IMAGE ANNOTATION AND FEATURE ENGINEERING VIA STRUCTURAL SPARSITY AND LOW-RANK APPROXIMATION

by

DEGUANG KONG

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2013

To my parents and wife.

ACKNOWLEDGEMENTS

I would like to thank my supervising professor Dr. Chris H.Q. Ding for his constantly motivating and encouraging me, and also for his invaluable advice during my doctoral studies. As a world-leading data mining researcher, he gave detailed comments on my research, projects, code, and everything he could reach, which benefited me very much. He gave guidance on my different projects, which I have obtained fantastic research experiences. He is really nice and always reaches me with valuable insights. Without his help and instructions, I cannot imagine I can move forward to this step. I also feel very lucky to work with so many smart people in my groups, e.g., Feiping Nie, Dijun Luo, Hua Wang, Xiao Cai, Linbin Yu, Miao Zhang, etc, from whom I have learned a lot.

I also would like to thank my committee members, Dr. Heng Huang, Dr. Jeff Lei, Dr. Junzhou Huang, for their interests in my research and for taking time to serve in my dissertation committee. I am very grateful to all the professors who taught me in the past few years when I spent in different universities, both at China and U.S.A. Without their help and instructions, I cannot finish my degrees and thesis.

Finally, I would like to express my deep gratitude to my parents who have consistently encouraged me and inspired me during my undergraduate and graduate studies. They encourage me not to lose heart when I am faced with difficulties. I am very grateful to my wife for her patience and sacrifice. I also thank my friends who have helped me a lot in my study: we have a lot of fun while playing together!

November 15, 2013

ABSTRACT

IMAGE ANNOTATION AND FEATURE ENGINEERING VIA STRUCTURAL SPARSITY AND LOW-RANK APPROXIMATION

Deguang Kong, Ph.D.

The University of Texas at Arlington, 2013

Supervising Professor: Chris Ding

Nowadays, in order to sense environment and understand human behaviors, data analysis plays a more and more important role to handle heterogeneous data ranging from different domains, e.g., image categorization/annotation, customer segmentation, traffic prediction, ad optimization, recommendation systems, privacy analysis, etc. The large amount of multivariate data raises the fundamental problem of data mining: how to discover meaningful compact patterns hidden in the high-dimensional noisy observations? One approach is to do dimension reduction, which finds the low-dimensional subspace and thus encodes data in a low-dimensional structure. The other approach is to do feature selection or feature engineering, which manipulates the features to capture the most discriminant patterns for classification/clustering tasks.

The goal of this thesis is to develop new and efficient machine learning algorithms to solve many challenging problems appeared in image analysis, instead of simply application of existing methods to solve them. As compared to text mining, image analysis/mining is a more challenging task because image is more complicated to understand and analyze. The tremendous number of image data is available due to the advances in image acquisition and

storage techniques. However, current analysis and modeling techniques for image data are not mature, and still far behind.

In this thesis, to further improve the low-dimensional embedding results, an iteratively locally linear embedding algorithm is proposed, which captures the global structure of non-linear manifold through iteratively updating the embedding. To handle noisy data (e.g., data with missing values, corrupted values) classification problem, a robust data recovery model via Schatten-p norm is proposed to preprocessing the noisy data, where the rank of the data is implicitly decreased. To utilize the feature structure with constraints, an efficient feature learning algorithm via group lasso is proposed to handle features on arbitrary structure, whose convergence can be rigorously proved. To handle the problem of limited labeled data in image categorization/annotation tasks, efficient maximum consistency label propagation methods are proposed to improve the performance of graph-based semi-supervised learning methods, which utilizes both the labeled data information and graph manifold information. Extensive experiments indicate the good performance of proposed algorithms.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

Introduction

Nowadays, a large amount of data have been produced from different sources, e.g., text data, image data, sensor data, system data, etc. It is changing dramatically for people's daily life. This is also known as "data explosion" due to the rapid increase in the amount of published data and the effects of its abundance. These years have witnessed the data explosion, and data is everywhere. It is predicted that people will generate more data as humankind than it is generated in the previous 5,000 years in the next five years. How to process these huge amount of data becomes more important than ever before.

Text data (e.g., web logs, microlog, email, etc) is one important type of data. The goal of text analysis (a.k.a text mining) is to derive high-quality information from structured or unstructured text document. These high quality information are explicitly or implicitly expressed as patterns or trends. Typically, text mining is to find the relevance, novelty and interestingness from structured or unstructured data, which may involve text categorization, text clustering, concept extraction, sentiment analysis, document summarization, social tagging, etc. Lots of analysis and modeling technology has been proposed for text analysis, e.g., Bayesian model [1], Latent semantic indexing model [2], Latent Dirichlet allocation model [3], etc. Analysis and modeling techniques for text is relatively mature and adequate.

Among the large number of available data, image data accounts for a large portion of them. Advances in image acquisition and storage techniques have paved the way for tremendous growth of image data. For example, in Facebook, 350 million photos are uploaded every day. In google, weekly image search traffic is 1,400,000 per year on February 2013. As compared to text data, image data will reveal more useful information to hu-

man users. Image analysis and mining focuses on extraction of knowledge/patterns from images, which involves inter-disciplinary efforts in computer vision, image processing, machine learning, data mining, etc. Although some techniques (e.g., dictionary learning [4], sparse coding [5], deep learning [6]) have been proposed to solve the emerging image analysis problem, there is still a great gap to solve different challenging image analysis problems in real world. Analysis and modeling techniques for image data are not mature, and still far behind. For example, on google, when you search a similar image, the results will be very poor if the image is not in the google database. There is an urgent requirement to develop advanced machine learning techniques to solve image analysis problems (e.g., image classification, image segmentation, image analysis, etc).

The goal of this thesis is to develop new and efficient machine learning algorithms to solve many challenging problems appeared in image analysis. More specially, we focus on development of new technologies for image classifications: dimension reduction, feature engineering, low-rank data approximation and label propagation. We propose the following methods in this thesis.

- Iteratively locally linear embedding algorithm for dimension reduction;
- Efficient constrained feature selection algorithm via group lasso;
- Schatten-p norm model for robust data recovery;
- Efficient maximum consistency label propagation methods for semi-supervised learning and image annotation.

We start by briefly reviewing the key techniques used for feature engineering, dimension reduction and classification. We show applications of these techniques for image classification/annotation. Then we present some challenges of current feature engineering, dimension reduction methods, and show the problems of applications of those algorithms for image annotation tasks. Finally, we summarize the contributions of our works.

1.1 Motivating examples in image analysis

See a motivating example shown below. Given a vehicle image taken from a camera or a cell phone, can you find the near duplicate image and tell the exact information (company, year, model, etc.) of the query vehicle? To solve this problem, this involves image representation, image similarity measurement, and image classification. In order to achieve good image retrieval performance, there are a lot of challenging tasks needed to be solved.

See another motivating example shown below (Fig. 1.1). In practical settings, only a few number of images are labeled, and thus how to label the large number of unlabeled images remains a challenging issue. There is an urgent need of efficient and effective methods to label them. In Fig. 1.1, we only know several labeled cars, and the goal is to tell the labels for all the other cars. More challengingly, we need to tell the exact model of the car. Although the brand of the cars are the same: Honda, the model of the cars are totally different. This is known as "fine-grained" classification [7] in computer vision, which is a hot topic nowadays. As compared to coarse-grained classification, "fine-grained" classification problem is more challenging, because images from different sub-classes share lots of similar patterns in practice.

1.2 Review of feature Engineering, Dimension Reduction and Classification Techniques

Information explosion era has witnessed the rapid increase of the amount of data and the abundance of high-dimensional observations. The trend of "big data" presents enormous challenges in different applications. The availability of large amount of data will change everyone's daily life. How to handle with these data becomes the central problem of many applications, e.g., image categorization/annotation, customer segmentation, traffic prediction, ad optimization, recommendation systems, privacy analysis, etc. Data mining/machine learning plays a fundamental role to handle these data analysis tasks ap-

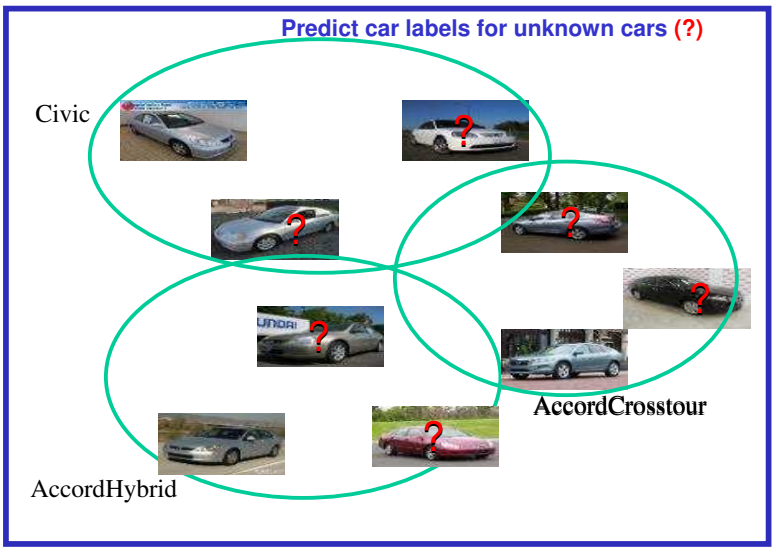


Figure 1.1: Prediction of labels for unknown cars (marked as red question mark). Few labeled cars: Civic, Accordcrosstour, AccordHybrid.

peared in different domains. Thus there is an urgent need to develop efficient and scalable algorithms to deal with data analysis problems through dimension reduction, feature engineering, semi-supervised learning. In this thesis, we will focus on some fundamental algorithms which may help to improve the capabilities of data analysis systems and applications. More specially, we will go through dimension reduction, feature engineering, and label propagation parts. The big picture of our work is shown in Fig 1.3.

Why feature engineering?

In machine learning, "feature" is a key concept used for different predictive tasks. See an example shown in Fig 1.4. In order to predict whether a patient get cancer or not, there are many factors which determine this. For example, he(or she) is of high pressure at work; he (or she) seldom take physical exercise; he (or she) smokes a lot, etc. Each factor of them can be viewed as a "variable" or "feature". Since the goal is to find cancer earlier before that they can grow and spread, we need to identify which factors are most significant to cause cancer according to different patients' record.

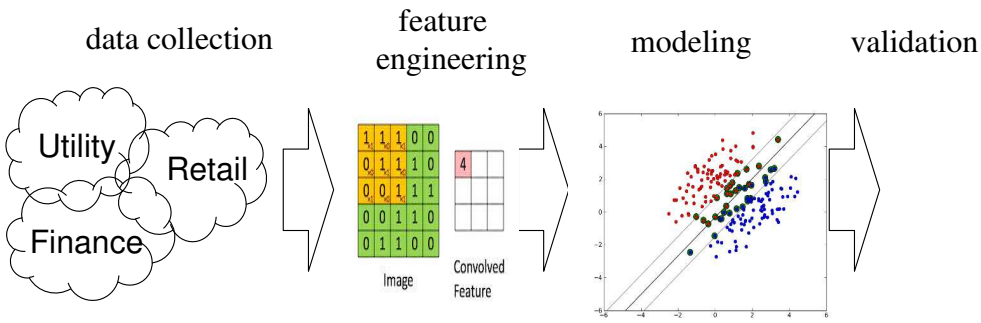


Figure 1.2: General data mining pipeline. (1) Collection of data from different sources; (2) Feature engineering on data; (3) Model construction according to requirement of data analysis tasks; (4) Validation and interpretation of experiment results.

Good input features are very important for different machine learning algorithms. For example, for a neural network (deep learning) module, the chosen features will affect the needed number layers, the number of hidden neurons, and the number of training examples. Feature engineering is to manipulate the features to satisfy the requirement of different classifiers, which may involve feature concatenation, feature selection, and feature configuration in different parameters, etc. Feature engineering is expected to help understand the properties and capabilities of different features, and identify which features are helpful for the tasks you are trying to solve. Domain knowledge can help for feature engineering experiment design, while the experiment results collected from designed models will motivate a better understanding of the problem.

It is well known that the initial pick of feature is an expression of prior knowledge. For example, for image data, we may use pixels, contours, and texture features; for signal, we may use samples, spectrograms, etc; for time series, we may use ticks, trends, reversals, etc; for biological data, we may use DNA sequence, marker sequence, genes, etc; for text data, we may use words, term frequency, inverted document frequency, grammatical classes and relations, etc. There are many methods to combine different features, e.g., polynomial combinations of features from different domains, logical conjunctions of features, com-

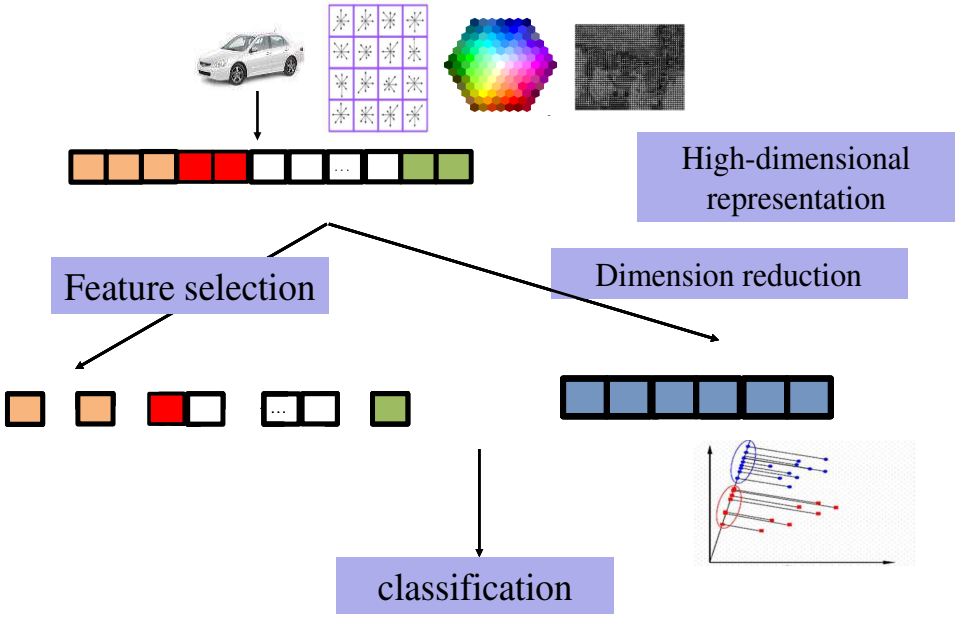


Figure 1.3: Feature selection, dimension reduction, and classification for image analysis.

bination of features in a tree-structure, etc. This will introduce large number of features. Kernel mapping on feature space generally leads to non-linear combinations of features. Feature selection on feature space selects a few number of important features. Different features may be strongly relevant, weekly relevant, or totally irrelevant. It is known that to find all relevant features for a classification problem is a NP-hard problem, because we need to do an exhaustive search through all subsets of features. Using filter method [8] for feature selection can select a few number of features, which are independent of classifiers. Feature selection using structural sparsity [9] based techniques has been proposed, for example, L_0 structural sparsity [10], L_1 structural sparsity [11], $L_{1/2}$ structural sparsity. The goal of these sparse based regularization is to seek relevant features which have non-zero values. This is also known as "compressed sensing" [12].

Wrapper method needs to use specific learning systems and algorithms. Backward and forward tracking algorithms are the two most popular wrapper method. For example, in backward feature selection method, it starts with all features, and then tries to remove

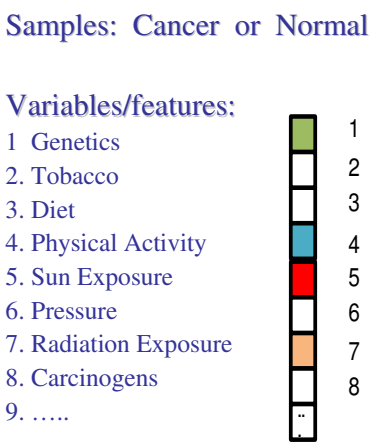


Figure 1.4: Features(variables) in prediction of cancer disease

each feature and measure its effect on validation set. The metric used to remove features is to find the features which causes the least harm. The above process is iterated, until the desirable number of features are selected. There are many variants of above feature selection methods. However, the computational cost is usually very high, and there is also risk of over-fitting on the validation set.

Greedy method is another popular feature selection method which selects features one by one. For example, in decision tree model, each decision making process can be viewed as a feature selection process. Pruning of decision tree is equivalent to pruning of features. In random forest model, ensembles of classifiers involves selection of new features. Boosting model/bagging model is used to combine different features.

Feature learning for image analysis has been widely used for handwritten digit recognition, face recognition, sonar image analysis, vehicle recognition, zip code recognition, etc. Multi-layer, multi-view [13], multi-task features [14], e.g., convolution on the features, connections of features, have been proposed for image analysis.

Why dimension reduction?

In many machine learning problems, the dimension of data is very large. For example, in face recognition problem, if pixel is used as features and the resolution of each image is 56x46 dimension, the dimension of each image will be 2576 after vectorization. Directly do classification/clustering on the high dimension space will not give accurate results. This is known as "curse of dimensionality". One popular way is do dimension reduction, which reduces the number of feature dimension via projection the original high-dimensional space into a low-dimensional space. Thus the goal of dimension reduction is to reduce the number of random variables and to avoid the effects of curse of dimensionality. In low-dimensional subspace, many standard classification methods (e.g., K-nearest neighbors, support vector machine, etc) can be applied.

Many linear dimension reduction methods are proposed, where dimension reduction can be conducted in one step using linear discriminant analysis (LDA), principal component analysis (PCA) [15], canonical correlation analysis (CCA) [16], factor analysis, etc. This results in low-dimensional feature embedding. The subspace computed by PCA captures the dimensions which have the maximum variability across all the data, reflected by the covariance structure of the data. Factor analysis also captures the correlation structure. LDA is to find the subspace which maximizes the margins between between-class distance and within-class distance.

Many nonlinear dimensionality reduction techniques are also proposed to project original data into non-linear space, using kernel PCA [17], locally linear embedding (LLE) [18], multidimensional scaling [19], Isomap (using geodesic distances in data space) [20], diffusion maps (using diffusion distances), semi-definite embedding, etc.

For example, Kernel PCA uses the non-linear mapping to find the low-dimensional latent structure in the kernel space. LLE is to capture the low-dimensional, neighborhood preserving embedding of high-dimensional data, using interpolation of data from its nearest neighbors. Laplacian embedding is to compute the low-dimensional embedding

through graph Laplacian using the reconstructed graph weight information. Metric multidimensional scaling (MDS) maps the original high dimensional space to a lower dimensional space, which preserves the pairwise distances. In semi-definite embedding [21], it proposes to learn the kernel matrix as an instance of semi-definite programming.

For multi-dimensional data, tensor analysis is a powerful tool for image analysis through multi-linear subspace learning, like Parafac model, Tensor decomposition [22], Tucker decomposition [23], bi-linear model, etc.

When handling very-high dimensional datasets (e.g., top 5 similar image in image database, top 5 similar video in video database), locality sensitive hashing (a.k.a random projection) [24] is also used to project original feature space into low-dimensional space for searching, query purpose. Recently, learning for hashing function becomes more popular because it provides a much better way to efficiently encode the high-dimensional data.

Why label propagation?

In image annotation/categorization, we can only obtain a few number of labeled data points. To label a large number of unlabeled data will be time-consuming and also labor expensive. In machine learning, this is usually solved using "semi-supervised learning (a.k.a transductive learning)" [25]. Different methods have been proposed to solve these problems. Among these methods, label-propagation [26] methods have proven to be very efficient and effective methods.

The goal of label propagation is to label data points whose labels are unknown. In the label propagation process, the labels are propagated from labeled data points to unlabeled data point according to certain propagation rules. Different label propagation methods have been proposed, such as Harmonic function [27], Local and Global consistency [28], Green's function [29], etc. In principle, the difference of these different algorithms is the different label propagation rules (a.k.a operator).

The key idea of graph-based label propagation method is to build a graph whose nodes represent labeled data and unlabeled data, and edges represent similiarity/relations between different data points. The labels are propagated from known nodes to all unknown nodes on graphs. During the label propagation process, different cost criterion has been proposed to solve the involved optimization problem, which essentially reduces to different graph transductive learning strategy.

In these different methods, graph structure, graph Laplacian is utilized to spread labels from labeled data samples to the whole dataset. A basic assumption about graph manifold learning is that, labeling process should be smooth over the graph manifold. This assumption yields graph regularization terms based on graph Laplacian. From inductive settings to transductive settings, graph manifold information is utilized to explore the local geometry data in graphs (e.g., k-NN graph).

In harmonic function method [30, 31], the label for an unknown node is given by a weighted average of the neighbors' current labels, which is a compromise of its initial labels and regularization term through graph Laplacian.

In local and global consistency method [32], in order to label unknown nodes, it uses the normalized graph weights to compute the contributions of other graph nodes, which leads to different "label spreading" criteria. In general, the time complexity of these algorithms is expected to be $O(kn^2)$, where k is the number of neighbors for a node in a graph, n is the number of nodes in a graph.

Intuitively, the label propagation on graphs has close connections with random walk on graphs [33]. For example, to assign a label to data point $\mathbf{x}_i$ is depending on the probability of arriving to a positively and negatively labeled example when making a random walk starting from $\mathbf{x}_i$ and until some labeled data is reached. This will give inductive rules for deriving label propagation equations on graphs.

It has been shown that label propagation on graphs is equivalent to an optimization problems by minimization of a cost function derived on graphs. Thus, it established the theoretical foundation for label propagation on graphs. Label propagation on graphs often leads to iteratively updating algorithms, which can be solved using linear system equation. Label propagation on graphs can also be interpreted from heat kernel view or electric network view [34]. The solution to the limit case of label propagation is known to be given by the voltage in an electric network, where labeled nodes are connected to voltage sources, and resistors correspond to the weights on graphs. Prior class distribution information can be incorporated into this label propagation procedure.

Above non-parametric local learning algorithm [35] essentially relies on a neighborhood graph which is used to approximate manifold which is very close to the data density. This means the label for data $\mathbf{x}$ is mainly depending on the unlabeled data which are close to $\mathbf{x}$ on graph manifold.

