Neighborhood preserving nonnegative matrix factorization 1 neighborhood preserving nonnegative matrix factorization.

143. D.P.Bertsekas: Nonlinear Programming. Athena Scientific Belmont, 1999.

144. Kim, S. J., Koh, K., Lustig, M., Boyd, S., and Gorinevsky, D.: An interior point method for large scale l1 regularized least squares. IEEE Journal of Selected Topics in Signal Processing, 1(4), Dec 2007.

CITED LITERATURE (Continued)

145. Natarajan, B. K.: Sparse approximate solutions to linear systems. SIAM J. Comput., 24(2):227–234, Apr 1995.

146. Belkin, M., Niyogi, P., and Sindhwani, V.: Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. Technical report, JOURNAL OF MACHINE LEARNING RESEARCH, 2006.

147. Saul, L. K. and Roweis, S. T.: An introduction to locally linear embedding. Technical report, 2000.

148. Perona, P. and Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence, 12:629–639, 1990.

149. Shi, J. and Malik, J.: Normalized cuts and image segmentation. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 22(8):888–905, 2000.

150. Takeda, H., Farsiu, S., and Milanfar, P.: Kernel regression for image processing and reconstruction. IEEE TRANSACTIONS ON IMAGE PROCESSING, 16(2):349–366, 2007.

151. Portilla, J., Strela, V., Wainwright, M. J., and Simoncelli, E. P.: Image denoising using a scale mixture of gaussians in the wavelet domain. IEEE TRANS IMAGE PROCESSING, 2003.

152. Elmoataz, A., Lezoray, O., and Bougleux, S.: Nonlocal discrete regularization on weighted graphs: A framework for image and manifold processing. IEEE Transactions on Image Processing, 17(7):1047–1060, July 2008.

153. Weickert, J.: Anisotropic diffusion in image processing, 1998.

154. Sbastien Bougleux, Abderrahim Elmoataz, M. M.: Local and nonlocal discrete regularization on weighted graphs for image and mesh processing. Int J Comput Vis, 84:220236, 2009.

155. Cho, T. S., Zitnick, C. L., Joshi, N., Kang, S. B., Szeliski, R., and Freeman, W. T.: Image restoration by matching gradient distributions. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(4):683–694, April 2012.

156. Hyvrinen, A.: Sparse code shrinkage: Denoising of nongaussian data by maximum likelihood estimation, 1999.

157. Hoyer, P.: Independent component analysis in image denoising, 1999.

CITED LITERATURE (Continued)

158. Gilboa, G. and Osher, S.: Nonlocal linear image regularization and supervised segmentation. Multiscale Modeling & Simulation, 6(2):595–630, 2007.

159. Hyvrinen., A.: Fast ica matlab package. 2015.

160. Dong, W., Zhang, L., Shi, G., and Li, X.: Nonlocally centralized sparse representation for image restoration. IEEE Transactions on Image Processing, 22(4):1620–1630, April 2013.

161. Flucher, M.: ”compactness criteria” in variational problems with concentration. Springer, pages 35–42, 1999.

162. Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers, 2010.

163. Yin, M., Gao, J., Lin, Z., Shi, Q., and Guo, Y.: Dual graph regularized latent low-rank representation for subspace clustering. IEEE Transactions on Image Processing, 24(12):4918–4933, Dec 2015.

164. Feng, X. and Milanfar, P.: Multiscale principal components analysis for image local orientation estimation. In Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers, 2002., volume 1, pages 478–482 vol.1, Nov 2002.

165. Donoho, D. L. and Johnstone, I. M.: Minimax estimation via wavelet shrinkage. Technical report, 1992.

166. Pustelnik, N., Chaux, C., and Pesquet, J.-C.: Parallel proximal algorithm for image restoration using hybrid regularization. 20:2450 – 2462, 10 2011.

167. Image noise.

168. Kolda, T. G.: Multilinear operators for higher-order decompositions. 2006.

169. Wang, J. and Shim, B.: On the recovery limit of sparse signals using orthogonal matching pursuit. IEEE Transactions on Signal Processing, 60(9):4973–4976, Sept 2012.

170. Veta, M., Diest, P. J. V., Kornegoor, R., Huisman, A., Viergever, M. A., and W, J. P.: Automatic nuclei segmentation in h&e stained breast cancer histopathology images. PLoS ONE, page 70221, 2013.

VITA

Name	Lingdao Sha
Education	Ph.D., Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, Illinois 2018
	M.S., Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, Illinois 2017
	M.S., Statistics, University of Illinois at Chicago, Chicago, Illinois 2017
	B.S., Electrical and Computer Engineering, Beijing University of Posts and Telecommunications, Beijing, 2011
Experience	Computer Vision Scientist, Tempus lab, Chicago, IL, 10/2017- present
	Research Assistant, Multimedia Communications Laboratory, Dept. of ECE, University of Illinois at Chicago, 09/2011-05/2018
	Research Assistant, Research Histology and Tissue Imaging Lab, Dept. of Pathology, University of Illinois at Chicago, 05/2013-05/2017
	Teaching Assistant, Dept. of ECE, University of Illinois at Chicago, 08/2011-03/2013
Publications	L. Sha, D. Schonfeld, J. Wang, Graph Laplacian Regularization with Sparse Coding for Image Restoration and Representation, IEEE TCSVT, submitted.
	L. Sha, D. Schonfeld, J. Wang, Kronecker Least Angle Regression, IEEE TIP, submitted.
	A. Sethi, L. Sha, P. Gann, Computer vision detects subtle histological effects of dutasteride on benign prostate, BJU International, June 2018.

VITA (Continued)

A. Sethi, L. Sha, P. Gann, Empirical Comparison of Color Normalization Methods for Epithelial-Stromal Classification in H&E Images, Journal of Pathology Informatics 2016.

L. Sha, D. Schonfeld, Dual Graph Regularized Sparse Coding for Image Representation, VCIP 2017.

L. Sha, D. Schonfeld, J. Wang, Locally Linear Embedded Sparse Coding for Image Representation, ICASSP 2017

L. Sha, D. Schonfeld, J. Wang, Graph Regularized Sparse Coding by Modied Online Dictionary Learning, IS&T EI 2017 (Oral)

L. Sha, D. Schonfeld, A. Sethi, Color Normalization of Histology Slides using Graph Regularized Sparse NMF, SPIE Medical Imaging, 2017

L. Sha, D. Schonfeld, Q. Li, Parallax Mulit-Viewer Autostereoscopic Three-Dimensional Display, SPIE 2014.

AWARD	Wexler award	2011-2012
	Student Travel Award	2014, 2017