However, it is has been pointed out that above methods may not scale well, when the intrinsic dimension of these manifold becomes large [36]. For example, Laplacian regularization algorithm learns about the shape of manifold, which is reflected from the principal eigen-functions of the Laplacian of neighborhood [37]. The dimension of manifold, and whether data strictly lie on manifold determine the effect of the generalization error, which are the influences of neighborhood data points. The structure of manifold determines the performance of label propagation methods. If the manifold is flat in large region, then simply increase of the number of neighbors may be helpful for manifold learning. If the manifold has high curvature/variance, we cannot simply increase the number of neighbors, because it will increase bias, without significantly improvement of variance.

In real world, there are different categories of classification tasks, e.g., image categorization, customer segmentation, topic discovery. Fig.1.5 shows the image annotation

Figure 1.5: For image annotation task, each image is labeled as multiple concepts. Left image is labeled as: sky, mountain, sea, boat, sand; right image is labeled as: road, building, sky.

tasks, where the left image is labeled as: sky, mountain, sea, boat, sand; and the right image is labeled as: road, building, sky.

1.3 Research Challenges

There are tons of works about feature engineering, dimensional reduction and label propagation. However, understanding and improvement of above algorithms is not trivial. Application of above algorithms can be challenging, especially when considering some specific algorithms. We provide the challenges of above algorithms in the following in more detail. We argue that it is not trivial to solve the following problems.

Feature Engineering

Nowadays, structural sparsity based learning models are the most popular ways for feature learning and feature engineering. The advantage of structural sparsity model is that it can explore the structures of features, and enforce the solution of model to be sparse. Based on different properties of structural sparseness, Lasso [11], group lasso [38], exclusive lasso [39], generalized lasso [40], fused lasso [41] has been proposed to deal with multi-dimensional variables with different structures. Can we achieve better classification performance if feature structure is enforced? Is it closely connected to constraint feature selection? Can we provide some efficient algorithms to solve these constraint feature se-

lection problem? Is there some universal algorithms to handle the group lasso problem on arbitrary structure?

Dimension Reduction

Unsupervised dimension reduction is widely used in practice. Locally linear embedding is one of the most popular methods for non-linear feature embedding. The advantages of locally linear embedding algorithm is to explore the neighborhood information of data, and learn the embedding. The idea is to express each data $\mathbf{x}_i$ as a linear combinations of its neighbors, and then construct the embedding $\mathbf{y}_i$ so that they can be expressed as the same linear combinations of its neighbors. The challenge here is that, can we provide a method which can further improve the performance of embedding results? Like adaptive dimension reduction which combines dimension reduction and unsupervised learning (clustering) together, can we provide a method to improve the reduced embedding(subspace) adaptively?

Low rank model [42] is widely used for data recovery purpose. It has close connections with principal component analysis (PCA), factor analysis, latent semantic analysis, etc. Low rank approximation model is usually used to solve a minimization problem, which measures the cost function w.r.t the fit between given data and low rank data approximation matrix. The low dimensional constraint is enforced to ensure the rank of the recovered data is low. It can be widely used in recommendation systems [43], where the data matrix has missing values and approximation is categorial. In distance matrix completion problems, the constraint can also be enforced as semi-definite positive. The challenge is that, can we provide some efficient algorithms to further improve the performance of low rank data recovery? In practice, there are lots of missing values and corrupted values in data matrix, can we provide some methods to recovery the missing values for data classification purpose?

Label Propagation

Many graph-based label propagation [44] are proposed. However, it is not clear what is the difference among those different types of methods? Can we explore the connections and differences among different graph-based label propagation methods? If we can establish the connections between random walk and manifold learning, can it help to improve the performance of label propagation methods? Can it be applied to more graph-mining problems appeared in social networks and advertisement optimization?

In label propagation methods, we usually have a strong assumption that data lie on a low-dimension manifold. This manifold could be smooth, and also without high curvature. What is the true dimension of manifold? Can we determine the structure of manifold?

1.4 Contributions

To solve above challenging problems, in this thesis, we provide efficient algorithms targeting at specific challenges, to achieve the state-of-the-art performance in a number of tasks in dimension reduction, feature engineering, and label propagation.

- Locally Linear embedding (LLE) is one of the most popular dimension reduction method. We are interested in further improvement of standard LLE algorithm. We systematically improve the two main steps of LLE: (a) learning the graph weights $\mathbf{W}$, (b) learning the embedding $\mathbf{Y}$. We propose a sparse nonnegative $\mathbf{W}$ learning algorithm, and also a weighted formulation for learning embedding $\mathbf{Y}$. One interesting discovery is that, we find the embedding result is identical to normalized cuts spectral clustering. We further propose to iterate the two steps in LLE repeatedly to improve the results. Extensive experiment results show that iterative LLE algorithm significantly improves both classification and clustering results. It can be applied into image categorization/annotation tasks.

- Standard trace norm model is used for data recovery purpose. However, the reconstructed data can be shrank and singular values can be greatly suppressed. To solve this problem, we present two low-rank data recovery models through replacing the rank constraint by a Schatten p norm. The proposed model is attractive due to its suppression on the shrinkage of singular values at smaller p. The limitations of standard trace norm model are: the shrinkage of reconstructed data, the suppression of singular values. We analyze the optimal solution of model 1, and characterize the rank of optimal solution. We design two algorithms to solve model 2, one is based on Augmented Lagrangian method (ALM) [45], where a challenge step is to solve associated proximal operator. The other is based on an iterative re-weighted scheme, similar to reweighted L_2 scheme, where rigorous convergence analysis is provided. Extensive experiment results on 6 occluded datasets on computer vision tasks indicate good performance of proposed method.
- Feature structure information plays an important role for regression and classification tasks. We consider a more generic problem: group lasso problem, where structures over feature space can be represented as a combination of features in a group. These groups can be either overlapped or non-overlapped, which are specified in different structures, e.g., structures over a line, a tree, a graph or even a forest. We propose a new approach to solve this generic group lasso problem, where certain features are selected in a group, and an arbitrary family of subset is allowed. We employ accelerated proximal gradient method to solve this problem, where a key step is to solve the associated proximal operator. We propose an iterative re-weighted method to compute the proximal operator, where its convergence is rigorously proved. Experimental results on different structures (e.g., group, tree, graph structures) demonstrate the efficiency and effectiveness of the proposed algorithm.

- Random walk plays a significant role in computer science. The popular PageRank [46] algorithm uses random walk. Personalized random walks force random walk to "personalized views" of the graph according to users' preferences. In this paper, we show the close relations between different preferential random walks and label propagation methods used in semi-supervised learning. We further present a maximum consistency algorithm on these preferential random walk/label propagation methods [47] to ensure maximum consistency from labeled data to unlabeled data. Proposed algorithm restricts label propagation from source(labeled data) to reliably newly-labeled data only, and progressively expands to all unlabeled data, therefore ensuring maximum consistency from labeled data to unlabeled data. Extensive experimental results on 9 datasets provide performance comparisons of different preferential random walks/label propagation methods. They also indicate that our maximum consistency algorithm clearly improves the classification accuracy over existing methods. It can be applied to text mining, image categorization/annotation tasks.

1.5 Organization

This thesis is organized as follows.

In Chapter 2, we provide an improvement of standard Locally Linear embedding algorithm. We name it as "iteratively locally linear embedding algorithm", because it computes the embedding results iteratively. The contribution of this paper is to provide a new way to compute the embedding, which also establishes the relations between LLE and standard Laplacian embedding and spectral clustering. It can be applied into image categorization/annotation tasks, and also different clustering tasks. Material of this paper was published in two papers, presented at ICML2012 [48], SDM2013 [49].

In Chapter 3, we provide a new model for noisy data recovery, which can handle dataset with missing values and corrupted values. The proposed new model is attractive because it avoids the shrinkage of singular values. Efficient algorithms are derived to solve the proposed model. There are lots of applications using the proposed model, for example, application for recommendation systems, matrix completion problem, image recovery problems, etc. It can also be used a preprocessing step for image categorization/annotation tasks. Material of this paper was published in ECML2013 [50].

In Chapter 4, constraint feature learning is very helpful in practice because it can enforce structures over feature space. Group lasso is one effective way to enforce the constraint over feature space. There are many methods to enforce the group sparsity, whereas these different groups could be either overlapped or non-overlapped, and these different structures could be put over a line, a tree, a graph or even a forest. Iteratively re-weighted method is derived to solve the proposed problem, where its convergence is rigorously proved. There will be lots of applications which, e.g., medical image analysis, bio-marker analysis, benefit from proposed algorithms. Material of this paper was published in ICDM2013 [51].

In Chapter 5, random walk has many applications in information retrieval, medical image analysis, page rank, etc. The interesting discovery of this paper is to establish the relations between random walk and label propagation. This fits well for many semi-supervised learning problem. More interestingly, a maximum consistency algorithm is proposed to ensure the reliably propagation of labels from labeled data to unlabeled data. This is also known as “multi-stage” learning algorithm, which can be used for other classifiers, e.g., k-nearest neighbor classifier, support vector machine, green function, etc. Empirical study indicates the good performance of proposed algorithms. Material of this paper was published in ECML2012 [52].

Chapter 6 concludes the dissertation, summarizes our findings and empirical results, and discusses future research directions. We have other works related to above researches. Due to space limit, we will not put them in this thesis. If interested, please refer to [53, 54, 55, 56, 57, 58].

CHAPTER 2

An Iterative Locally Linear Embedding Algorithm

Locally Linear embedding (LLE) is a very popular dimension reduction method. This is very effective to do non-linear embedding. The two steps of LLE is shown in Fig.2.1 The goal of this section is to propose a method, which systematically improves the two main steps of LLE: (A) learning the graph weights $\mathbf{W}$, and (B) learning the embedding $\mathbf{Y}$.

We propose a sparse nonnegative $\mathbf{W}$ learning algorithm. We propose a weighted formulation for learning $\mathbf{Y}$ and show the results are identical to normalized cuts spectral clustering. We further propose to iterate the two steps in LLE repeatedly to improve the results. Extensive experiment results show that iterative LLE algorithm significantly improves both classification and clustering results.

2.1 Background of locally linear embedding

Recently, there have been many algorithms proposed for nonlinear dimension reduction, which include Isomap [59], locally linear embedding (LLE) [18], kernel-LLE [60], Hessian LLE [61], local tangent alignment [62], Laplacian embedding [63, 64], and many variations. Above dimension reduction algorithms usually cover two main steps: (A) for each data point, learn the local geometry information $\mathbf{W}$. This $\mathbf{W}$ can be viewed as similarity between data points or the edge weights of a graph whose nodes are the data points. We call this $\mathbf{W}$-learning, or learning the graph weights. (B) Using the learned $\mathbf{W}$ to embed the high-dimensional data points into a lower-dimensional space $\mathbf{Y}$. We call this $\mathbf{Y}$-learning, or

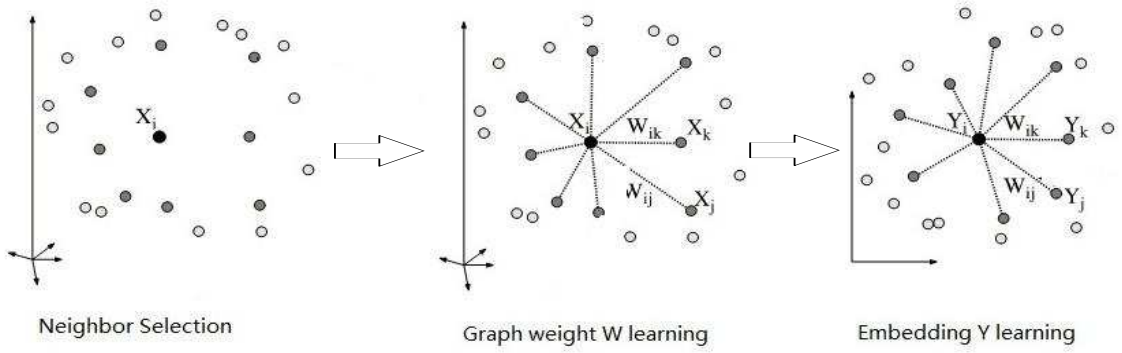


Figure 2.1: Procedure of locally linear embedding process. (1) Neighborhood selection; (2) Graph weight $\mathbf{W}$-learning; (3) Embedding $\mathbf{Y}$-learning.

learn the embedding. The performance of those algorithms are determined both by learning the local information and also by constructing the mapping relations.

In past decades, many clustering algorithms have been proposed such as K-means, spectral clustering and its variants [65], normalized cut [66], ratio cut [67], etc. Among them, the use of manifold information in graph cuts has shown the state-of-the-art clustering performance.

One key observation is that both LLE and spectral clustering utilize the data manifold information. This motivates us to investigate deeper relations between the LLE $\mathbf{Y}$-learning and spectral clustering in terms of Laplacian embedding (because the embedding is precisely the relaxed cluster indicators for the spectral clustering). Indeed, we discover that a properly modified formulation of $\mathbf{Y}$-learning provides a solution which is identical to the normalized Laplacian embedding (see §2.3). We incorporate this improvement into our final iterative LLE algorithm.

Another observation is that the data geometry information encoded in $\mathbf{W}$ also plays a central role in the performance of these algorithms. We investigate the $\mathbf{W}$-learning process and propose a nonnegative, kernelized, sparse $\mathbf{W}$-learning algorithm (see §2.4).

Furthermore, we propose to *iteratively repeat* the two mains steps ($\mathbf{W}$-learning and $\mathbf{Y}$-learning) to improve the results progressively. Here we use the learned embedding $\mathbf{Y}$ to augment the input data to learn a better $\mathbf{W}$, which leads to a better $\mathbf{Y}$ in turn. This is repeated until the process converges (details are given in §2.3). This iterative procedure incorporates both the improved $\mathbf{Y}$ learning and the improved $\mathbf{W}$ learning into a coherent iterative LLE algorithm.

The experiment results for clustering and semi-supervised learning tasks on 9 datasets show clear performance improvements.

2.2 LLE and New Formations

Brief overview of LLE

LLE [18] is a nonlinear dimension reduction approach. Suppose data $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n] \in \Re^{p \times n}$, consists of n data points $\mathbf{x}_i$, each with dimensionality p. LLE expects each data point and its neighbors to lie on or close to a locally linear manifold, which governs how the weight coefficients $\mathbf{W}$ are constructed from Eq.(2.1). It then reconstruct each data point (low k-dimensional embedding vectors $\{\mathbf{y}_i\}$) from its neighbors via the same neighborhood relations by minimizing a quadratic cost function Eq.(2.2),

$$\min_{\mathbf{W}} \sum_i ||\mathbf{x}_i - \sum_{j \in \mathcal{N}_i} W_{ij} \mathbf{x}_j||^2, \tag{2.1}$$

$$\min_{\mathbf{Y}} \sum_i \|\mathbf{y}_i - \sum_j W_{ij} \mathbf{y}_j\|^2, \tag{2.2}$$

where weight W_{ij} summarizes the contribution of the jth data point to the construction of ith data point. $\mathcal{N}_i$ is the kNN neighborhood of x_i. The shift invariance of $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n] \in \Re^{k \times n}$ is enforced by restricting $\sum_j W_{ij} = 1$.

LLE Improvements in two directions

In this paper, we propose improved formulations in both main steps in LLE. (A) In the $\mathbf{W}$-learning step of Eq.(1), we propose new improved formulations to learn $\mathbf{W}$. We first

make $\mathbf{W}$ nonnegative in this section. We will further propose a kernelized sparse learning in §4. (B) In the $\mathbf{Y}$-learning step of Eq.(2), we propose slightly modified formulation and prove that the solution to Eq.(2) is identical to Normalized Cut or Laplacian embedding. Our iterative LLE algorithm is based on these improved formations in both LLE steps.

To make a connection to graph embedding, we (1) restrict $\mathbf{W}$ to be nonnegative, i.e., we add constraint $\mathbf{W} \geq 0$ to Eq.(2.1) (as done in [68]); (2) we symmetrize $\mathbf{W}$ to obtain $\mathbf{Z} = \frac{1}{2}(\mathbf{W} + \mathbf{W}^T)$ as the graph edge weight/similarity matrix; (3) we impose $\mathbf{D}$-orthogonal constraint on $\mathbf{Y}$, i.e., $\mathbf{YDY}^T = \mathbf{I}$, where $\mathbf{D} = \mathrm{diag}(\mathbf{Z}e)$ is a diagonal matrix containing node degrees.

With these three changes, the LLE equations of Eqs.(2.1,2.2) become

$$\min_{\mathbf{W}} \sum_i ||\mathbf{x}_i - \sum_{j \in \mathcal{N}_i} W_{ij}\mathbf{x}_j||^2, \quad s.t. \quad \mathbf{W} \geq 0, \tag{2.3}$$

$$\min_{\mathbf{Y}} \sum_i d_i ||\mathbf{y}_i - \sum_j (\mathbf{D}^{-1}\mathbf{Z})_{ij}\mathbf{y}_j||^2 \; s.t. \; \mathbf{YDY}^T = \mathbf{I}, \tag{2.4}$$

where $d_i = D_{ii}$.

We note several important changes here. In Eq.(2.4), $\mathbf{D}^{-1}$ is inserted for two important reasons: (1) Note that $\sum_j (\mathbf{D}^{-1}\mathbf{Z})_{ij}\mathbf{y}_j$ is the average values of $\mathbf{y}_i$'s neighbors. Thus Eq.(2.4) enforces the smoothness of function $\{\mathbf{y}_i\}$. (2) It also enforces the shift invariance of obtained $\mathbf{Y}$, because $\sum_j (\mathbf{D}^{-1}\mathbf{Z})_{ij} = 1$. This implies that if $\{\mathbf{y}_i^*\}$ is an optimal solution, so is $\{\mathbf{y}_i^* - \mathbf{c}\}$ where $\mathbf{c}$ is a constant vector. Note that we add d_i as the weight of each point $\mathbf{y}_i$, for reasons immediately clear below.

LLE $\mathbf{Y}$-learning is identical to Normalized Cut Spectral Clustering

Now we show that LLE $\mathbf{Y}$-learning formulation of Eq.(2.4) is identical to normalized cut.

In fact, this is a general result, not restricted to LLE. It holds for any symmetric nonnegative graph similarity function $\mathbf{Z}$. More precisely we have theorem (1),

Theorem 1. *For any symmetric nonnegative graph similarity function* $\mathbf{Z}$ *of the formulation of Eq.(2.4), the optimal solution of* $\mathbf{Y}$ *is identical to the optimal solution* $\mathbf{H}$ *of normalized cut spectral clustering, given graph weight matrix* $\mathbf{Z}$*.*

In the following, we first briefly introduce normalized cut and present the proof of Theorem 1.

Review on Normalized Cut.

Normalized cut [66] is an effective graph partitioning (clustering) technique to identify clusters inherent in the data, given the pairwise similarity matrix $\mathbf{Z}$. It is well-known now multi-way normalized cut can be solved by the following problem,

$$\min_{\mathbf{G}} \mathrm{Tr}(\mathbf{G}^T(\mathbf{I}-\tilde{\mathbf{Z}})\mathbf{G}) \quad s.t. \quad \mathbf{G}^T\mathbf{G}=\mathbf{I}_k,$$

where $\tilde{\mathbf{Z}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{Z}\mathbf{D}^{-\frac{1}{2}}$. and $\mathbf{G} = [\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_k]$ are relaxed cluster indicators. The optimal solution for G is the smallest k eigenvectors from $(\mathbf{I}-\tilde{\mathbf{Z}})$, i.e.,

$$(\mathbf{I}-\tilde{\mathbf{Z}})\mathbf{g}_k = \mu_k \mathbf{g}_k. \tag{2.5}$$

The cluster indicator $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_k]$ is

$$\mathbf{h}_k = \mathbf{D}^{-\frac{1}{2}}\mathbf{g}_k, \qquad \mathbf{H} = \mathbf{D}^{-\frac{1}{2}}\mathbf{G}. \tag{2.6}$$

Relation to Laplacian Embedding.

It is easy to see that $\mathbf{H}^T = \mathbf{V} \equiv [\mathbf{v}_1, \cdots, \mathbf{v}_n]$ is identical to the solution of

$$\min_{\mathbf{V}} \sum_{ij} W_{ij}\|\mathbf{v}_i - \mathbf{v}_j\|^2 \quad s.t. \quad \mathbf{V}\mathbf{D}\mathbf{V}^T = \mathbf{I}_k. \tag{2.7}$$

This Laplacian embedding with degree normalization $\mathbf{V}\mathbf{D}\mathbf{V}^T = \mathbf{I}_k$ is effective for clustering problems because the embedding coordinates are the continuous relaxation of the cluster indicators of the multi-way normalized cut spectral clustering. Similarly, Laplacian embedding using coordinates with standard normalization $\mathbf{V}\mathbf{V}^T = \mathbf{I}_k$ is precisely the

continuous relaxation of the cluster indicators of multi-way ratio cut spectral clustering [67]; The widely used linear embedding, Principal component analysis (PCA) is precisely the continuous relaxation of the cluster indicators of the multi-way K-means clustering [69, 70].

Theorem 1 can be equivalently expressed for Laplacian embedding.

Proof of Theorem 1

To prove the theorem 1, we need Lemma 1.

Lemma 1. *The optimal solution to Eq.(2.4) is,*

$$\mathbf{Y}^* = \mathbf{F}^T\mathbf{D}^{-\frac{1}{2}}, \tag{2.8}$$

where $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_k] \in \Re^{n\times k}$ *is the smallest* k *eigenvectors of* $(\mathbf{I}-\tilde{\mathbf{Z}})^2$, $\tilde{\mathbf{Z}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{Z}\mathbf{D}^{-\frac{1}{2}}$, *i.e.,*

$$(\mathbf{I} - \tilde{\mathbf{Z}})^2\mathbf{f}_k = \lambda_k\mathbf{f}_k. \tag{2.9}$$

Proof of Lemma 1.

Proof. Note $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n] \in \Re^{k\times n}$. Let

$$\tilde{\mathbf{y}}_i = \mathbf{y}_i - \sum_j (\mathbf{D}^{-1}\mathbf{Z})_{ij}\mathbf{y}_j, \tag{2.10}$$

and then $\tilde{\mathbf{Y}} = [\tilde{\mathbf{y}}_1, \tilde{\mathbf{y}}_2, \cdots, \tilde{\mathbf{y}}_n] \in \Re^{k\times n}$. It is easy to see $\tilde{\mathbf{Y}} = \mathbf{Y} - \mathbf{Y}\mathbf{Z}\mathbf{D}^{-1}$. Now Eq.(2.4) can be written as

$$\begin{aligned}\sum_{i=1}^{n} d_i||\tilde{\mathbf{y}}_i||^2 = \sum_{i=1}^{n}\sum_{j=1}^{k} d_i\tilde{Y}_{ji}^2 = \sum_{j=1}^{k}\sum_{i=1}^{n}\tilde{Y}_{ji}D_{ii}(\tilde{Y}^T)_{ij}\\ = \text{Tr}(\tilde{\mathbf{Y}}\mathbf{D}\tilde{\mathbf{Y}}^T) = \text{Tr }(\mathbf{Y} - \mathbf{Y}\mathbf{Z}\mathbf{D}^{-1})\mathbf{D}(\mathbf{Y} - \mathbf{Y}\mathbf{Z}\mathbf{D}^{-1})^T\\ = \text{Tr }\mathbf{Y}(\mathbf{I} - \mathbf{Z}\mathbf{D}^{-1})\mathbf{D}^{\frac{1}{2}}\mathbf{D}^{\frac{1}{2}}[\mathbf{Y}(\mathbf{I} - \mathbf{Z}\mathbf{D}^{-1})]^T\\ = \text{Tr }\mathbf{Y}\mathbf{D}^{\frac{1}{2}}(\mathbf{I} - \tilde{\mathbf{Z}})(\mathbf{I} - \tilde{\mathbf{Z}})\mathbf{D}^{\frac{1}{2}}\mathbf{Y}^T.\end{aligned}$$

Thus Eq.(4) becomes

$$\min_{\mathbf{Y}} \text{ Tr}(\mathbf{Y}\mathbf{D}^{\frac{1}{2}}(\mathbf{I} - \tilde{\mathbf{Z}})^2\mathbf{D}^{\frac{1}{2}}\mathbf{Y}^T) \quad s.t. \ \mathbf{Y}\mathbf{D}\mathbf{Y}^T = \mathbf{I}. \tag{2.11}$$

To optimize Eq.(2.11) is equivalent to optimize,

$$\min_{\mathbf{F}} \mathrm{Tr}\mathbf{F}^T(\mathbf{I}-\tilde{\mathbf{Z}})^2\mathbf{F}, \quad s.t. \quad \mathbf{F}^T\mathbf{F}=\mathbf{I}, \tag{2.12}$$

where $\mathbf{F}=\mathbf{D}^{\frac{1}{2}}\mathbf{Y}^T$. It is easy to see the optimal solution $\mathbf{F}=[\mathbf{f}_1,\mathbf{f}_2,\cdots,\mathbf{f}_k]$ for Eq.(2.12) is the smallest k eigenvectors from $(\mathbf{I}-\tilde{\mathbf{Z}})^2$, i.e., Eq.(9). Thus the optimal solution $\mathbf{Y}^*=(\mathbf{D}^{-\frac{1}{2}}\mathbf{F})^T=\mathbf{F}^T\mathbf{D}^{-\frac{1}{2}}$. □

Proof of Theorem 1.

Proof. Because $(\mathbf{I}-\tilde{\mathbf{Z}})$ is semi-definite positive, the eigenvectors $\mathbf{g}_k$ of Eq.(2.5) can be uniquely mapped to eigenvectors $\mathbf{g}_k$ of

$$(\mathbf{I}-\tilde{\mathbf{Z}})^2\mathbf{g}_k=\mu_k^2\mathbf{g}_k. \tag{2.13}$$

Comparing Eq.(6) of normalized cut against Eq.(2.9) of LLE, one can see $\mathbf{f}_i=\mathbf{g}_i, \mu_i^2=\lambda_i, \mathbf{F}=\mathbf{G}$. Compared Eq.(2.6) of normalized cut against Eq.(2.8) of LLE, one can see $\mathbf{H}=\mathbf{Y}^T$. This completes the proof. □

2.3 An Iterative LLE Learning Algorithm (ILLE)

We now use the above results, coupled with two new schemes(A,B) to derive a new learning algorithm.

Motivation of iterative LLE

(A)Iterative process of LLE

In LLE, starting from $\mathbf{X}$, we learn $\mathbf{W}$, and then learn $\mathbf{Y}$ as the low-dimensional embedding of data $\mathbf{X}$. In this paper, we propose to use $\mathbf{Y}$ as the new data and iterate this process to further improve the embedding. The key observation is that the class structure of the data is more clear in $\mathbf{Y}$ than in $\mathbf{X}$ (this is the original embedding purpose of LLE). Thus we use $\mathbf{Y}$ as the new data and repeat this process to learn an improved $\mathbf{Y}$.

(B) Kernel generalization

From experiments on several datasets, the results of using linear formulation on $\mathbf{X}$ for learning $\mathbf{W}$ in Eq.(2.1) are generally not as good as other state-of-art methods. Here we use the kernel trick to generalize this to arbitrary nonlinear similarity function. We re-write Eq.(2.1) as

$$\min_{\mathbf{W}} \sum_i ||\phi(\mathbf{x}_i) - \sum_{j \in \mathcal{N}_i} \mathbf{W}_{ij} \phi(\mathbf{x}_i)||^2, \tag{2.14}$$

where $\phi(\mathbf{x}_i)$ is a mapping to a higher dimensional space. The important thing here is that the exact form of the mapping function is not needed; only the inner product $\mathbf{K}_{ij} = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$ is needed.

Using matrix notation, the LLE of Eq.(2.1) can be written as $\min_{\mathbf{W}} ||\mathbf{X} - \mathbf{X}\mathbf{W}^T||^2$, and Eq.(2.14) can be written as

$$||\phi(\mathbf{X}) - \phi(\mathbf{X})\mathbf{W}^T||^2 = \text{Tr}(\mathbf{K} - \mathbf{W}\mathbf{K} - \mathbf{K}\mathbf{W}^T + \mathbf{W}\mathbf{K}\mathbf{W}^T). \tag{2.15}$$

This is useful, because once we compute $\mathbf{Y}$ from Eq.(2.4), we can build a kernel from $\mathbf{Y}$ and substitute it into Eq.(2.15) to learn a new $\mathbf{W}$ (and thus $\mathbf{Z}$).

Proposed algorithm

By incorporating the above schemes of (A,B), we outline our iterative LLE learning algorithm as follows.

(1)Given kernel $\mathbf{K}^t$, solve for $\mathbf{W}^t$ with Eq.(2.15) or Eq.(2.17)[1].

(2)Given pairwise similarity $\mathbf{W}^t$, solve for $\mathbf{Y}^t$ using Lemma 1.

(3)Given embedding $\mathbf{Y}^t$, compute a new kernel $\mathbf{K}^{t+1}$ either as the final result of our algorithm (both embedding $\mathbf{Y}^t$ and kernel $\mathbf{K}^{t+1}$) or as input to step (1). Details of $\mathbf{K}^{t+1}$ construction is given in §3.3.

Initially $\mathbf{K}^1$ is obtained from data $\mathbf{X}$, we repeat above 3 steps for serval iterations to obtain a better kernel. See Algorithm 1 for more details. Note in step(1), we have two alternatives to compute $\mathbf{W}^t$. Thus we have two versions of iterative LLE - one based

[1] $\mathbf{S}$ of Eq.(2.17) can be viewed as pairwise similarity

on simply iterating LLE process, and the other based on learning a sparse kernel using algorithm of Eq.(3.17) in §4.

Discussion. Here we did not give the global convergence proof of this iterative LLE algorithm. The algorithm is very intuitive and natural. It is motivated by a simple observation: class structure is more clear in embedding $\mathbf{Y}$ than in original data $\mathbf{X}$.

Construction of the new kernel In step (3) of our algorithm, once the low-dimensional embedding $\mathbf{Y}^t$ is obtained, we have the following choices.

(a) Construct a new kernel from $\mathbf{Y}^t$. There are many way to construct kernel. One possible approach is to construct the kernel $\mathbf{K_Y}$ by simply using the Gaussian Kernel, i.e., $\mathbf{K_Y} = e^{-\gamma||\mathbf{y}_i - \mathbf{y}_j||^2}$, where γ is the scale parameter. Another way is to construct new kernel $\mathbf{K_Y}$ as the linear kernel in low-dimensional space, i.e., $\mathbf{K_Y} = \mathbf{Y}\mathbf{Y}^T$.

(b) Construct the kernel $\mathbf{K}^{t+1}$ either as the final result of our algorithm or as input to step (1). There are many choices, (b1)$\mathbf{K}^{t+1} = \mathbf{K}^t_\mathbf{Y}$; (b2) Kernel $\mathbf{K}^{t+1}$ is a combination of $\mathbf{K}^t_\mathbf{Y}$ and the previous kernel $\mathbf{K}^t$. There are two way to achieve this, additively $\mathbf{K}^{t+1} = \mathbf{K}^t + \mathbf{K}^t_\mathbf{Y}$, or multiplicatively $\mathbf{K}^{t+1} = \mathbf{K}^t \odot \mathbf{K}^t_\mathbf{Y}$, where we use $\odot$ to denote the element-wise matrix multiplication, e.g., if $\mathbf{C} = \mathbf{A} \odot \mathbf{B}$, then $C_{ij} = A_{ij} \times B_{ij}$.

In choice (b1), we simply ignore the previous kernel and set the new kernel $\mathbf{K}^{t+1} = \mathbf{K}^t_\mathbf{Y}$. Note both additive and multiplicative operations in choices (b2) ensure the new kernel $\mathbf{K}^{t+1}$ is also semi-definite positive(s.d.p) if the original kernel $\mathbf{K}^t$ is s.d.p.

Discussion. In our experiments, we tried different choices. We find the results obtained from (b2) are generally better than (b1), and the multiplicative combination usually achieves better results than additive combination. Thus in our experiment we use (b2) with multiplicative combination to construct the new kernel in step 3.

Algorithm 1 Iterative LLE algorithm(ILLE)

Input: Original Kernel $\mathbf{K}^1$ obtained from data $\mathbf{X}$, maximal iteration T

Output: Pairwise similarity $\mathbf{W}$, embedding $\mathbf{Y}$

Algorithm:

1: **for** $t = 1$ to T **do**

2: Compute $\mathbf{W}^t$ of Eq.(2.15) or Eq.(2.17) with current kernel $\mathbf{K}^t$

3: Compute $\mathbf{Z}^t = \frac{1}{2}(\mathbf{W} + \mathbf{W}^t)$.

4: Compute embedding $\mathbf{Y}^t$ using Lemma 1.

5: Compute a new kernel $\mathbf{K}^{t+1}$ given embedding $\mathbf{Y}^t$.

6: **end for**

7: **Output**: Pairwise similarity $\mathbf{W} = \mathbf{K}^{t+1}$, embedding $\mathbf{Y} = \mathbf{Y}^t$.

2.4 Improved $\mathbf{W}$-Learning Formulation

Here we propose an improvement to the $\mathbf{W}$-learning step of LLE. So far for LLE of Eq.(2.1) and the new kernel version of Eq.(2.15), we maintain the original LLE convention that $\mathbf{W}$ preserves the kNN structure, i.e. $W_{ij} \neq 0$ for only $j \in \mathcal{N}_i$ (kNN of object i).

This constraint is too strong for constructing the data similarity matrix $\mathbf{W}$. Thus, in our approach, we relax this to let $\mathbf{W}_{ij}$ be nonzero even if $j \notin \mathcal{N}_i$. In other words, we bypass kNN entirely.

We now present a new approach to learn the pairwise similarity matrix $\mathbf{S} \in \Re^{n \times n}$, where $\mathbf{S}_{ij}$ represents the i-th data's contribution to reconstruct data point x_j. We hope the newly learned $\mathbf{S}$ has much clear structure. We use the symbol $\mathbf{S}$ to emphasize that $\mathbf{W}$ is learned using the new approach. Our objective function for learning $\mathbf{S}$ is,

$$\min_{\mathbf{S} \geq 0} \|\mathbf{X} - \mathbf{X}\mathbf{S}\|^2 + \alpha \mathrm{Tr}(\mathbf{S}^T\mathbf{S}) + \beta ||\mathbf{S}||_{1,1}, \tag{2.16}$$

where α and β are regularization parameters, $||\mathbf{S}||_{1,1} = \sum_{ij} |\mathbf{S}_{ij}|$. The first term $\|\mathbf{X} - \mathbf{X}\mathbf{S}\|^2 = \sum_i ||\mathbf{x}_i - \sum_j \mathbf{S}_{ji}\mathbf{x}_j||^2$ is used to minimize the reconstruction error from the original

data. The second term penalizes the complexity of $\mathbf{S}$. The third term of L_1 norm is to promote the sparsity of the solution.

Using mapping ϕ: $\mathbf{X} \rightarrow \phi(\mathbf{X})$ to map data $\mathbf{X}$ to a higher dimensional space in kernel machine. Eq.(2.16) becomes

$$\min_{\mathbf{S}\geq 0} \|\phi(\mathbf{X}) - \phi(\mathbf{X})\mathbf{S}\|^2 + \alpha \text{Tr}(\mathbf{S}^T\mathbf{S}) + \beta||\mathbf{S}||_{1,1}, \tag{2.17}$$

which is equivalent to,

$$\min_{\mathbf{S}\geq 0} \text{Tr}(\mathbf{K} - 2\mathbf{K}\mathbf{S} + \mathbf{S}^T\mathbf{K}\mathbf{S}) + \alpha \text{Tr}(\mathbf{S}^T\mathbf{S}) + \beta||\mathbf{S}||_{1,1}. \tag{2.18}$$

Eq.(2.18) is identical to Eq.(2.16) when $\mathbf{K} = \mathbf{X}^T\mathbf{X}$.

Eq.(2.18) is a convex optimization problem and $\mathbf{S}$ has a unique global solution. Furthermore, Eq.(2.18) can be written as

$$\min_{\mathbf{S}\geq 0} \text{Tr}[\mathbf{K} + (\beta\mathbf{E} - 2\mathbf{K})\mathbf{S} + \mathbf{S}^T(\mathbf{K} + \alpha\mathbf{I})\mathbf{S}], \tag{2.19}$$

where $\mathbf{E}$ is a matrix of all ones. Because $\mathbf{K}$ is s.d.p., by adding $\alpha\mathbf{I}$ with $\alpha > 0$, $(\mathbf{K} + \alpha\mathbf{I})$ is a well-conditioned matrix. It can be solved efficiently (see below). Usually L_1 norm term is difficult to handle. Here, however, it does not add any difficulty when handled together with the nonnegativity constraint. The L_1 term can be ignored entirely: $||\mathbf{S}||_{1,1} = \text{Tr}(\mathbf{E}\mathbf{S})$.

Computational algorithm for Eq.(2.17)

Here we present an efficient algorithm to solve Eq.(2.17) and prove its convergence rigorously.

The algorithm starts with an initial guess of $\mathbf{S} = \mathbf{E}$($\mathbf{E}$ is a matrix of all ones), iteratively updates $\mathbf{S}$ according to

$$\mathbf{S}_{ij} \leftarrow \mathbf{S}_{ij} \frac{\mathbf{K}_{ij}}{(\mathbf{K}\mathbf{S} + \alpha\mathbf{S})_{ij} + \frac{\beta}{2}}. \tag{2.20}$$

This algorithm converges very fast. The computational algorithm for Eq.(2.17) is very simple and can be efficiently implemented.

Convergence of Updating rule of Eq.(3.17) We have Theorem(2) to prove the convergence of the algorithm when $\mathbf{K}$ is non-negative.

Theorem 2. *Updating* $\mathbf{S}$ *using the rule of Eq.(3.17), the objective function of Eq.(2.17) monotonically decreases.*

The proof of this theorem is lengthy and is similar to that in [71, 56]. We therefore skip the proof in this paper.

Correctness of Updating Rule of Eq.(3.17)

We prove that the converged solution satisfies the Karush-Kuhn-Tucker condition of the constrained optimization theory. We have Theorem 3 to prove it.

Theorem 3. *At convergence, the converged solution* $\mathbf{S}$ *of the updating rule of Eq.(3.17) satisfies the KKT condition of the optimization theory.*

Proof. The KKT condition for $\mathbf{S}$ with constraints $\mathbf{S}_{ij} \geq 0$ is $\frac{\partial J(\mathbf{S})}{\partial \mathbf{S}_{ij}} \mathbf{S}_{ij} = 0, \ \forall \, i, j$.

The derivative of $J(\mathbf{S})$(Eq.2.17) is $\frac{\partial J(\mathbf{S})}{\partial \mathbf{S}_{ij}} = (-2\mathbf{K} + 2\mathbf{K}\mathbf{S} + 2\alpha\mathbf{S} + \beta\mathbf{E})_{ij}$. Thus the KKT condition for $\mathbf{S}$ is

$$(-2\mathbf{K} + 2\mathbf{K}\mathbf{S} + 2\alpha\mathbf{S} + \beta\mathbf{E})_{ij}\mathbf{S}_{ij} = 0 \quad \forall \, i, j. \tag{2.21}$$

On the other hand, once $\mathbf{S}$ converges, according to the updating rule of Eq.(3.17), the converged solution $\mathbf{S}$ satisfies

$$\mathbf{S}_{ij} = \mathbf{S}_{ij} \frac{\mathbf{K}_{ij}}{(\mathbf{K}\mathbf{S} + \alpha\mathbf{S} + \frac{\beta}{2}\mathbf{E})_{ij}}, \tag{2.22}$$

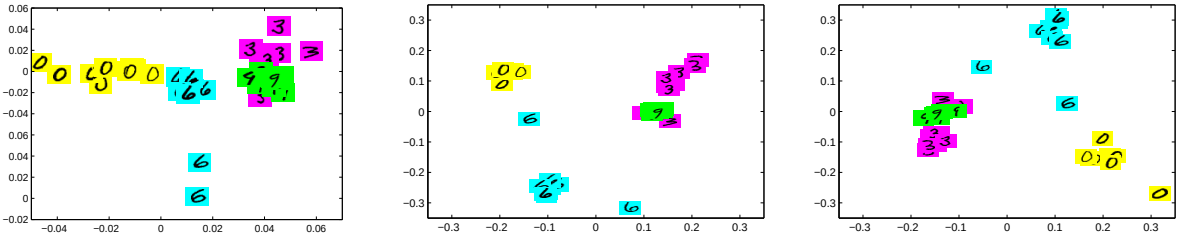
which can be written as $[-\mathbf{K}_{ij} + (\mathbf{K}\mathbf{S} + \alpha\mathbf{S} + \frac{\beta}{2}\mathbf{E})_{ij}]\mathbf{S}_{ij} = 0$. This is identical to Eq.(2.21). Thus the converged solution satisfies the KKT condition. □

2.5 Experiments

We perform the proposed algorithms on nie datasets. We do both semi-supervised learning and clustering on these datasets. We evaluate the proposed iterative LLE learning

Table 2.1: Dataset descriptions.

Dataset	#Size	#Dimension	#Class
AT&T	400	644	40
Mnist	150	784	10
Umist	360	644	20
Binalpha	1014	320	36
Yale	1984	2016	31
Caltec	600	432	20
MSRC	210	432	7
Newsgroup	499	500	5
Reuters	900	1000	10



(a) embedding result from $\mathbf{K}_0$ (b) embedding result from LLE1 (c) embedding result from LLE4

Figure 2.2: 2D visualizations of embedding results using (1) initial/input kernel $\mathbf{K}_0$; (2) LLE1: results on learned $\mathbf{Y}$ after 1 LLE iteration; (3) LLE4: results on learned $\mathbf{Y}$ after 4 LLE iterations; using 4 digits "0","3","6","9" on MNIST dataset

algorithm(§3) and sparse similarity learning algorithm(§4), and then show the embedding results from our approach.

Dataset. These data sets come from a wide range of domains, including three face datasets AT&T, umist and yale [72], two digit datasets mnist [73] and binalpha [1], two image scene datasets Caltec101(Caltec) [74] and MSRC [75], and two text datasets Newsgroup[2], Reuters[3]. Table 2.1 summarizes the characteristics of them.

We show both the iterative LLE (algorithm 1 in §2.3) and the sparse similarity learning algorithm (§2.4) results. Given original kernel $\mathbf{K}^0$, $\mathbf{S}$ is obtained from 1-time running

[1]http://www.kyb.tuebingen.mpg.de/ssl-book/ benchmarks.html

[2]http://people.csail.mit.edu/jrennie/20Newsgroups/

[3]http://www.daviddlewis.com/resources/testcollections/reuters21578/

of sparse similarity learning algorithm in §2.4. Then we obtain the final embedding results $\mathbf{Y}$ after repeating out iterative LLE algorithm for 4 times. For step 1 of algorithm 1(§2.3), we use kernel constructed from Eq.(2.17) for the subsequent iterations. For step 3 of algorithm 1(§2.3), given current embedding $\mathbf{Y}^t$, we obtain the new kernel $\mathbf{K}^{t+1}$ using choice (b2) with multiplicative combination in every iteration.

Clustering Results We use clustering algorithms to evaluate the learned $\mathbf{Y}$ in LLE. We compare three standard clustering algorithms: (1) normalized cut, which in the context of our iterative LLE, is simply K-means clustering on learned embedding $\mathbf{Y}$; (2) spectral clustering [65], which is K-means clustering on embedding $\mathbf{Y}$ normalized onto unit sphere. (3) symmetric NMF, which runs on the learned $\mathbf{W}$ in iterative LLE. All of results are the averages of 10 K-means clustering with random starts.

We use accuracy, normalized mutual information (NMI) and purity as the measurement of the clustering qualities and the results are shown in Table 2.2. We show the clustering results obtained from using (1) the original/input kernel ($\mathbf{K}^0$), (2) LLE1: results on learned $\mathbf{Y}$ after 1 LLE iteration. (3) LLE4: results on learned $\mathbf{Y}$ after 4 LLE iterations.

For image datasets, we use gaussian kernel $\mathbf{K}^0_{ij} = e^{-\gamma||\mathbf{x}_i - \mathbf{x}_j||^2}$. For text datasets, we use linear kernel. We tune the graph construction parameter γ to obtain the best results from kernel $\mathbf{K}^0$. From Table 2, we observe that LLE1 and LLE4 consistently achieve better clustering results, as compared to the results obtained from original kernel $\mathbf{K}^0$.

Semi-supervised learning results

We use $\mathbf{K}^0$, LLE1 and LLE4 results (learned $\mathbf{W}$) as the input to run three semi-supervised methods: harmonic function[76], local and global consistency[32], green's function[77]. We compare the classification accuracy of above three methods by using original kernel($\mathbf{K}^0$) and the results obtained from LLE1 and LLE4 on 9 data sets. For all the methods and datasets, we randomly select 10%, 20% of labeled data for each class, and use the rest as unlabeled data. We do 10 fold and 5 fold cross validation, respectively.

Finally, we report the average of the semi-supervised classification accuracy in Table 2.3. In all cases, we obtain higher classification accuracy by applying iterative LLE learning algorithm (shown as LLE4 and LLE1).

Demonstration of embedding results We demonstrate the advantages of iterative LLE learning algorithm (§2.3) and sparse similarity learning algorithm (§2.4) using two-dimensional visualization. We randomly select four digits from MNIST dataset ("0", "3", "6", "9"). Given Gaussian Kernel as the input, the iterative LLE algorithm (§2.3) and sparse similarity learning algorithm (§2.4) are run. The other parameters are set as mentioned before. The embedding results obtained from original Gaussian Kernel $\mathbf{K}_0$, 4-time running of iterative LLE learning algorithm (LLE4) and 1-time running of $\mathbf{W}$-learning algorithm (LLE1) are shown in Figs.(2.2a, 2.2b, 2.2c). In original results from Gaussian Kernel, all images from different groups collapse together. For the results obtained from LLE4 and LLE1, the images from different groups are balanced and distributed more evenly. This indicates much better embedding results.

Insights from experiment results.

Overall, from initial/input kernel $\mathbf{K}^0$ to LLE1, LLE4, both clustering and semi-supervised learning results consistently improved. Comparing results obtained between LLE1 and initial/input kernel $\mathbf{K}^0$, the performance boost is from the learned $\mathbf{W}$ using the algorithm of §2.4. Comparing results obtained between LLE1 and LLE4, the performance boost is from the iterative learning of LLE. From the statistics shown in Tables 2.2, 2.3, we observe that the boost from LLE1 to LLE4 is usually higher than that from $\mathbf{K}^0$ to LLE1, indicating that the iterative aspect contributes more.

2.6 Lessons learned

In summary, the main contribution of this section is in three-fold. (1) We show that an improved $\mathbf{Y}$-learning formulation of LLE is identical to normalized cut spectral clustering. (2) We present an improved $\mathbf{W}$-learning algorithm that learns a nonnegative, sparse pairwise similarity from an input kernel function. (3) An iterative procedure of the above two steps is proposed to progressively refine/improve the solution. Experiments show that the iterative LLE incorporating (1,2,3) leads to better clustering and semi-supervised learning results.

In the future work, we will investigate the de-noising power of proposed iterative LLE algorithm. We are curious about whether we can get correct embedding results given noisy observations with missing values.

Table 2.2: Accuracy (ACC), normalized mutual information (NMI), and purity (PUR) comparisons of different clustering algorithms: Normalized Cut, Symmetric NMF and Spectral Clustering. $\mathbf{K}^0$: results obtained on the original/input kernel. LLE1: results on learned $\mathbf{Y}$ after 1 LLE iteration. LLE4: results on learned $\mathbf{Y}$ after 4 LLE iterations. All results shown are percentage.

Dataset	Metric	Normalized Cut			Symmetric NMF			Spectral Clustering		
		$\mathbf{K}^0$	LLE1	LLE4	$\mathbf{K}^0$	LLE1	LLE4	$\mathbf{K}^0$	LLE1	LLE4
AT&T	ACC	44.77	50.24	66.50	48.09	49.12	50.04	41.09	53.18	58.31
	NMI	70.14	74.23	83.82	62.28	65.87	70.51	59.40	68.50	74.67
	PUR	49.30	54.87	71.49	48.33	50.32	54.78	48.00	49.24	52.41
Mnist	ACC	64.37	64.87	65.61	73.29	76.43	81.84	73.29	74.21	75.14
	NMI	65.77	66.84	67.25	69.83	72.03	74.92	73.03	73.38	74.93
	PUR	66.55	67.12	68.37	74.16	76.87	81.88	74.69	74.89	75.61
Umist	ACC	48.44	48.85	49.11	49.46	49.87	50.24	43.13	44.87	45.76
	NMI	64.62	64.98	65.15	64.56	65.34	66.95	63.26	63.78	63.89
	PUR	52.06	52.92	53.71	52.43	53.14	54.98	48.85	49.23	50.72
Binalpha	ACC	40.52	42.23	45.91	40.65	42.78	44.67	39.18	42.45	44.26
	NMI	56.25	57.65	60.35	54.49	55.61	59.54	53.57	56.72	58.51
	PUR	43.58	45.54	49.57	43.60	45.71	48.73	41.82	45.23	48.07
Yale	ACC	9.02	12.21	15.49	10.72	11.34	14.78	10.83	10.98	12.89
	NMI	11.24	13.43	20.12	13.98	16.84	20.45	12.72	13.45	16.58
	PUR	9.93	15.53	16.57	11.71	13.23	15.69	11.72	12.37	13.76
Caltec	ACC	36.31	42.43	49.51	43.98	47.83	52.50	43.67	45.74	47.98
	NMI	42.63	45.45	54.86	48.25	52.01	56.43	48.02	50.23	51.84
	PUR	39.02	42.58	53.18	46.21	50.38	55.71	46.41	49.65	51.32
MSRC	ACC	53.23	60.89	66.65	57.86	62.34	66.77	65.85	66.78	68.42
	NMI	44.08	50.23	55.81	46.81	49.87	56.16	54.78	55.23	56.36
	PUR	55.89	61.43	69.95	60.12	64.23	69.62	67.38	68.84	69.64
Newsgroup	ACC	27.58	32.23	40.36	26.62	34.78	51.63	42.22	44.38	46.51
	NMI	12.92	18.24	19.41	17.65	27.86	30.22	18.01	20.32	23.19
	PUR	28.43	32.54	41.95	29.12	42.45	59.20	41.72	44.81	48.90
Reuters	ACC	19.22	23.87	30.59	24.02	35.98	41.35	33.48	34.49	35.83
	NMI	15.69	18.42	22.22	11.30	26.83	32.74	24.26	25.80	27.78
	PUR	19.97	23.34	33.39	24.98	31.90	45.92	37.91	37.98	38.43
Average	ACC	38.16	41.98	47.75	41.63	45.61	50.42	43.64	46.34	48.34
	NMI	42.59	45.50	49.89	43.24	48.03	52.00	45.23	47.49	49.75
	PUR	40.53	43.99	50.91	43.41	47.58	54.06	46.50	48.03	49.87

Table 2.3: Accuracy comparisons of semi-supervised learning on 9 datasets. Learning algorithms used: Harmonic function, Green's function and Local and global consistency(LG-consistency). $\mathbf{K}^0$: results obtained on the original/input kernel. LLE1: results on learned $\mathbf{W}$ after 1 LLE iteration. LLE4: results on learned $\mathbf{W}$ after 4 LLE iterations. Results shown are based on 10% or 20% labeled data.

Dataset	Percent-labeled	Harmonic function			Green's function			LG-consistency		
		$\mathbf{K}^0$	LLE1	LLE4	$\mathbf{K}^0$	LLE1	LLE4	$\mathbf{K}^0$	LLE1	LLE4
AT&T	10%	65.63	70.23	73.14	69.67	70.12	71.11	70.48	71.45	72.12
	20%	74.93	78.87	83.37	78.01	79.03	79.73	78.43	80.23	82.94
	10%	68.83	68.89	69.91	63.35	63.90	64.21	65.72	67.89	69.19
Mnist	20%	81.16	82.09	82.83	72.67	73.42	74.16	75.51	79.38	81.33
	10%	48.64	50.45	51.19	47.91	48.03	48.42	48.87	49.35	50.68
Umist	20%	63.78	67.89	70.43	60.75	61.23	61.54	63.28	68.78	70.48
	10%	47.71	49.89	52.61	46.79	47.09	49.24	46.76	48.93	50.35
Binalpha	20%	53.51	59.23	61.78	52.70	53.28	54.34	52.59	59.37	61.21
	10%	30.31	35.43	38.54	29.13	31.99	32.94	34.67	37.65	43.23
Yale	20%	45.48	52.45	54.18	32.09	33.45	36.55	38.98	48.90	57.49
	10%	44.46	48.76	54.38	44.79	45.08	45.24	44.52	48.75	53.64
Caltec	20%	49.87	53.25	63.67	49.03	50.23	52.34	49.93	53.74	63.62
	10%	57.46	60.35	66.50	59.47	60.01	60.24	60.12	63.45	65.82
MSRC	20%	62.26	65.43	70.95	61.42	62.23	63.54	63.33	68.79	72.15
	10%	65.16	67.34	69.85	53.35	54.23	55.47	56.39	57.78	58.37
Newsgroup	20%	72.27	73.25	74.35	59.72	60.91	61.14	58.84	60.19	61.32
	10%	64.25	65.78	66.23	53.29	55.79	57.81	53.27	58.98	61.44
Reuters	20%	73.61	73.98	74.56	62.35	63.45	68.74	61.09	67.90	72.17
	10%	54.72	57.46	60.26	51.97	52.92	53.85	53.42	56.03	58.32
Average	20%	64.10	67.38	70.68	58.75	59.69	61.34	60.22	65.25	69.19

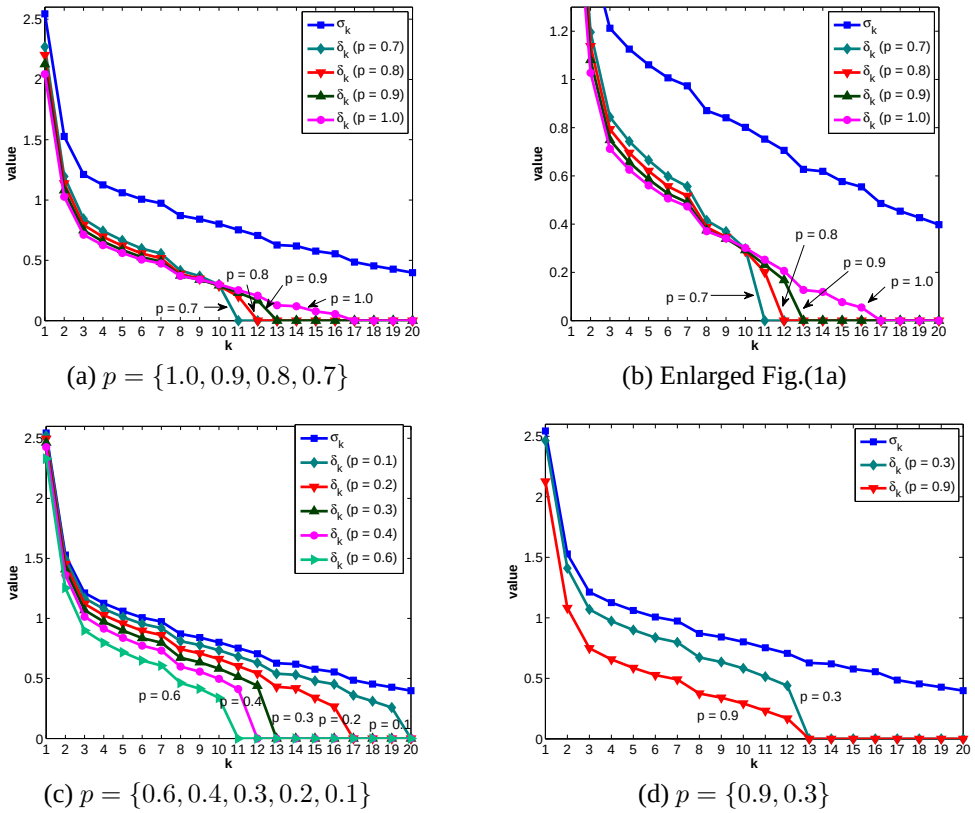
CHAPTER 3

Low Rank Data Recovery with Minimal Shrinkage

Standard trace norm model is used for data recovery purpose. However, the reconstructed data can be shrank and singular values can be greatly suppressed. To solve this problem, we present two low-rank data recovery models through replacing the rank constraint by a Schatten p norm. We analyze the optimal solution of model 1, and characterize the rank of optimal solution. We design two algorithms to solve model 2, one is based on Augmented Lagrangian method (ALM), where a challenge step is to solve associated proximal operator. The other is based on an iterative re-weighted scheme, similar to reweighted L_2 scheme, where rigorous convergence analysis is provided. Extensive experiment results on 6 occluded datasets on computer vision tasks indicate good performance of proposed method.

3.1 Background of low rank data recovery

In big-data era, data is always noisy, development of robust noise tolerant algorithm for data recovery, is always useful and highly demanded. On the other hand, the available of large amount of data makes it more difficult to control the quality the data. The chances of the damaged data or noisy data are increasing. Given input noisy data $\mathbf{X}$, the goal of low rank data recovery problem [78, 79, 80], is to find a low rank approximation $\mathbf{Z}$. Recovered data $\mathbf{Z}$ is expected to be low rank, and retain minimum reconstruction errors (such as least square error) as compared to input data matrix $\mathbf{X}$. In practice, input data can be noisy and also has missing values. This problem has attracted a lot of attentions due to its widely



(a) $p = \{1.0, 0.9, 0.8, 0.7\}$ (b) Enlarged Fig.(1a)

(c) $p = \{0.6, 0.4, 0.3, 0.2, 0.1\}$ (d) $p = \{0.9, 0.3\}$

Figure 3.1: Optimal solution δ_k given singular value σ_k of input data $\mathbf{X}$, at different $p = \{1, 0.9, 0.8, \cdots, 0.1\}$ values with fixed $\beta = 0.5$, on dataset Mnist with 20 images, i.e., $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{20}\}$. To avoid clutter, part of Fig.1a is zoomed in and shown in Fig.1b. In Fig.1d, the solution at $p = 0.3$ is a *faithful* low-rank solution, and the solution at $p = 0.9$ is a *suppressed* low-rank solution.

applications in recommendation systems [81], collaborative prediction [82], image/video completion [83], etc.

Data recovery problem has close relations with dimension reduction or low dimension subspace recovery, since for most of high-dimensional data, they may have low-dimensional subspace. Many efforts have been devoted along the direction of principal component analysis (PCA) [84], compressive sensing [85], affine rank minimization [80], etc. For example, Principal component analysis (PCA) seeks for a low-dimensional subspace given data matrix, which can be efficiently computed using singular value decom-

position (SVD). However, a major drawback of classical PCA [86] is that, it breaks down under grossly corrupted or noisy observations, such as noises/corruptions in images, and dis-measurement in bio-informatics, etc. In Regularized PCA model (*e.g.*, [87, 88]), it aims at reducing the rank of the data without explicitly reducing the dimension. However, they do not return the clear representation of subspace and low-dimensional data explicitly.

It is well known that it is a NP-hard problem to directly minimizing the rank of data for recovering input data. Since trace norm can be viewed as a convex envelope of rank function [89], different methods (*e.g.*, [90, 91, 78, 92, 93, 94]), have been proposed by minimizing the trace norm. In this paper, we point out that, standard trace norm model suffers from a serious problem: *shrinkage of reconstructed data and suppression of singular values* (see more details in Figs.(1-2) and §3.3). We find that the trace norm relaxation may deviate the solution away from the real solution of original rank minimization problem.

The goal of this paper is to develop new methods to solve the approximation of the rank minimization problem. In this paper, we reformulate the noisy data recovery problem using schatten p norm, where efficient algorithms are presented. To summarize, the main contribution of this paper is listed as follows.

- From model construction point of view, we present new models for noisy data recovery, which minimize both data recovery error and rank of recovery data. The proposed models give the minimum shrinkage of recovered data.
- From algorithmic development point of view, we present a complete analysis for proposed model, where the rank of optimal solution is characterized by Theorem 1. Efficient algorithms are developed.
- Extensive experiments on noisy datasets indicate better noisy data recovery performance at smaller p values (p is parameter of our model).

3.2 Proposed Data recovery models

Notation

Let $\mathbf{X} = (x_1 \cdots x_n) \in \Re^{d \times n}$ be input n data, each of dimension d. For standard Schatten p norm of matrix $\mathbf{Z}$,

$$||\mathbf{Z}||_{sp} = (\sum_{k=1}^{r} \sigma_k^p)^{\frac{1}{p}} = \left(\text{Tr}[(\mathbf{Z}^T\mathbf{Z})^{\frac{p}{2}}]\right)^{\frac{1}{p}}, \tag{3.1}$$

where σ_k is the singular value of $\mathbf{Z}$, $r = rank(\mathbf{Z})$.

Given a data matrix $\mathbf{X}$, it is often of interest to compute a matrix $\mathbf{Z}$ that is "close" to $\mathbf{X}$ and satisfies the constraint $rank(\mathbf{Z}) < rank(\mathbf{X})$. Singular value decomposition [95] is the most popular method for such approximations. There are alternative methods that replace this constraint with a more friendly constraint, like, for example, the trace norm. In this paper, we present two models:

Model 1: Schatten p model

We wish to solve the data recovery problem, i.e.,

$$\min_{\mathbf{Z}} \ \frac{1}{2}\|\mathbf{Z} - \mathbf{X}\|_F^2 + \beta \text{Tr}[(\mathbf{Z}^T\mathbf{Z})^{\frac{p}{2}}], \tag{3.2}$$

where $\text{Tr}(\mathbf{Z}^T\mathbf{Z})^{\frac{p}{2}} = \sum_{k=1}^{r} \sigma_k^p$, and σ_k is the singular value of $\mathbf{Z}$, β is a parameter to control the scale of schatten p term.

The fact is that the approximation has the same eigen-vectors as the original matrix, and that only eigen-values are shrinked in standard matrix linear algebra. The particular shrinkage of p Schatten norm is better than trace norm ($p = 1$, see Fig. 1), which is corresponding to soft thresholding. At $p = 0$, this is corresponding to hard thresholding (exactly the rank).

Model 2: Robust Schatten p model

We wish to find low-rank data recovery $\mathbf{Z}$ given $\mathbf{X}$, i.e.,

$$\min_{\mathbf{Z}} \ \|\mathbf{Z} - \mathbf{X}\|_1 + \beta \text{Tr}[(\mathbf{Z}^T\mathbf{Z})^{\frac{p}{2}}]. \tag{3.3}$$

This is used for noisy data recovery purpose, which can be viewed as an extension of robust PCA [87].

Motivation

The goal of proposed models is to provide minimum shrinkage of reconstructed data and suppression of singular values. This is the reason, why we replace the trace norm regularization with schatten p regularization. More detailed analysis is provided in §3-4. Our experiment results indicate that proposed models at smaller p values give better recovery performance.

As p becomes small, it is closer to the desired rank constraint:

$$\lim_{p\to 0} \text{Tr}(\mathbf{Z}^T\mathbf{Z})^{\frac{p}{2}} = \lim_{p\to 0} \sum_k \sigma_k^p = rank(\mathbf{Z}).$$

$$= \sum_k \sigma_k.$$

This indicates that the lower p, the better that Schatten norm resembles the rank. Since we wish to do reconstruction with low rank, thus parameter p is usually set to $0 \leq p \leq 1$. In general $p > 1$ case is un-interesting.

Differences of two models The difference of above two models of Eqs.(3.2, 3.3) lies in the first term. In Model 1 of Eq.(3.2), Frobenius norm or the least square error is used to minimize the reconstruction error. In Model 2 of Eq.(3.3), the L_1-norm is used to minimize the reconstruction error. As is known to us, L_1 error is more robust to noises and outliers, because $||\mathbf{X} - \mathbf{Z}||_1 = \sum_{ij} |\mathbf{X} - \mathbf{Z}|_{ij}$, where residue term is *not* squared. In real world, the observations (like images, text features, etc) can be contaminated by noises or outliers. Model of Eq.(3.2) is for the data recovery problem polluted by Gaussian noise, while model of Eq.(3.3) is for data contaminated by Laplacian noises. Both models can be used to solve noisy data recovery, matrix completion problem, etc. For second term, for computational

purpose, we add p power to standard term $||\mathbf{Z}||_{sp}$, which plays the same role as standard schatten term for low rank approximation purpose.

Relations with previous methods

At $p = 1$, Eq.(3.3) is equivalent to standard trace-norm model, which optimizes

$$\min_{\mathbf{Z}} ||\mathbf{Z} - \mathbf{X}||_1 + \beta||\mathbf{Z}||_*, \tag{3.4}$$

where $||\mathbf{Z}||_* = \text{Tr}(\mathbf{Z}^T\mathbf{Z})^{\frac{1}{2}}$ is the trace norm, and σ_k is the singular value of $\mathbf{Z}$. This study is a special case of our model. Note in [87], Schatten p-Norm model at $p = 1$ is called as Robust PCA, because it can correctly recover underlying low-rank structure $\mathbf{Z}$ from the data $\mathbf{X}$ in the presence of gross errors and outlying observations.

3.3 Illustration of two Schatten p-norm models

Due to the non-smoothness of Schatten norm at $p < 1$, the computational algorithm is challenging. We provide detailed analysis and efficient algorithms of both models in §3.4, §3.5 and §3.6. Here we discuss the general features of the optimal solutions to these two models. The key conclusion is that the solutions at small p are much better than the solution at $p = 1$, which is a previously studied model.

Illustration of Model 1

To illustrate results of Model 1, we use 20 images from real-world dataset mnist (more details of this dataset is in §7). Let δ_k be the singular values of the optimal solution $\mathbf{Z}^*$. Let σ_k be the singular values of input data $\mathbf{X}$. We show solution δ_k in Fig.**??** along with σ_k. We fix $\beta = 0.5$, but let p vary from $p = 1$ to $p = 0.1$. From Fig.1, we see that at $p = 1$, the optimal solution $\mathbf{Z}^*_{p=1}$, which is represented by $(\delta_1, \delta_2, \cdots, \delta_{20})$, is a simple downshift of $(\sigma_1, \sigma_2, \cdots, \sigma_{20})$. The high rank part ($k = 17 - 20$) is zero. As p decreases, more high rank part of the solution $\{\delta_k\}$ becomes zero, while the lower rank part of $\{\delta_k\}$ moves closer to $\{\sigma_k\}$ of the input data. For example, in Fig.1a, Fig.1b, in optimal solution

$\mathbf{Z}^*_{p=0.9}$, the high rank part ($k = 13 - 20$) becomes zero, while the low-rank part ($k = 1 - 7$) is higher than that of $\mathbf{Z}^*_{p=1}$, i.e., this part moves towards corresponding $\{\sigma_k\}$.

In general in low-rank data recovery, we wish the low-rank part of $\mathbf{Z}^*$ is close to those of the input data, while the high-rank part is cut-off (close to zero). Looking in Fig.1d, the solution at $p = 0.3$ is a "faithful" low-rank solution, because the low-rank part is more close or *faithful* to the original data. The solution at $p = 0.9$ is a "suppressed" low-rank solution because the low-rank part is far below the original data, i.e., they are *suppressed*. Clearly, the solution at $p = 0.3$ is more desirable than solution at $p = 0.9$, even though both solutions are low-rank: rank($\mathbf{Z}^*_{p=0.9}$)= rank($\mathbf{Z}^*_{p=0.3}$) = 12.

The Schatten p norm model at small p provides the desirable "faithful" low-rank solution, while the previous work using $p = 1$ also provides a low-rank solution, but the low-rank part is more *suppressed*.

Illustration of Model 2

Model 2 of Eq.(3) differs from Model 1 by using the L_1 norm in error function. This enables the model to do robust data recovery (e.g., moving outliers back to the correct subspace). However, this model does not change the observed *suppression* in Model 1 at p close to 1 (see Fig.1d). The suppression of singular values leads to the *shrinkage* effect in reconstructed data.

We demonstrate the robust data recovery and the shrinkage effects for Model 2 at different p values on a simple toy data in Fig.(3.2a). The original data $\mathbf{X}$ are shown as black circles. Reconstructed data $\mathbf{z}_i$ are shown as red-squares. We show the reconstructed results at $p = 0.2$ Fig.(2b, 2e, 2f), $p = 0.5$ (Fig.2c, 2g), $p = 1$ (Fig.2d, 2h). We have two observations.

First, at $0 \leq p \leq 1$, outliers ($\mathbf{x}_{13}, \mathbf{x}_{14}, \mathbf{x}_{15}$) all move towards the correct subspace, indicating the desired denoising data recovery effects.

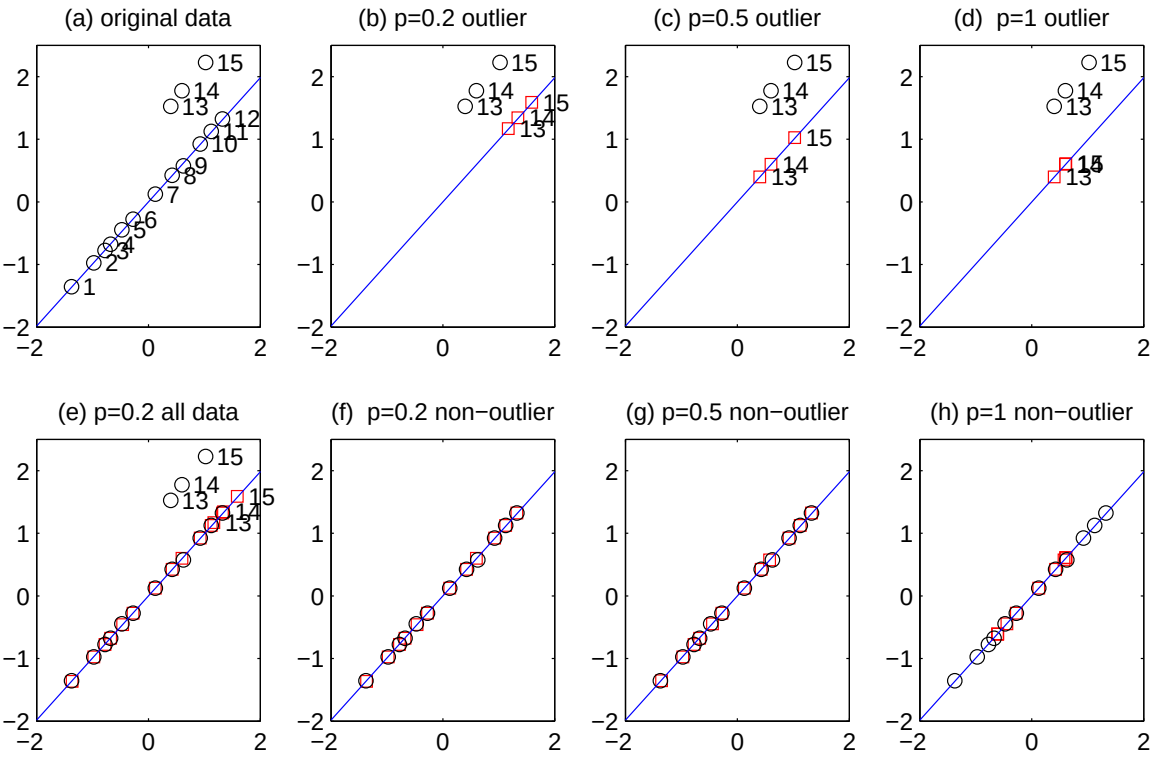


Figure 3.2: Demonstration of robust Schatten-p model of Eq.(3.3) on a toy data shown in panel (a): original data shown as black circles. $(\mathbf{x}_1 \cdots \mathbf{x}_{12})$ are non-outliers and $(\mathbf{x}_{13} \cdots \mathbf{x}_{15})$ are outliers. Reconstructed data $\mathbf{z}_i$ are shown as red-diamonds. Blue line indicates the subspace computed from standard PCA on non-outlier data. Results of Schatten model at $p = 0.2$ are shown in (e). This $p = 0.2$ results are split to outliers and non-outliers as shown in (b) and (f). Similarly, results for $p = 0.5$ shown in (c) and (g); results for $p = 1$ shown in (d) and (h). At $p = 1$, non-outliers shrink towards coordinate (0,0). At smaller p, non-outliers shrink far less.

Second, for non-outlier data, the reconstructed data shrink strongly at $p = 1$, but they shrink much less at $p = \{0.2, 0.5\}$. This shrinkage is result of the singular value suppression in computed $\mathbf{Z}$. At $p = \{0.2, 0.5, 1\}$, the largest singular value are $\{5.35, 4.49, 2.93\}$, while the second singular values are very small, i.e., $\{$1.7e-8, 1.7e-16, 9.8e-9$\}$, respectively.

In summary, the Schatten model at small p enables us to do robust data recovery but without significant shrinkage in previous models which use $p = 1$.

To our knowledge the singular value suppression and shrinkage (both at $p = 1$ and smaller p values) have not been studied previously.

3.4 Analysis and Algorithm of Model 1

We show how to solve Model 1 of Eq.(3.2) at different p values. This also serves as the basic step in solving Model 2 of Eq.(3) using the ALM of §5. To our knowledge, this problem has not been studied before.

Property 1. The global optimal solution for Eq.(3.2) at all $0 \leq p \leq 1$, can be efficiently computed, even though it is non-convex at $p < 1$.

Property 2. Rank of the optimal solution $\mathbf{Z}^*$ has a closed form solution:

Theorem 4. *Let the singular value decomposition (SVD) of* $\mathbf{X}$ *be* $\mathbf{X} = \sum_k \sigma_k \mathbf{u}_k \mathbf{v}_k^T$. *Then rank of optimal solution* $\mathbf{Z}^*$*:* $rank(\mathbf{Z}^*) =$ *largest* k*, such that*

$$\sigma_k \leq \Big(\frac{\beta p(2-p)^{(2-p)}}{(1-p)^{(1-p)}}\Big)^{\frac{1}{2-p}}, \quad 0 < p \leq 1. \tag{3.5}$$

In particular, $p = 1, \sigma_k \leq \beta; p = \frac{1}{2}, \sigma_k \leq (\sqrt{\frac{27}{16}}\beta)^{\frac{2}{3}}$.

Property 3. Optimal solution $\mathbf{Z}^*$ has a closed form solution at $p = \frac{1}{2}$.

Property 4. Optimal solution $\mathbf{Z}^*$ at $0 < p < 1$ can be obtained using Newton's method.

To prove above 4 properties for Model 1 of Eq.(2), we need the following useful lemma.

Lemma 2. *Let the singular value decomposition (SVD) of* $\mathbf{X}$ *be* $\mathbf{X} = \sum_k \sigma_k \mathbf{u}_k \mathbf{v}_k^T$. *The global optimal* $\mathbf{Z}$ *for Eq.(3.2) is given by* $\mathbf{Z} = \sum_k \delta_k \mathbf{u}_k \mathbf{v}_k^T$*, where* δ_k *is given by solving,*

$$\min_{\delta_1,\cdots,\delta_r} \sum_{k=1}^{r} \Big[\frac{1}{2}(\delta_k - \sigma_k)^2 + \beta\delta_k^p\Big], \;\; s.t.\; \delta_k \geq 0, k = 1 \cdots r. \tag{3.6}$$

Proof of Lemma 1

Proof. Let the optimal solution of $\mathbf{Z}$ have the SVD $\mathbf{Z} = \mathbf{F}\Delta\mathbf{G}^T$ where $\mathbf{F} = (\mathbf{f}_1 \cdots \mathbf{f}_r)$ and $\mathbf{G} = (\mathbf{g}_1 \cdots \mathbf{g}_r)$ are the singular vectors of $\mathbf{Z}$, and $\Delta = \mathrm{diag}(\delta_1 \cdots \delta_r)$ be their singular

values. The key is to prove that the singular vectors of $\mathbf{Z}^*$ are the same as those of the input data $\mathbf{X}$. Using von Neumann's trace inequality

$$|\text{Tr}(\mathbf{Z}^T\mathbf{X})| \leq \text{Tr}\Delta\Sigma = \sum_{k=1}^{r} \delta_k\sigma_k. \tag{3.7}$$

From this, we have

$$\text{Tr}(\mathbf{U}\Delta\mathbf{V}^T)^T\mathbf{X} = \text{Tr}\Delta\Sigma \geq \text{Tr}(\mathbf{Z}^T\mathbf{X}) = \text{Tr}(\mathbf{F}\Delta\mathbf{G}^T)^T\mathbf{X}, \tag{3.8}$$

where the inequality comes from Eq.(3.7). The inequality

$$\text{Tr}(\mathbf{U}\Delta\mathbf{V}^T)^T\mathbf{X} \geq \text{Tr}(\mathbf{F}\Delta\mathbf{G}^T)^T\mathbf{X}$$

implies

$$\frac{1}{2}\|\mathbf{U}\Delta\mathbf{V}^T - \mathbf{X}\|^2 + \beta\text{Tr}\Delta^p \leq \frac{1}{2}\|\mathbf{F}\Delta\mathbf{G}^T - \mathbf{X}\|^2 + \beta\text{Tr}\Delta^p.$$

This indicates $(\mathbf{U}, \mathbf{V})$ are better singular vectors for $\mathbf{Z}$ than $(\mathbf{F}, \mathbf{G})$. This proves that the optimal singular vectors for $\mathbf{Z}$ must be the same singular vectors of $\mathbf{X}$. Setting $\mathbf{Z} = \mathbf{U}\Delta\mathbf{V}^T$ in Eq.(3.2), we obtain Eq.(3.6). □

Analysis of Property 1

Due to Lemma 1, we now solve the simpler problem of Eq.(3.6) instead of the original harder problem of Eq.(3.2). Clearly the optimization of Eq.(3.6) decouples into r independent subproblems, each for one δ_k:

$$\min_{\delta_k} \ \frac{1}{2}(\delta_k - \sigma_k)^2 + \beta\delta_k^p, \ \ s.t.\ \delta_k \geq 0. \tag{3.9}$$

KKT complementarity slackness condition for $\delta_k \geq 0$ leads to $\left[(\delta_k - \sigma_k) + p\beta\delta_k^{p-1}\right]\delta_k = 0$. The optimization of Eq.(3.9) decouples into r independent subproblems, and each of them is of the type:

$$\min_{x\geq 0} \ J(x) = \frac{1}{2}(x - a)^2 + \beta x^p, \tag{3.10}$$

where $x, a \in \Re$. Here the correspondence between Eq.(3.10) and Eq.(3.9) is $a = \sigma_k$, $x = \delta_k$.$J(x)$ is a weight sum over two functions: $J(x) = f_1(x) + \beta f_2(x)$, where $f_1(x) =$

$\frac{1}{2}(x-a)^2, f_2(x) = x^p$. $f_1(x)$ has a local minima at $x_1 = a$. $f_2(x)$ is a singular function, $p \leq 1$ with singularity at $x_2 = 0$, which is also a local minima.

Therefore, $J(x)$ in general has two local minima (x_1^*, x_2^*). Because $f_2(x)$ is singular at x_2, for Eq.(3.10), the singular point (local minima) does not change with different weight β. Thus $x_2^* = 0$ is always a local minima.

When β is small, $x_1^* = a$. As β increases, x_1^* moves towards 0. At certain (β, p), this local minima disappears, $J(x)$ has only one local minima $x_2^* = 0$. This condition is determined by the same condition as in Theorem 1 or Eq.(12) with $\sigma_k = a$. x_1^* is easily computed using Property 4.

In summary, the optimal solution of Eq.(10) is either the trivial one $x_2^* = 0$ or $\min(x_1^*, x_2^*)$, when x_1^* exits. This means Eq.(9) can be easily solved. Thus Eq.(6) can be easily solved for each rank one at a time.

Proof of Theorem 1

Proof. First, optimization of Eq.(3.2) is equivalent to optimizing Eq.(3.10), which can be further written as,

$$\min_{z \geq 0} \; g(z) = \frac{1}{2}(z-1)^2 + \hat{\beta} z^p, \tag{3.11}$$

where $z = x/a, \hat{\beta} = \beta a^{(p-2)}$. First, we note a key quantity, the zero crossing point z_0 exists, where the second derivative $g''(z)$ changes its sign, i.e., $g''(z_0) = 0$. We need two lemmas.

Lemma 3. *This cross point z_0 always exists at any β.*

Lemma 4. *If the slope of cost function of Eq.(3.11) at the crossing point z_0 is negative, i.e., $g'(z_0) < 0$, there exists two distinct local minima: $z_2 = 0$ and $z_1 > 0$. If $g'(z_0) \geq 0$, $z_2 = 0$ is the global optimal solution.*

Lemmas 2 and 3 give the key properties of optimization of Eq.(3.11). Set $g''(z_0) = 0$, we obtain $z_0 = [\hat{\beta} p(1-p)]^{\frac{1}{2-p}}$. Lemma 2 states that $z_2 = 0$ is the global solution, $g'(z_0) =$

$z_0 - 1 + \hat{\beta} p z_0^{p-1} \geq 0$, i.e., $[\hat{\beta} p(1-p)]^{\frac{1}{2-p}} - 1 + \hat{\beta} p [\hat{\beta} p(1-p)]^{\frac{p-1}{2-p}} \geq 0$. Solving for β, we have,

$$\beta \geq \frac{1(1-p)^{(1-p)}}{p(2-p)^{(2-p)}} \cdot {\sigma_k}^{(2-p)}, \quad 0 < p \leq 1. \tag{3.12}$$

This indicates that the optimal solution δ_k of Eq.(3.11) is zero (i.e., $\delta_k = 0$), if Eq.(3.12) holds. This completes the proof. □

Analysis of Property 3

Clearly, at $p = \frac{1}{2}$, from Eq.(3.9), we need to solve $\delta_k - \sigma_k + (\beta/2)\delta_k^{-1/2} = 0,\ s.t.\ \delta_k \geq 0$. Let $\rho_k = (\frac{\delta_k}{\sigma_k})^{1/2}$, $\mu = \frac{\beta}{2{\sigma_k}^{\frac{3}{2}}}$, this becomes $\rho_k^3 - \rho_k + \mu = 0$, where $\rho_k \geq 0$. The analytic solution of this cubic equation can be solved in closed form.

Analysis of Property 4

From analysis of property 1, the optimization of Eq.(3.10) has two local optima: $x_1^* > 0, x_2^* = 0$. Our algorithm is: (b1) to use Newton's method to compute x_1^*; (b2) compare $J(x_1^*), J(x_2^*)$, and pick the smaller one. It is easy to see $J'(x) = x - a + \beta p x^{p-1}$, $J''(x) = 1 + \beta p(p-1)x^{p-2}$. Using standard Newton's method, we can update x through $x \leftarrow x - \frac{J'(x)}{J''(x)}$. This algorithm has quadratic convergence. In practical applications, we found this Newton's algorithm typically converges to local minima within a few iterations.

3.5 Efficient ALM algorithm

Augmented lagrange multipliers(ALM) have been widely used to solve different kinds of optimization problems ([87], [96]). Here we adapt standard ALM method [97, 96] to solve Schatten-p model of Eq.(3.3). It is worth noting that it is not trivial to solve Eq.(3.3) using ALM method. One challenging step is to solve the associated Schatten-p term shown in §4.

Algorithm 2 ALM algorithm to solve Eq.(3.3)

Input: data matrix $\mathbf{X}$, parameter $\rho > 1$.

Output: low rank approximation $\mathbf{Z}$.

Procedure:

1: Initialize $\mathbf{E}$, $\mathbf{Z}$, Ω, $\mu > 0$, $t = 0, ;\ \ \rho = 1.1$

2: **while** Not converge **do**

3: Updating $\mathbf{E}$ according to Eq.(3.16)

4: Updating $\mathbf{Z}$ according to Eq.(3.17)

5: Updating Ω: $\Omega := \Omega + \mu(\mathbf{Z} - \mathbf{X} - \mathbf{E})$

6: Updating μ: $\mu := \rho\mu$

7: **end while**

According to ALM algorithm, by imposing constraint variable $\mathbf{E} = \mathbf{Z} - \mathbf{X}$, the problem of Eq.(3.3) is equivalent to solve,

$$\min_{\mathbf{E},\,\mathbf{Z}} \ \|\mathbf{E}\|_1 + \beta \mathrm{Tr}(\mathbf{Z}^T\mathbf{Z})^{\frac{p}{2}}, \qquad s.t. \qquad \mathbf{Z} - \mathbf{X} - \mathbf{E} = 0. \tag{3.13}$$

According to ALM algorithm, we need to solve,

$$\min_{\mathbf{E},\,\mathbf{Z}} \ \|\mathbf{E}\|_1 + \langle \Omega, \mathbf{Z} - \mathbf{X} - \mathbf{E}\rangle + \frac{\mu}{2}||\mathbf{Z} - \mathbf{X} - \mathbf{E}||_F^2 + \beta \mathrm{Tr}(\mathbf{Z}^T\mathbf{Z})^{\frac{p}{2}}, \tag{3.14}$$

where Lagrange multiplier is Ω and μ is penalty constant. For this problem, Ω and μ updated in a specified pattern:

$$\Omega \leftarrow \Omega + \mu(\mathbf{Z} - \mathbf{X} - \mathbf{E}), \ \ \mu \leftarrow \rho\mu.$$

We need to search for optimal $\mathbf{E}$, $\mathbf{Z}$ iteratively until the algorithm converges. Now we discuss how we solve $\mathbf{E}$, $\mathbf{Z}$ in each step.

Update $\mathbf{E}$ To update the error matrix $\mathbf{E}$, we derive Eq.(3.15) with fixed $\mathbf{Z}$ and obtain the following form:

$$\min_{\mathbf{E}} \ \frac{\mu}{2}||\mathbf{E} - \mathbf{A}||_F^2 + ||\mathbf{E}||_1 \tag{3.15}$$

where $\mathbf{A} = \mathbf{X} - \mathbf{Z} + \frac{\Omega}{\mu}$. It is well-known that the solution to the above LASSO type problem [11] is given by,

$$\mathbf{E}_{ij} = sign(\mathbf{A}_{ij}) \max(|\mathbf{A}_{ij}| - \frac{1}{\mu}, 0). \tag{3.16}$$

Update $\mathbf{Z}$ To update $\mathbf{Z}$ while fixing $\mathbf{E}$, we minimize the relevant part of Eq.(3.14), which is

$$\min_{\mathbf{Z}} \beta \mathrm{Tr}(\mathbf{Z}\mathbf{Z}^T)^{\frac{p}{2}} + \frac{\mu}{2}||\mathbf{Z} - \mathbf{X} - \mathbf{E} + \frac{\Omega}{\mu}||_F^2. \tag{3.17}$$

Setting $\mathbf{B} = \mathbf{X} + \mathbf{E} - \frac{\Omega}{\mu}, \hat{\beta} = \frac{\beta}{\mu}$, this optimization becomes Eq.(3.2), which has been solved in §4.

3.6 Iterative algorithm to solve Model 2

We present another efficient iterative algorithm to solve Eq.(3.3), where the variable matrix $\mathbf{Z}$ is updated iteratively. Suppose $\mathbf{Z}_t$ is the value of $\mathbf{Z}$ at t-th step. At step t, the key step of our algorithm is to iteratively update j-th column ($\mathbf{z}_j$) of $\mathbf{Z}$ one at a time, according to

$$\mathbf{z}_j = \mathbf{A}^{-1}(\mathbf{A}^{-1} + p\lambda \mathbf{D}_j^{-1})^{-1}\mathbf{x}_j, \tag{3.18}$$

where $\mathbf{A} = (\mathbf{Z}_t\mathbf{Z}_t^T)^{p/2-1}, \mathbf{W}_{ij} = 1/|(\mathbf{Z}_t - \mathbf{X})_{ij}|, \mathbf{D}_j = \mathrm{diag}(\mathbf{w}_j)$, $\mathbf{w}_j$ is the j-th column of $\mathbf{W}$. This process is iteratively done for $1 \leq j \leq n$. Then $\mathbf{Z}$ is updated until the algorithm converges. More detailed algorithm is summarized in Algorithm 2. In computing $\mathbf{z}_j$ of Eq.(3.18), we first use conjugate gradient method to compute $\tilde{\mathbf{z}}_j$, where $(\mathbf{A}^{-1} + p\lambda\mathbf{D}_j^{-1})\tilde{\mathbf{z}}^j = \mathbf{x}_j$, and then $\mathbf{z}_j = \mathbf{A}^{-1}\tilde{\mathbf{z}}_j$.

Convergence of algorithm

Let $J(\mathbf{Z}) = \|\mathbf{Z} - \mathbf{X}\|_1 + \beta \mathrm{Tr}(\mathbf{Z}^T\mathbf{Z})^{\frac{p}{2}}$, we have

Theorem 5. *Updating* $\mathbf{Z}$ *using Eq.(3.18),* $J(\mathbf{Z})$ *decreases monotonically.*

The proof requires the following two Lemmas.

Algorithm 3 An iterative algorithm to solve Eq.(3.3)

Input: $\mathbf{X}$, λ

Output: $\mathbf{Z}$

1: **while** *not converge* **do**

2: compute $\mathbf{A}^{-1}$

3: **for** $j = 1 : n$ **do**

4: compute $\mathbf{D}_j^{-1}$, solve $\mathbf{z}_j$ according to Eq.(3.18)

5: **end for**

6: **end while**

Lemma 5. *Define the objective function*

$$J_2(\mathbf{Z}) = \|\mathbf{Z} - \mathbf{X}\|_{\mathbf{W}}^2 + p\beta Tr(\mathbf{Z}^T\mathbf{A}\mathbf{Z}). \tag{3.19}$$

where $\|\mathbf{A}\|_{\mathbf{W}}^2 = \sum_{ij} A_{ij}^2 W_{ij}$. *The updated* $\mathbf{Z}_{t+1}$ *using Eq.(3.18) satisfies*

$$J_2(\mathbf{Z}_{t+1}) \leq J_2(\mathbf{Z}_t) \tag{3.20}$$

Lemma 6. *The updated* $\mathbf{Z}_{t+1}$ *using Eq.(3.18) satisfies*

$$J(\mathbf{Z}_{t+1}) - J(\mathbf{Z}_t) \leq \frac{1}{2}\big[J_2(\mathbf{Z}_{t+1}) - J_2(\mathbf{Z}_t)\big] \tag{3.21}$$

Proof of Theorem 2

Proof. From Eq.(3.20), clearly, LHS of Eq.(3.21) is LHS ≤ 0. □

Proof of Lemma 4

Proof. Setting $\partial J_2(\mathbf{Z})/\partial Z_{ij} = 0$, we have $(\mathbf{Z} - \mathbf{X})_{ij}W_{ij} + p\lambda(\mathbf{A}\mathbf{Z})_{ij} = 0$. This can be written as $Z_{ij}W_{ij} + p\lambda(\mathbf{A}\mathbf{Z})_{ij} = X_{ij}W_{ij}$. In matrix form, $\mathbf{D}_j\mathbf{z}_j + p\lambda\mathbf{A}\mathbf{z}_j = \mathbf{D}_j\mathbf{x}_j$. Thus we have

$$\mathbf{z}_j = (\mathbf{D}_j + p\lambda\mathbf{A})^{-1}\mathbf{D}_j\mathbf{x}_j = [\mathbf{D}_j(\mathbf{A}^{-1} + p\lambda\mathbf{D}_j^{-1})\mathbf{A}]^{-1}\mathbf{D}_j\mathbf{x}_j, \tag{3.22}$$

which gives Eq.(3.18). □

Proof of Lemma 5

Proof. Let Δ = LHS - RHS of Eq.(3.21). We have $\Delta = \alpha + \beta$ where

$$\begin{aligned}\alpha &= \sum_{ij}\Big[|(\mathbf{Z}_{t+1}-\mathbf{X})_{ij}| - |(\mathbf{Z}_t-\mathbf{X})_{ij}| - \frac{(\mathbf{Z}_{t+1}-\mathbf{X})_{ij}^2}{2|(\mathbf{Z}_t-\mathbf{X})_{ij}|}\frac{(\mathbf{Z}_t-\mathbf{X})_{ij}^2}{2|(\mathbf{Z}_t-\mathbf{X})_{ij}|}\Big] \\ &= \sum_{ij}\frac{-1}{2|(\mathbf{Z}_t-\mathbf{X})_{ij}|}\Big[|(\mathbf{Z}_{t+1}-\mathbf{X})_{ij}| - |(\mathbf{Z}_t-\mathbf{X})_{ij}|\Big]^2 \le 0.\end{aligned}$$

and

$$\begin{aligned}\beta &= \lambda\Big[\mathrm{Tr}(\mathbf{Z}_{t+1}\mathbf{Z}_{t+1}^T)^{\frac{p}{2}} - \mathrm{Tr}(\mathbf{Z}_t\mathbf{Z}_t^T)^{\frac{p}{2}}\Big] - \frac{p}{2}\lambda\Big[\mathrm{Tr}\mathbf{Z}_{t+1}^T(\mathbf{Z}_t\mathbf{Z}_t^T)^{\frac{p}{2}}\mathbf{Z}_{t+1} - \mathrm{Tr}\mathbf{Z}_t^T(\mathbf{Z}_t\mathbf{Z}_t^T)^{\frac{p}{2}}\mathbf{Z}_t\Big] \\ &= \lambda\Big[\mathrm{Tr}(\mathbf{Z}_{t+1}\mathbf{Z}_{t+1}^T)^{\frac{p}{2}} - \mathrm{Tr}(\mathbf{Z}_t\mathbf{Z}_t^T)^{\frac{p}{2}}\Big] - \frac{p}{2}\lambda\mathrm{Tr}\Big[\big(\mathbf{Z}_{t+1}\mathbf{Z}_{t+1}^T - \mathbf{Z}_t\mathbf{Z}_t^T\big)(\mathbf{Z}_t\mathbf{Z}_t^T)^{\frac{p}{2}}\Big] \\ &\le 0, \qquad (3.23)\end{aligned}$$

where in the last inequality, we set $\mathbf{A} = \mathbf{Z}_{t+1}\mathbf{Z}_{t+1}^T,\ \mathbf{B} = \mathbf{Z}_t\mathbf{Z}_t^T$ and used Lemma 6 below. Clearly $\Delta = \alpha + \beta \le 0$. □

Lemma 7. *[98] For any two symmetric positive definite matrices* $\mathbf{F}, \mathbf{G}$ *and* $0 < p \le 2$*,*

$$Tr\,[\mathbf{F}^{p/2} - \mathbf{G}^{p/2}] \le \frac{p}{2} Tr\,[(\mathbf{F}-\mathbf{G})\mathbf{G}^{p/2-1}] \qquad (3.24)$$

Due to space limit, we omit the proofs of Lemma 6 here.

3.7 Connection to related works

We note [99] proposes an algorithm to solve squared schatten p model, i.e., $\min_{\mathbf{Z}} f(\mathbf{Z}) + \beta\Big(\mathrm{Tr}(\mathbf{Z}^T\mathbf{Z})^{\frac{p}{2}}\Big)^{\frac{2}{p}}$, which cannot be directly applied here. [100] proposes an iterative reweighted algorithm for trace norm minimization problem, in the similar vein as what has been proposed for adaptive lasso. However, it cannot be directly applied to solve Eq.(3.3). As compared to [98, 101, 102, 103], our goal is for noisy data recovery problem raised in computer vision, instead of for matrix completion problems with missing values.

Table 3.1: Description of Data sets

Dataset	#data	#dimension	#class
AT&T_oc	400	2576	40
Binalpha_oc	1404	320	36
Umist_oc	360	644	20
YaleB	256	2016	4
CMUPIE_oc	680	1024	68
Mnist_oc	150	784	10

3.8 Experiments

We use six widely used image data sets, including four face datasets: AT&T Umist, YaleB [72] and CMUPIE; and two digit datasets: Mnist [73] and Binalpha [1]. We generate occluded image datasets corresponding to 5 original data sets (except YaleB). For YaleB dataset, the images are taken under different poses with different illumination conditions. The shading parts of the images play the similar role of occlusion (noises). Thus we use the original YaleB data with first 4 persons in our experiments. For the other 5 datasets, half of the images are selected from each category for occlusion with block size of wxw pixels (e.g., $w = 10$). The locations of occlusions are random generated without overlaps among the images from the same category. *Occluded* images (with occlusion size 7×7) generated from Umist data sets are shown in Fig. 3.4. Table 3.1 summarizes the characteristics of these occluded data sets.

We did all experiments using Eq.(3.3). At $p < 1$, objective function in Eq.(3.3) is not convex any more, and we cannot get global minima. We initialize $\mathbf{Z}$ using trace norm minimization solution, i.e., set $p = 1$ in Eq.(3.3). In the following experiments, we did both algorithms proposed in §5-6, and reported the results using the one achieving smaller objectives.

Illustrative examples.

[1] http://www.kyb.tuebingen.mpg.de/ssl-book/ benchmarks.html

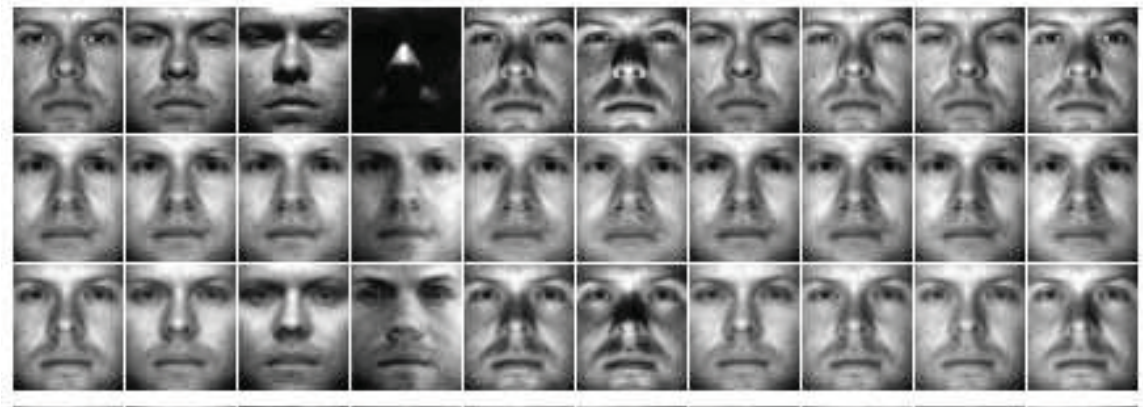


Figure 3.3: Reconstructed images ($\mathbf{Z}$) of YaleB dataset using Model 2 of Eq.(3) shown in 1 panel. First line: original images of one person, Second line: reconstructed images $\mathbf{Z}$ at $p = 1$, Third line: reconstructed images at $p = 0.2$. One can see $p = 1$ images are very similar to each other (most fine details are lost), while $p = 0.2$ images retain some fine details and are closer to original images.

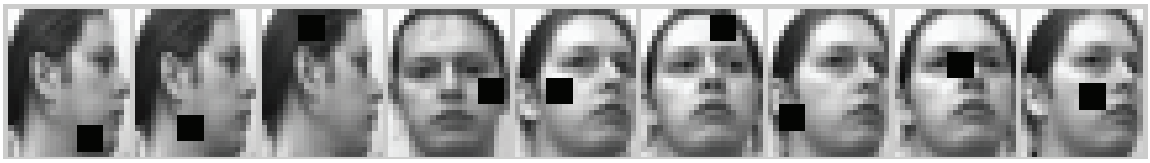


Figure 3.4: Occluded image dataset Umist.

To visualize the denoising effect of proposed method, we apply our model on YaleB dataset. YaleB contains images with different shading which plays similar role of occlusion (noises). Thus we did not add occlusion and use the original data. In this demonstration and following experiment, each data (image) is linearized into a vector each $\mathbf{x}_i$, and the input matrix $\mathbf{X}$ is constructed as $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n)$. We typically set the rank k equal to the number of classes in the dataset. Due to space limit, computed $\mathbf{Z}$ at different p values for the two persons are shown. In Fig.(3.3), 20 images are shown as 2 panels, each panel for one person. On each panel, the first line images are original images $\mathbf{X}$, the 2nd line are computed $\mathbf{Z}$ at $p = 1$, and 3rd line are computed $\mathbf{Z}$ at 0.2.

Clearly, at different p values (such as $p = \{1, 0.2\}$), Schatten p-Norm model can effectively recover the original data by removing the shadings. See 2nd line on each panel

Table 3.2: True data recovery: True signal reconstruction error at different p on six datasets

dataset	$\frac{\|\|\mathbf{X}_E\|\|_F}{\|\|\mathbf{X}_0\|\|_F}$	Noise-free reconstruction error at different p				
		$p=1$	$p=0.75$	$p=0.5$	$p=0.2$	$p=0$
AT&T	0.3657	0.2672	0.2240	0.2199	0.2159	**0.2132**
Binalpha	0.2359	0.2023	0.1974	0.1845	**0.1594**	0.1729
Umist	0.3123	0.2816	0.2290	0.2199	0.2153	**0.2151**
YaleB	N/A	0.2304	0.2264	0.2174	**0.1912**	0.2126
CMUPIE	0.2542	0.2012	0.1925	0.1845	**0.1594**	0.1729
Mnist	0.5574	0.5123	0.4993	0.4814	**0.4542**	0.4553

in Fig.(3.3), almost every person is recovered to same template, not any difference any more. In contrast, we have much better visualization results (with more details) when $p = 0.2$ (see 3rd line on each panel). Moreover, these fine details are expected to be helpful for classification on images from different persons.

True data recovery: true signal reconstruction error

Given noisy data $\mathbf{X}$, $\mathbf{X} = \mathbf{X}_0 + \mathbf{X}_E$, where $\mathbf{X}_0$ is the true signal and $\mathbf{X}_E$ is the noise. Our goal is to recover $\mathbf{X}_0$ using Eq.(3.3). We did experiments on above 6 datasets. To evaluate the performance, we define the true signal data recovery error, $E_{\text{true-signal}} = \frac{||\mathbf{Z}-\mathbf{X}_0||_F}{||\mathbf{X}_0||_F}$. Clearly, smaller $E_{\text{true-signal}}$ values indicate better recovery. Computed true signal reconstruction error are shown in Table.3.2. The experiment results indicate that true signal reconstruction errors are *smaller* at smaller p values. We also list $\frac{||\mathbf{X}_E||_F}{||\mathbf{X}_0||_F}$ values in Table.3.2 to indicate the level of occlusions. Interestingly, $E_{\text{true-signal}} < \frac{||\mathbf{X}_E||_F}{||\mathbf{X}_0||_F}$ on all datasets at different p values. This further confirms "de-noisy" effects of proposed data recovery model.

Loss of fine details in recovered data and its measure.

Due to suppression of higher order/frequency terms associated with smaller singular values, fine details of original data $\mathbf{X}$ are lost in the recovered $\mathbf{Z}$. As a consequence, recovered individual images are very similar to each other. One numeric measure is the variance of reconstructed images. We therefore define $var(\mathbf{Z}) = \sum_{i=1}^{n} ||\mathbf{z}_i - \bar{\mathbf{z}}||^2$, $\bar{\mathbf{z}} =$

Table 3.3: Loss of fine-details: variance of reconstructed $\mathbf{Z}$ on six datasets, original images: $\mathbf{X}_0$, occluded images: $\mathbf{X}$

dataset	$\mathbf{X}_0$	$\mathbf{X}$	Variance of $\mathbf{Z}$ at different p				
			$p = 1$	$p = 0.75$	$p = 0.5$	$p = 0.2$	$p = 0$
AT&T	8.89	9.03	5.83	7.13	7.45	8.11	7.80
Binalpha	27.90	31.13	13.40	22.89	25.38	26.89	26.73
Umist	7.01	7.42	3.87	5.31	5.71	6.38	6.01
YaleB	9.75	9.75	7.28	8.22	8.59	9.19	8.76
CMUPIE	12.09	13.16	8.12	10.07	10.54	11.30	10.87
Mnist	9.24	10.26	0.49	4.41	5.45	7.04	5.85

$\frac{1}{n}\sum_{i=1}^{n}\mathbf{z}_i$, where $\mathbf{z}_i \in \Re^{d\times 1}$ is the reconstructed image corresponding to each original image $\mathbf{x}_i$. Larger variance values indicate more fine details are preserved in the solution $\mathbf{Z}$. Computed variance of $\mathbf{Z}$ are shown in Table.3.3. Clearly, reconstructed images preserve more detailed information at small p (say $p = 0.2$). One demonstrating example is shown in Fig.3, where fine details of individual images are mostly suppressed at $p = 1$, but are generally preserved/reetained at $p = 0.2$.

Classification results using recovered $\mathbf{Z}$.

So far we have discussed low rank recovery capability of computed $\mathbf{Z}$. Reconstructed low rank $\mathbf{Z}$ is expected to have much clear structure after removing noises and outliers. As a by-product of solving low-rank data recovery problem, computed $\mathbf{Z}$ can be used for classification tasks. We compare the classification results by using the occluded images $\mathbf{X}$ and recovered data $\mathbf{Z}$ at different p. The experiments are done on two widely used classifiers: k nearest neighbor (kNN) and support vector machine[1] using 5 fold cross validation. Since the regularization coefficient is also a hyper-parameter, the performance of each Schatten-p norm model is evaluated at an optimal value of β (which is determined by cross validation). The experiment results are shown in Table.3.5. We have two important observations from experiment results. (1) Performances for image categorization tasks are improved by

[1]http://www.csie.ntu.edu.tw/ cjlin/libsvm/

Table 3.4: Classification accuracy(shown as percentage) on six occluded datasets using input corrupted data $\mathbf{X}$ and reconstructed $\mathbf{Z}$ at different p values

dataset	method	$\mathbf{X}$	Reconstructed $\mathbf{Z}$ at different p				
			$p=1$	$p=0.75$	$p=0.5$	$p=0.2$	$p=0$
AT&T	SVM	29.75	30.52	33.75	34.25	**36.78**	35.53
	KNN	25.75	28.63	**30.25**	28.75	29.31	28.33
Binalpha	SVM	38.35	44.78	42.74	43.84	**48.53**	47.43
	KNN	52.09	56.78	55.10	54.65	**58.23**	57.87
Umist	SVM	59.83	65.89	63.17	64.35	**68.33**	67.67
	KNN	89.12	93.89	92.67	93.75	**94.23**	93.01
YaleB	SVM	46.11	52.12	51.89	53.78	54.67	**54.96**
	KNN	85.43	90.89	90.36	91.15	**91.76**	91.40
CMUPIE	SVM	29.24	33.57	**36.74**	34.21	35.39	34.98
	KNN	58.12	64.03	65.38	64.27	**66.39**	65.64
Mnist	SVM	49.38	51.93	53.24	**57.18**	56.79	54.67
	KNN	76.63	81.35	80.75	81.56	**82.47**	82.34

Table 3.5: Classification accuracy(shown as percentage) on six occluded datasets using input corrupted data $\mathbf{X}$ and reconstructed $\mathbf{Z}$ at different p values

dataset	method	$\mathbf{X}$	Reconstructed $\mathbf{Z}$ at different p				
			$p=1$	$p=0.75$	$p=0.5$	$p=0.2$	$p=0$
Umist	KNN	84.12	90.89	92.67	93.75	**94.23**	93.01
YaleB	KNN	82.43	88.89	90.36	91.15	**91.76**	91.40
Mnist	KNN	73.63	79.35	80.75	81.56	**82.47**	82.34

using computed $\mathbf{Z}$ at different p values; (2) Classification accuracy is consistently better at smaller p values on both SVM and kNN classifiers, as compared to that at large p values. All above results suggest us to use Schatten p-Norm at small p values.

3.9 Lessons learned

We present novel models for low-rank data recovery, where efficient algorithms are proposed. Extensive experiment results indicate schatten p model gives relatively better reconstructed results at small p values. In the next step, we will further explore how to scale our model for large-size problems.

CHAPTER 4

Efficient Algorithms for Selecting Features with Arbitrary Group Constraints

Feature structure information plays an important role for regression and classification tasks. We consider a more generic problem: group lasso problem, where structures over feature space can be represented as a combination of features in a group.These groups can be either overlapped or non-overlapped, which are specified in different structures, e.g., structures over a line, a tree, a graph or even a forest. We propose a new approach to solve this generic group lasso problem, where certain features are selected in a group, and an arbitrary family of subset is allowed. We employ accelerated proximal gradient method to solve this problem, where a key step is solve the associated proximal operator. We propose a fast method to compute the proximal operator, where its convergence is rigorously proved. Experimental results on different structures (e.g., group, tree, graph structures) demonstrate the efficiency and effectiveness of the proposed algorithm.

4.1 Background of feature selection using structural sparsity

Lasso is widely used for variable selection in high-dimensional space by capturing the structure information. Structured-sparsity-inducing regularization is proposed to encourage the joint selection of closely related input variables [104], [9], [105], [106], [107]. Desirable for practical applications, arbitrary structures can be allowed on the feature set, which further generalizes the concept of structure sparsity, such as group lasso [107], fused lasso (linear-ordering of variables) [106], lasso on forest [108], etc, where a number of composite gradient methods (e.g., [109], [110], [111]) have been developed to solve above problems.

Structured sparsity [10, 112] can enforce sparsity on variables of groups, trees, graphs, and even forest, where the solution is zeros or nonzeros in group-wise. For example, $L_{2,p}$ [10] regularization encourages the variable values in a group to be zero or non-zeros; tree sparsity [113] encourages the variables in a hierarchical tree structure (see Fig. 1) on a tree to be zero or non-zeros; graph sparsity [114] enforces the structure formed by the node variables and the involved edges to be zeros or not. Structure sparsity will make the models easier for interpretation and cheaper to use. Moreover, prior knowledge can be incorporated to make the model sparse.

In this section, our goal is to solve a more generic structure regularization problem (i.e., group, tree, graph or even forest) with overlapping groups, which means these different groups can be overlapped, and each input variable can be simultaneously attributed to multiple groups by incorporating different prior knowledge [9], which allows an arbitrary combinations of features. This problem is more challenging due to the non-separability of penalties, where standard lasso cannot be easily applied. Moreover, generic solver (e.g., interior-point method (IPM)) is computational expensive and does not scale very well.

Many efforts (*e.g.*, [9], [115], [116], [117], [118], [119], etc) have been devoted to solve the related group lasso problem. For example, Jenatton et al [9] propose an alternating algorithm to solve the overlapping lasso problem, Chen et al. [115] present a smoothing technique to solve the overlapping group lasso problem, Argyrious et al. [118] use the proximal gradient method to solve the overlapping lasso where the proximal operator is computed through a fixed point method. Most of (if not all) these works, however, solve group lasso problem with *specific fixed* structures(e.g., tree structures, wavelet transforation coefficient relations, etc). To the best of knowledge, few algorithms can solve the group lasso problem with an arbitrary structure.

We develop an efficient algorithm for *generic* group lasso problem, which allows an arbitrary definition of structures (e.g., group, tree, graph, forest structure, etc). This

generalization makes our algorithm flexible to deal with any kinds of group lasso problem, whether groups are overlapped or not. Furthermore, an efficient algorithm is developed to solve this generic group lasso problem, where a key step is to solve the proximal operator using an efficient method. To summarize, the main contribution of this paper is listed as follows.

- An efficient algorithm is employed to solve general group lasso problem, to satisfy different kinds of structure requirement (i.e., group, tree, graph, forest, etc).
- An iteratively reweighted algorithm is provided, to solve the proximal operator, which is the main contribution of this work. Its convergence is rigorously proved. The proposed algorithm is simple and effective, and thus could be useful in different settings as well.
- Experiment results on both synthetic and gene-expression datasets indicate the effectiveness and efficiency of proposed algorithm.

The remainder of this section is organized as follows. Section 4.2 states the motivation of this work, and presents models to be solved in this work. Section 4.3 gives the overview of the proximal gradient method to solve this problem while Section 4.4 presents the efficient method to solve the proximal operator. In Section 4.5, we give an extension strategy to our algorithm followed by a general auxiliary function based method in Section 4.6. Section 4.7 discusses the related works. Section 4.8 shows experimental results and finally we draw concluding remarks in Section 4.9.

Notations.

In this paper, all matrices are written as boldface uppercase, vectors are written as boldface lowercase, and scalars are denoted by lower-case letters (a, b). n is the number of data points, p is the dimension for data, and k is the number of class for each dataset. For any vector $\mathbf{x} \in \Re^p$, L_q norm of $\mathbf{w}$ is $\|\mathbf{w}\|_q = \left(\sum_{i=1}^{p} |w_j|^q \right)^{\frac{1}{q}}$ for $q \in [1, \infty]$, and $\|\mathbf{w}\|_\infty = \max_{1 \le j \le p} |w_j|$.

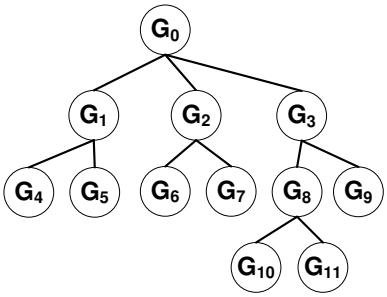


Figure 4.1: An example of overlapping tree structure with variable index on each node. Root group $\mathcal{G}_0 = \{1-10\}$, depth-1 nodes include groups $\mathcal{G}_1 = \{1,2\}, \mathcal{G}_2 = \{3,4,5,6\}, \mathcal{G}_3 = \{7,8,9,10\}$, depth-2 node include groups $\mathcal{G}_4 = \{1\}, \mathcal{G}_5 = \{2\}, \mathcal{G}_6 = \{3,4\}, \mathcal{G}_7 = \{5,6\}, \mathcal{G}_8 = \{7,8,9\}, \mathcal{G}_9 = \{10\}$, and depth-3 nodes include groups $\mathcal{G}_{10} = \{7,8\}, \mathcal{G}_{11} = \{9\}$.

We first give the definitions of groups. Let $\mathbf{x}_{\mathcal{G}_g} \subseteq \{x_1, x_2, \cdots, x_p\}$ contains the g-th group of features, and $\mathcal{G}_g$ is the index for the g-th group of features. For example, if $\mathcal{G}_g = \{1,2,4\}$, $\mathbf{x}_{\mathcal{G}_g} = \{x_1, x_2, x_4\}$, then

$$\|\mathbf{x}_{\mathcal{G}_g}\|_2 = \sqrt{x_1^2 + x_2^2 + x_4^2},$$

$$\|\mathbf{x}_{\mathcal{G}_g}\|_\infty = \max(|x_1|, |x_2|, |x_4|).$$

See a tree structure described in Fig.4.1. It two groups have overlapped elements/features, we say they are overlapped. For example, group $\mathcal{G}_2$ and $\mathcal{G}_7$ have two overlapped features: $5, 6$ because $\mathcal{G}_2 = \{3,4,5,6\}$, $\mathcal{G}_7 = \{5,6\}$.

4.2 Generic group lasso problem

In this paper, we consider the following generic group lasso penalized problem:

$$\min_{\mathbf{x}\in\Re^p} g(\mathbf{x}) = f(\mathbf{x}) + \phi_{\lambda_1,\lambda_2}(\mathbf{x}), \tag{4.1}$$

where $f(\mathbf{x})$ is a smooth convex loss function over input variable $\mathbf{x} \in \Re^p$ (e.g., least square loss), and

$$\phi_{\lambda_1,\lambda_2}(\mathbf{x}) = \lambda_1||\mathbf{x}||_1 + \lambda_2 \sum_{g=1}^{G} w_g ||\mathbf{x}_{\mathcal{G}_g}||_q \tag{4.2}$$

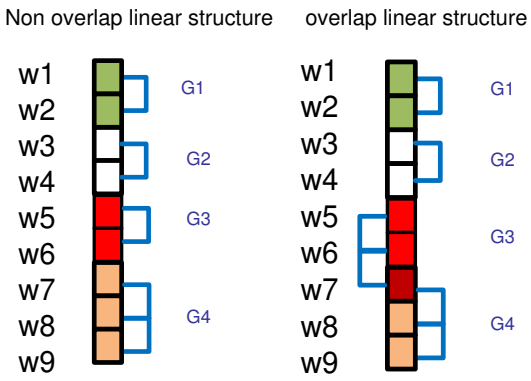


Figure 4.2: An example of linear structure, with variable index on each node. Left: non-overlap linear structure,$\mathcal{G}_1 = \{1, 2\}, \mathcal{G}_2 = \{3, 4\}, \mathcal{G}_3 = \{5, 6\}, \mathcal{G}_4 = \{7, 8, 9\}$; Right:overlap linear structure, $\mathcal{G}_1 = \{1, 2\}, \mathcal{G}_2 = \{3, 4\}, \mathcal{G}_3 = \{5, 6, 7\}, \mathcal{G}_4 = \{7, 8, 9\}$.

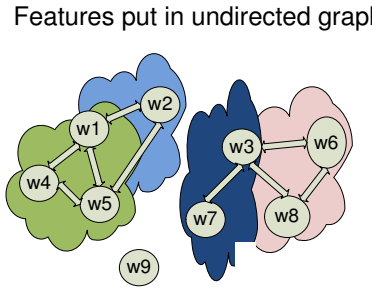


Figure 4.3: An example of feature constraint on undirected graph. Each group is the maximum clique on undirected graph. $\mathcal{G}_1 = \{1, 2, 5\}, \mathcal{G}_2 = \{1, 4, 5\}, \mathcal{G}_3 = \{3, 7\}, \mathcal{G}_4 = \{3, 6, 8\}$.

is the group lasso penalty, and $\lambda_1 \geq 0$, $\lambda_2 \geq 0$ are regularization parameters, $w_g > 0$ is a scalar, $q = \{2, \infty\}$, $1 \leq g \leq G$, G is the number of groups. $\mathbf{x}_{\mathcal{G}_g} \subseteq \{x_1, x_2, \cdots, x_p\}$ contains the g-th group of features, and $\mathcal{G}_g$ is the index for the g-th group of features.

Different forms of group, tree or graph structures can be reformulated in this general form, such as [113, 108], etc. A motivating example of tree structure is shown in Fig.(4.1), where G =12 groups are displayed, associated with p =10 variables.

Motivation of our formulation of Eq.(4.1)

The first term $||\mathbf{x}||_1$ promotes the flat sparsity of solution, which is standard LASSO problem [11]. The second term $\sum_{g=1}^{G} w_g ||\mathbf{x}_{\mathcal{G}_g}||_q$ selects a number of groups, which enforces

the structure sparsity [10]. Thus, Eq.(4.1) jointly enforces flat sparsity and structure sparsity, which selects features in both individual and group-wise way.

Actually the first term of Eq.(4.2) can be resolved into the second term, which introduces p additional groups. The g groups are specified and they may overlap. The penalty of Eq.(4.2) is a special case of more general Composite Absolute Penatly (CAP) family [120]. When groups are disjoint with $\lambda_1 = 0$ and $\lambda_2 \geq 0$, the model of Eq.(4.1) reduces to the group lasso [107]. If $\lambda_1 > 0$ and $\lambda_2 = 0$, the model of Eq.(4.1) reduces to standard Lasso [11]. The above objective function of Eq.(4.2) has the benefit of being convex, which eliminates the possibility of convergence to a local minimum.

Extension to non-convex group lasso using L_r norm

Further we can enforce more sparse L_r norm ($0 < r < 1$) to encourage structure sparsity. Because $\sum_{g=1}^{G} ||\mathbf{x}_{\mathcal{G}_g}||_q$ approximates the *true* number of groups we want to select: $\sum_{g=1}^{G} s(z)$, where $z = ||\mathbf{x}_{\mathcal{G}_g}||_q$, and the scalar number function $s(\cdot)$ is defined as:

$$s(z) = \begin{cases} 1; & \text{if } z \neq 0 \\ 0; & \text{if } z = 0 \end{cases} \tag{4.3}$$

We approximate the number function $s(\cdot)$ by $f_r(z) = z^r$ at small r [50]. Clearly, $z^{0.2}$ is a better approximation of $r(z)$ than $f(z) = z^{0.5}$. Thus parameter r is usually set to $0 \leq r \leq 1$. In general $r > 1$ case is un-interesting.

Using function $f_r(z) = z^r (0 \leq r \leq 1)$, this leads to several other versions of general group lasso problem,

$$\phi_{\lambda_1,\lambda_2}(\mathbf{x}) = \lambda_1 ||\mathbf{x}||_1 + \lambda_2 \sum_{g=1}^{G} w_g (||\mathbf{x}_{\mathcal{G}_g}||_2)^r, \tag{4.4}$$

$$\phi_{\lambda_1,\lambda_2}(\mathbf{x}) = \lambda_1 ||\mathbf{x}||_1 + \lambda_2 \sum_{g=1}^{G} w_g (||\mathbf{x}_{\mathcal{G}_g}||_\infty)^r. \tag{4.5}$$

Compared to Eq.(4.2), Eq.(4.2) can be viewed as a special case at $r = 1$. However, one drawback of this approach is that, at $0 \leq r < 1$, Eq.(4.4) and Eq.(4.5) are not convex any more.

4.3 Solving objective using proximal gradient method

In this paper, we employ accelerated proximal gradient method [111, 109] for solving Eq.(4.1) at $q = 2$ due to its fast convergence. $q = \infty$ case and non-convex case of Eq.(4.4) and Eq.(4.5) can be similarly solved. Due to space limit, we omit the detailed algorithms here.

The key step in computation of Eq.(4.1) is the computation of proximal operator. We present an effective algorithm to compute the proximal operator, which is well illustrated in Section 4.4.

A brief overview of proximal gradient method

Proximal method firstly constructs a model for approximating $f(.)$ at the point $\mathbf{x}_0$, such that

$$f(\mathbf{x}) = \Big(f(\mathbf{x}_0) + \langle \nabla f(\mathbf{x}_0), \mathbf{x} - \mathbf{x}_0 \rangle\Big) + \phi_{\lambda_1,\lambda_2}(\mathbf{x}) + \frac{L}{2}||\mathbf{x} - \mathbf{x}_0||^2,$$

where $L > 0$. This model $f(\mathbf{x})$ consists of the first-order Taylor expansion of the function $f(.)$ at $\mathbf{x}_0$, the non-smooth penalty $\phi_{\lambda_1,\lambda_2}(\mathbf{x})$, and another regularization term $\frac{L}{2}||\mathbf{x} - \mathbf{x}_0||^2$. Note f is a function: $\mathbb{R}^{n\times p} \to \mathbb{R}^n$. Its gradient is Lipschitz continuous if

$$||\nabla f(\mathbf{x}_1) - \nabla f(\mathbf{x}_2)|| \leq L\|\mathbf{x}_1 - \mathbf{x}_2\|,$$

for any $\mathbf{x}_1, \mathbf{x}_2 \in \Re^p$, where L is a constant. If f is gradient Lipschitz continuous, the following holds [111],

$$f(\mathbf{x}) \leqslant f(\tilde{\mathbf{x}}) + \langle \mathbf{x} - \tilde{\mathbf{x}}, \nabla f(\tilde{\mathbf{x}}) \rangle + \frac{L}{2}||\mathbf{x} - \tilde{\mathbf{x}}||^2.$$

More detailed introduction of proximal method can be found in [121, 111].

Let

$$J_{L_t,\delta_t}(\mathbf{x}) = \frac{L_t}{2}||\mathbf{x} - \delta_t||^2 + \phi_{\lambda_1,\lambda_2}(\mathbf{x}). \tag{4.6}$$

Next, a sequence of approximation solutions $\mathbf{x}$ is computed as follows: $\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}} J_{L_t,\delta_t}(\mathbf{x})$ where the search points δ_t is an affine combination of $\mathbf{x}_{t-1}$ and $\mathbf{x}_t$ as $\delta_t = \mathbf{x}_t + \beta_t(\mathbf{x}_t - \mathbf{x}_{t-1})$. Coefficient β_t is updated in each step, and step size L_t can be determined by the line search algorithm, such as [111] to preserve $f(\mathbf{x}_{t+1}) \leq J_{L_t,\mathbf{x}_t}(\mathbf{x}_{t+1})$. One key step is to minimize proximal operator of Eq.(4.6) [121], which is equivalent to optimizing the following Eq.(4.7),

$$\min_{\mathbf{x}} \frac{L_t}{2}||\mathbf{x} - \mathbf{a}_t||^2 + \phi_{\lambda_1,\lambda_2}(\mathbf{x}), \tag{4.7}$$

where $\mathbf{a}_t = \mathbf{x}_t - \frac{1}{L_t}\nabla f(\mathbf{x}_t)$, given current data $\mathbf{x}_t$.

Detailed Algorithm

We summarize the detailed algorithm in Algorithm 1. Basically, this algorithm really has nothing to do with the particular overlapping group lasso operator of Eq.(4.2), except step 4 and 10.

Convergence analysis

Suppose $\mathbf{x}^*$ is optimal solution for Eq.(4.1), then $\mathbf{x} \in \Re^p$ is called an ϵ-optimal solution to Eq.(4.1) if $g(\mathbf{x}) - g(\mathbf{x}^*) \leq \epsilon$ holds. It is known that proximal gradient method [122] can achieve ϵ-optimal solution in $\mathcal{O}(\frac{1}{\sqrt{\epsilon}})$ iterations.

Theorem 6. *[122] Let $\{\mathbf{x}^t\}$ be the sequence generated by proximal gradient method of Alg.1, then for any $t \geq 1$, we have,*

$$g(\mathbf{x}^t) - g(\mathbf{x}^*) \leq \frac{2L_f\|\mathbf{x}^0 - \mathbf{x}^*\|^2}{(1+t)^2}, \quad \forall \mathbf{x}^* \in \mathbf{X}_*.$$

The above theorem establishes the fast convergence of proximal gradient method. The efficiency of proximal gradient method is depending on the computation of proximal operator in Step 4/10 shown in Alg.1. As long as the proximal operator computation step

Algorithm 4 Proximal method for overlapping group lasso

Input: L_0, $\mathbf{x}_0$, δ_0

Output: $\mathbf{x}_{t+1}$

initialize $L_1 \leftarrow L_0$, $\mathbf{x}_1 \leftarrow \mathbf{x}_0$, $\delta_1 \leftarrow \delta_0$, $t \leftarrow 1$
while not converge **do**
  $\mathbf{a}_t \leftarrow \mathbf{x}_t - \frac{1}{L_t}\nabla f(\mathbf{x}_t)$.
  solve for $\mathbf{x}_{t+1} = \min_{\mathbf{x}} \frac{L_t}{2}||\mathbf{x} - \mathbf{a}_t||^2 + \phi_{\lambda_1,\lambda_2}(\mathbf{x})$ of Eq.(4.8)
  if $f(\mathbf{x}_{t+1}) < J_{L_t,\mathbf{a}_t}(\mathbf{x}_t)$ **then**
    $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_{t+1} + \frac{2(\delta_t - 1)}{1+\sqrt{1+4\delta_t^2}}(\mathbf{x}_{t+1} - \mathbf{x}_t)$,
    $\delta_{t+1} \leftarrow \frac{1+\sqrt{1+4\delta_t^2}}{2}$,
    $L_{t+1} \leftarrow L_t, t \leftarrow t+1$
  else
    search for smallest L_t, such that $\mathbf{x}_{t+1} = \min_{\mathbf{x}} \frac{L_t}{2}||\mathbf{x}-\mathbf{a}_t||^2+\phi_{\lambda_1,\lambda_2}(\mathbf{x})$ of Eq.(4.8) satisfies $f(\mathbf{x}_{t+1}) < J_{L_t,\mathbf{a}_t}(\mathbf{x}_{t+1})$
  end if
end while

gives good performance, we will achieve good performance. In the following, we will show an efficient algorithm to solve the associated proximal operator in each iteration.

4.4 An efficient algorithm for associated proximal operator computation

In this section, we show how to solve the associated proximal operator. Eq.(4.7) is equivalent to solving:

$$J_1(\mathbf{x}) = \frac{1}{2}||\mathbf{x} - \mathbf{a}||^2 + \beta_1||\mathbf{x}||_1 + \beta_2 \sum_{g=1}^{G} w_g ||\mathbf{x}_{\mathcal{G}_g}||, \tag{4.8}$$

where the correspondence between Eq.(4.8) and Eq.(4.7) is $\mathbf{a} = \mathbf{a}^t$, $\beta_1 = \lambda_1/L_t, \beta_2 = \lambda_2/L_t, q = 2$. In the following, we provide an iterative algorithm to solve Eq.(4.8), and then prove its convergence.

Let $\mathtt{I}_{\mathcal{G}_g} \in \{0,1\}^{p\times 1}$ be group index indicator for group $g(1 \le g \le G)$. For example, group 1 is $\sqrt{x_1^2 + x_2^2}$, then $\mathtt{I}_{\mathcal{G}_1} = [1, 1, 0, \cdots, 0]$. Thus the group variable $\mathbf{x}_g$ can be explicitly expressed as $\mathbf{x}_{\mathcal{G}_g} = \mathrm{diag}(\mathtt{I}_{\mathcal{G}_g}) \times \mathbf{x}$.

Key idea of Algorithm

The high level idea of proposed algorithm is to update $\mathbf{x}$ iteratively, until the algorithm converges. The similar idea is widely used in spare coding [11] or compressive sensing [85], which is known as *iteratively reweighted method* [123], [124].

Procedure

Instead of directly optimizing Eq.(4.8), we propose to optimize the following objective, i.e.,

$$J_2(\mathbf{x}) = \frac{1}{2}||\mathbf{x} - \mathbf{a}||^2 + \frac{1}{2}\beta_1 \mathbf{x}^T \mathbf{E}\mathbf{x} + \frac{1}{2}\beta_2 \mathbf{x}^T \mathbf{F}\mathbf{x}, \tag{4.9}$$

where $\mathbf{E}, \mathbf{F} \in \Re^{p\times p}$ are both diagonal matrices, and given by

$$\mathbf{E}_{ii} = \frac{1}{|\mathbf{x}_i|}, \quad \mathbf{F}_{ii} = \Big(\sum_{g=1}^{G} \frac{w_g \mathbf{I}_{\mathcal{G}_g}}{||\mathbf{x}_{\mathcal{G}_g}||}\Big)_i. \tag{4.10}$$

Note computation of $\mathbf{E}, \mathbf{F}$ depends on $\mathbf{x}$, thus minimization of $\mathbf{x}$ depends on both $\mathbf{E}, \mathbf{F}$. In the following, we propose an efficient algorithm to find out the optimal global solution for $\mathbf{x}$, where in each iteration, $\mathbf{x}$ is updated using current $\mathbf{E}, \mathbf{F}$, and $\mathbf{E}, \mathbf{F}$ are updated using current $\mathbf{x}$. This process is iterated until the algorithm converges.

Taking the derivative of Eq.(4.9) w.r.t $\mathbf{x}$ and set it to zero. We have

$$\frac{\partial J_2}{\partial \mathbf{x}} = \mathbf{x} - \mathbf{a} + \beta_1 \mathbf{E}\mathbf{x} + \beta_2 \mathbf{F}\mathbf{x} = 0. \tag{4.11}$$

Thus we obtain the optimal solution for Eq.(4.9) is given by,

$$\mathbf{x} = (\mathbf{I} + \beta_1 \mathbf{E} + \beta_2 \mathbf{F})^{-1}\mathbf{a}. \tag{4.12}$$

Note $\mathbf{E}, \mathbf{F}$ are all diagonal matrix, and thus the $(\mathbf{I} + \beta_1 \mathbf{E} + \beta_2 \mathbf{F})^{-1}$ can be efficiently computed, where $\mathbf{I} \in \Re^{p\times p}$ is identity matrix. Let $\mathbf{e} \in \Re^p, \mathbf{f} \in \Re^p$,

$$\mathbf{Q} = (\mathbf{I} + \beta_1 \mathbf{E} + \beta_2 \mathbf{F})^{-1},$$

where $\mathbf{e}_i = \frac{1}{|\mathbf{x}_i|}, \mathbf{f}_i = \Big(\sum_{g=1}^{G} \frac{w_g \mathbf{I}_{\mathcal{G}_g}}{||\mathbf{x}_{\mathcal{G}_g}||}\Big)_i$, then $\mathbf{Q}$ is a diagonal matrix and given by,

$$\mathbf{x} = \mathbf{Q}\mathbf{a}, \quad \mathbf{Q}_{ii} = \frac{1}{1 + \beta_1 \mathbf{e}_i + \beta_2 \mathbf{f}_i}. \tag{4.13}$$

Using the updating rule of Eq.(4.13), we can obtain the global optimal solution for Eq.(4.8). We call our algorithm *iterative reweighted* method due to iteratively updating of $\mathbf{x}, \mathbf{E}, \mathbf{F}$ in each iteration. To summarize, Algorithm 2 gives the entire updating process of each individual variable $\mathbf{x}$, $\mathbf{E}$, $\mathbf{F}$. In other words, Algorithm 2 monotonically decreases the objective of the problem in Eq.(4.8) in each iteration.

Algorithm 5 Iterative reweighted algorithm to solve Eq.(4.8)

Input: $\mathbf{a}$, β_1, β_2

Output: $\mathbf{x}$

Procedure:

1: $t = 0$.

2: **while** Not converge **do**

3: Update $\mathbf{x}^t$ using Eq.(4.13).

4: Update $\mathbf{e}_i^t = \frac{1}{|\mathbf{x}_i^t|}$

5: Update $\mathbf{f}_i^t = \left(\sum_{g=1}^{G} \frac{w_g \mathtt{I}_{\mathcal{G}_g}}{||\mathbf{x}_{\mathcal{G}_g}^t||} \right)_i$

6: $t = t + 1$.

7: **end while**

Time complexity analysis

We analyze the time complexity of proposed algorithm. In practice, we find the above algorithm converges very fast. Typically, it converges to global optimal solution within 10-20 times (to precision 1e-6). In each iteration, we need to update $\mathbf{x}, \mathbf{e}, \mathbf{f}$ iteratively, which takes time cost $\mathcal{O}(p), \mathcal{O}(p), \mathcal{O}(pG)$, respectively. Thus the total time cost of each iteration is $\mathcal{O}(pG)$, which is linear w.r.t the number of features and the number of groups.

A nice property of our algorithm is that, it can handle different kinds of group structure, no matter whether they are overlapped or not. The performance of our algorithm is determined by the number of features and the number of groups.

Convergence of algorithm

Theorem 7. *Under the updating rule of Eq.(4.12),* $J_1(\mathbf{x}^{t+1}) - J_1(\mathbf{x}^t) \leq 0$.

To prove Theorem 4.1, we need two lemmas.

Lemma 8. *Under the updating rule of Eq.(4.12),* $J_2(\mathbf{x}^{t+1}) < J_2(\mathbf{x}^t)$.

Lemma 9. *Under the updating rule of Eq.(4.12),*

$$\Big(J_1(\mathbf{x}^{t+1}) - J_1(\mathbf{x}^t)\Big) \leq \Big(J_2(\mathbf{x}^{t+1}) - J_2(\mathbf{x}^t)\Big). \tag{4.14}$$

Proof of Theorem 4.1.

Proof. From Lemma 4.2 and Lemma 4.3, it is easy to see $\Big(J_1(\mathbf{x}^{t+1}) - J_1(\mathbf{x}^t)\Big) \leq 0$.This completes the proof. □

Two useful propositions.

Before proof of Lemma 4.2, we present two properties satisfied in our algorithm. Proposition 1 is obvious, and we defer the presentation of proof of proposition 2 after proof of Lemma 4.3.

Proposition 1

$||\mathbf{x}||_1 = \mathbf{x}^T\mathbf{E}\mathbf{x}$.

Proposition 2

$\mathbf{x}^T\mathbf{F}\mathbf{x} = \sum_{g=1}^{G} w_g||\mathbf{x}_{\mathcal{G}_g}||$.

Proof of Lemma 4.2

Proof. Eq.(4.9) is a convex function, and optimal solution of Eq.(4.12) is obtained by taking derivative $\frac{\partial J_2}{\partial \mathbf{x}} = 0$, thus obtained $\mathbf{x}^*$ is global optimal solution, $J_2(\mathbf{x}^{t+1}) < J_2(\mathbf{x}^t)$. □

Proof of Lemma 4.3

Proof. Let $\Delta =$ LHS -RHS of Eq.(4.14). We have $\Delta = \beta_1\alpha + \beta_2\gamma$, where

$$\begin{aligned}
\alpha &= ||\mathbf{x}^{t+1}||_1 - ||\mathbf{x}^t||_1 - \frac{1}{2}(\mathbf{x}^{t+1})^T\mathbf{E}^t\mathbf{x}^{t+1} + \frac{1}{2}(\mathbf{x}^t)^T\mathbf{E}^t\mathbf{x}^t \\
&= ||\mathbf{x}^{t+1}||_1 - \frac{1}{2}||\mathbf{x}^t||_1 - \frac{1}{2}(\mathbf{x}^{t+1})^T\mathbf{E}^t\mathbf{x}^{t+1} \\
&= ||\mathbf{x}^{t+1}||_1 - \frac{1}{2}||\mathbf{x}^t||_1 - \frac{1}{2}\sum_i \frac{|x_i^{t+1}|^2}{|x_i^t|} \\
&= -\frac{1}{2}\sum_i \frac{\left(|x_i^{t+1}| - |x_i^t|\right)^2}{|x_i^t|} \leq 0.
\end{aligned} \tag{4.15}$$

$$\begin{aligned}
\gamma &= \sum_{g=1}^{G} w_g(||\mathbf{x}_{\mathcal{G}_g}^{t+1}|| - ||\mathbf{x}_{\mathcal{G}_g}^t||) - w_g{\mathbf{x}^{t+1}}^T\mathbf{F}^t\frac{\mathbf{x}^{t+1}}{2} + w_g{\mathbf{x}^t}^T\mathbf{F}^t\frac{\mathbf{x}^t}{2} \\
&= \sum_{g=1}^{G} w_g||\mathbf{x}_{\mathcal{G}_g}^{t+1}|| - \frac{1}{2}\sum_{g=1}^{G} w_g||\mathbf{x}_{\mathcal{G}_g}^t|| - \frac{1}{2}w_g(\mathbf{x}^{t+1})^T\mathbf{F}^t\mathbf{x}^{t+1} \\
&= \sum_{g=1}^{G} w_g||\mathbf{x}_{\mathcal{G}_g}^{t+1}|| - \frac{1}{2}\sum_{g=1}^{G} w_g||\mathbf{x}_{\mathcal{G}_g}^t|| - \frac{1}{2}\sum_{g=1}^{G}\frac{w_g}{||\mathbf{x}_{\mathcal{G}_g}^t||}||\mathbf{x}_{\mathcal{G}_g}^{t+1}||^2 \\
&= -\frac{1}{2}\sum_{g=1}^{G}\frac{w_g(||\mathbf{x}_{\mathcal{G}_g}^t - \mathbf{x}_{\mathcal{G}_g}^{t+1}||)^2}{||\mathbf{x}_{\mathcal{G}_g}^t||} \leq 0.
\end{aligned} \tag{4.16}$$

Due to proposition 1, Eq.(4.15) holds. Due to proposition 2, Eq.(4.16) holds. Clearly, $\Delta = \beta_1\alpha + \beta_2\gamma \leq 0$. □

Proof of Proposition 2

$$\begin{aligned}
\mathbf{x}^T\mathbf{F}\mathbf{x} = \sum_{i=1}^{p}\mathbf{x}_i^2\Big(\sum_{g=1}^{G}\frac{w_g\mathtt{I}_{\mathcal{G}_g}}{||\mathbf{x}_{\mathcal{G}_g}||}\Big)_i &= \sum_{g=1}^{G}\frac{w_g}{||\mathbf{x}_{\mathcal{G}_g}||}\sum_{i=1}^{p}\mathbf{x}_i^2(\mathtt{I}_{\mathcal{G}_g})_i \\
&= \sum_{g=1}^{G}\frac{w_g}{||\mathbf{x}_{\mathcal{G}_g}||}||\mathbf{x}_{\mathcal{G}_g}||^2 = \sum_{g=1}^{G} w_g||\mathbf{x}_{\mathcal{G}_g}||.
\end{aligned}$$

The proposed algorithm can also be used to solve a more general loss function w.r.t to group lasso, i.e.,

$$J(\mathbf{x}) = f(\mathbf{x}) + \beta\sum_{g=1}^{G} w_g||\mathbf{x}_{\mathcal{G}_g}||_\sigma, \tag{4.17}$$

where $f(\mathbf{x})$ is a loss function (e.g., least square) w.r.t data variables and class labels, and $||\mathbf{x}_{\mathcal{G}_g}||_\sigma$ could be an adaptive loss function [54] w.r.t an adaptive parameter σ, which can smoothly interpolate between L_1 loss and L_2 loss. For example, for a vector $\mathbf{x} \in \Re^p$,

$$\|\mathbf{x}\|_\sigma = \sum_i \frac{(1+\sigma)x_i^2}{|x_i| + \sigma}. \tag{4.18}$$

Eq.(4.17) can be similarly solved as Eq.(4.8).

4.5 Acceleration to the proposed algorithm

One strategy to accelerate our algorithm is to use the following theorem [119]:

Theorem 8. *Let $\mathbf{x}^*$ be optimal solution for Eq.(4.8), then $\mathbf{x}^*$ is also the optimal solution for:*

$$J_3(\mathbf{x}) = \min_{\mathbf{x}} \frac{1}{2}||\mathbf{x} - \mathbf{b}||^2 + \beta_2 \sum_{g=1}^{G} w_g ||\mathbf{x}_{\mathcal{G}_g}||, \tag{4.19}$$

where $\mathbf{b} = sign(\mathbf{a}) \otimes \max(|\mathbf{a}| - \beta_1, 0)$, $\otimes$ represents element-wise multiplication.

According to Theorem 8, we can solve Eq.(4.8) through Eq.(4.19), by setting $\mathbf{b} = sign(\mathbf{a}) \otimes \max(|\mathbf{a}| - \beta_1, 0)$. The algorithm is very similar to Algorithm 2 as shown above, without updating $\mathbf{e}_i$ in step 4.

4.6 Extension to General Loss function

If we have concrete examples of loss function in Eq.(4.1)(*e.g.*, least square loss). Algorithm 2 can be directly used to solve this problem without employing standard proximal gradient method. Let

$$J_4(\mathbf{x}) = \min_{\mathbf{x}} ||\mathbf{y} - \mathbf{D}\mathbf{x}||^2 + \phi_{\lambda_1,\lambda_2}(\mathbf{x}), \tag{4.20}$$

where $\mathbf{x} \in \Re^p$ is the regression coefficient, $\mathbf{y} = [y_1, y_2, \cdot, \cdot, \cdot, y_n]^T \in \Re^{n\times 1}$ is the output, $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_n]^T \in \Re^{n\times p}$ is input data. We can use more robust error function, i.e.,

$$J_5(\mathbf{x}) = \min_{\mathbf{x}} \sum_{i=1}^{n} ||y_i - \mathbf{d}_i\mathbf{x}|| + \phi_{\lambda_1,\lambda_2}(\mathbf{x}). \tag{4.21}$$

To solve Eq.(4.20) at $q = 2$, consider the following objective,

$$J_6(\mathbf{x}) = \min_{\mathbf{x}} ||\mathbf{y} - \mathbf{D}\mathbf{x}||^2 + \frac{1}{2}\lambda_1 \mathbf{x}^T \mathbf{E}\mathbf{x} + \frac{1}{2}\lambda_2 \mathbf{x}^T \mathbf{F}\mathbf{x}, \tag{4.22}$$

where $\mathbf{E}, \mathbf{F}$ are the same as those defined in Eq.(4.10). Then updating rules for $\mathbf{x}$ is given by,

$$\mathbf{x} = (2\mathbf{D}^T\mathbf{D} + \lambda_1\mathbf{E} + \lambda_2\mathbf{F})^{-1} 2\mathbf{D}^T\mathbf{y}, \tag{4.23}$$

where $\mathbf{E}, \mathbf{F}$ can be updated as that of Eq.(4.10) shown in Algorithm 2. The convergence of the algorithm can be easily proved as that of Algorithm 2. Due to space limit, we omit the details here.

Extension to general case for multi-class classification

Above we have discussed how to use group lasso for variable selection purpose. In practice especially for multi-class classification problem, we have original dataset $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_n]$ with p features for n data points, with associated class labels $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n]$ from c classes, where $\mathbf{D} \in \Re^{p\times n}$, $\mathbf{Y} \in \Re^{c\times n}$. We note previous works [118, 125, 56] have considered the problem of variable/feature selection based on structural sparsity (e.g., using $L_{2,1}$-norm for group sparsity),

$$\min_{\mathbf{W}} f(\mathbf{W}) + \alpha \sum_{i=1}^{p} \sqrt{\sum_{j=1}^{k} \mathbf{W}_{ij}^2} \tag{4.24}$$

where $\mathbf{W} \in \Re^{p\times c}$ is multi-class regression coefficient, and $f(\mathbf{W})$ is a function: $\Re^{p\times n} \mapsto \Re^{c\times n}$, e.g., least square loss $f(\mathbf{W}) = \|\mathbf{Y} - \mathbf{W}^T\mathbf{X}\|_F^2$. This can be viewed as a special case of the general group lasso problem, where there are p groups and the i-th group is the i-th row of $\mathbf{W}$, i.e., $\sum_{i=1}^{p} \sqrt{\sum_{j=1}^{k} \mathbf{W}_{ij}^2} = \sum_{g=1}^{G} \|\mathbf{W}_{\mathcal{G}_g}\|_2$, where $\|\mathbf{W}_{\mathcal{G}_i}\|_2 = \sqrt{\sum_{j=1}^{k} \mathbf{W}_{ij}^2}$.

This indicates the proposed algorithm can be used to solve multi-class/multi-label feature selection problems [53] with fast convergence. Moreover, we can further define

the group sparsity on projection $\mathbf{W}$, according to a priori knowledge, such as wavelet-coefficient [126], topographic dictionary [127], image spatial neighborhood information [128], graph embedding information [48], etc.

4.7 Connections to related works

Bach [129] analyzes the consistency of the group lasso and multiple kernel learning, which can be used for learning from heterogeneous data sources and for nonlinear variable selection. Jenatton et al [9] propose an alternating algorithm to solve the overlapping lasso problem. However, it involves an expensive matrix inversion computation and it may not scale very well. Alternating Direction method of Multipliers method (ADMM) [116] is adopted to solve this overlapping problem through the computation of a linear system. However, this approach may not scale well for large-size with high-dimension problems. Chen et al. [115] present a smoothing technique to solve the overlapping group lasso problem. Mairal et al [117] provide an algorithm to solve the sum of L_∞ norms, which can not be directly used for solving the overlapping lasso, which is defined as the sum of L_2 norms. Argyrious et al. [118] use the proximal gradient method to solve the related problem, where the proximal operator is computed through a fixed point method. Yuan et al. [119] use proximal method to solve overlapping group lasso problem, where the proximal operator is computed through solving a smooth and convex dual problem. Different from above works, our work proposes a new method for proximal operator computation, with *fast convergence* and very *simple* updating rules.

We study the group sparsity problem in the parametric setting (i.e., group lasso). Another line of research is doing sparse variable selection using nonparametric additive models (i.e., sparse additive model) [130, 131, 132], where the prior knowledge of the structure among the covariates are utilized to encourage variable selection. Most of works [130,

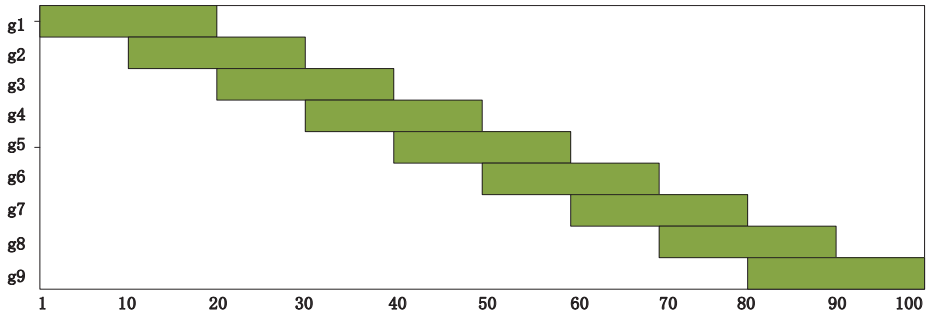


Figure 4.4: One demonstrating example of overlapping group structure. y-axis: group number, x-axis: variable index. $p = 100, G = 9$. $\mathcal{G}_1 = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{20}\}, \cdots, \mathcal{G}_9 = \{\mathbf{x}_{81}, \mathbf{x}_{82}, \cdots, \mathbf{x}_{100}\}$

131, 133], however, perform variable selection individually, where a group of basis functions constructed from a single covariate, which are greatly different from our work.

4.8 Experiment

In this section, extensive experiments are conducted to demonstrate the effectiveness of proposed algorithm. We use both synthetic datasets (including group structure, tree structure) and real world gene expression datasets to evaluate our method in various problem size and parameter settings. We compare our algorithm with several state-of-the-art methods, including SLasso algorithm [9], FoGLaaso algorithm [119], Prox-Grad method [115], and Alternating Direction Method of Multipliers (ADMM) [116].

In Eq.(4.1), we use the standard least square loss function.

Dataset

We use four datasets: (1) group structure of synthetic datasets, (2) tree structure of synthetic datasets; (3) pathway structure of gene expression datasets; (4) edge structure of gene expression datasets.

Synthetic dataset.

Group structure.

We randomly generate a overlapping adjacent group, where a group contains 20 adjacent inputs with an overlap of 10 variables between two successive groups. To be exact, $\mathcal{G}_1 = \{1, 2, ..., 20\}, \mathcal{G}_2 = \{11, 12, ..., 20\}, \mathcal{G}_g = \{J - 19, ..., J\}$ with $J = 10g + 10$, and g is the group size, J is the problem size. We show an example in Fig.4.4 where $J = 100$.

Tree structure.

We randomly generate a overlapping tree structure, which can be considered as a special graph structure. We generate an overlap tree structure (feature size $p = 2000$, group size $G = 200$), which is similar to the tree structure shown in Fig.4.1. Due to space limit, we cannot show the whole tree structure here. Part of tree has the same structure as the tree in Fig.4.1. For example, we have root group $\mathcal{G}_0 = \{1 - 2000\}$, depth-1 node include groups $\mathcal{G}_1 = \{1, 2\}, \mathcal{G}_2 = \{3, 4, 5, 6\}, \mathcal{G}_3 = \{7, 8, 9, 10\}$, etc; depth-2 nodes include groups $\mathcal{G}_{50} = \{1\}, \mathcal{G}_{51} = \{2\}, \mathcal{G}_{53} = \{3, 4\}, \mathcal{G}_{54} = \{5, 6\}$, etc; and depth-3 groups include $\mathcal{G}_{98} = \{3\}, \mathcal{G}_{99} = \{4\}$, etc.

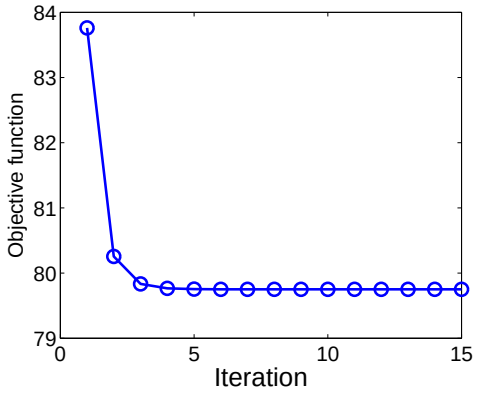
Gene expression dataset.

We perform our experiment to evaluate the efficiency of our algorithm on breast cancer gene expression dataset [134]. This dataset consists of 8141 genes in 295 breast cancer tumors (78 metastatic and 217 non-metastatic). To analyze the microarrays from the biological point of view, different methods have been used to organize the gene into overlapping gene sets. We follow [114], [119] and use two methods to generate the overlapping groups of gene sets:

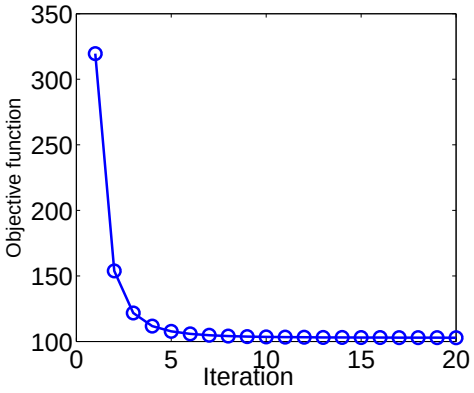
(1) pathway [135];

(2) edges [136].

For pathways, we use the canonical pathways from Molecular Signature Database (MSigDB) [135]. It contains 639 groups of genes, where 637 groups involve the genes used in pathways. The average number of genes for each group is 23.7, and the largest gene group has 213 genes. 3510 genes appear in these 637 groups with an average frequency



(a) *pathway* gene-expression



(b) *edge* gene-expression

Figure 4.5: Convergence of proximal operator computation of our algorithm (Algorithm 2) on (a) *pathway* gene-expression; (b)*edge* gene-expression. (a) Parameter setting: $p = 3510$, $G = 637$, $w = 0.5$, convergence criteria: 1e-6. (b) Parameter setting: $p = 1000$, $G = 7194$, $w = 2$, convergence criteria: 1e-6.

of number 4. We use the network [136] to extract 42,594 edges from the network, i.e., 42,594 groups of overlapping gene sets with size 2. The average frequency for 8,141 genes appeared in these 42,594 groups are 10.

Convergence of proposed proximal operator algorithm

In Figs.(4.5), we show the convergence of our algorithm (Algorithm 2) for computing the proximal operator on (1) *path-way* (2) edge gene-expression datasets. To further accelerate our algorithm, in our experiments, according to Theorem 2, we solve Eq.(4.19) instead of original problem of Eq.(4.2), using Algorithm 2. The experiment results indicate that our algorithm converges very fast to precision $(1e-6)$. It takes no more than 20 steps.

Comparisons of different methods for proximal operator.

The key advantage of our algorithm lies in the computation of the proximal operator of Eq.(4.8). There have been many methods proposed to solve it, including dual method proposed by Yuan et al [119], Dykstra-like proximal splitting method [119], and alternating direction method multipliers (ADMM) [116]. These proximal splitting methods

Table 4.1: Comparison of different proximal operator computation (Obj, CPU time) on *pathway* gene-expression dataset. Parameter setting: $p = 3510$, $G = 637$, convergence criteria: 1e-6.

method	$w = 0.5$		$w = 2$	
	Obj.	CPU/s	Obj.	CPU/s
our method	**79.7500**	**2.255**	**153.9001**	**2.402**
Dykstra	81.8974	4.267	154.7923	4.879
ADMM	82.3142	3.897	155.1287	3.912
Dual	79.7612	2.274	153.9018	2.418

Table 4.2: Comparison of different proximal operator computation (Obj, CPU time) on *edge* gene-expression dataset. Parameter setting: $p = 1000$, $G = 7194$, convergence criteria: 1e-6.

method	$w = 0.5$		$w = 2$	
	Obj.	CPU /s	Obj.	CPU/s
our method	**56.9837**	**1.163**	**102.7791**	**1.087**
Dykstra	58.9742	3.648	105.8314	3.485
ADMM	60.3214	3.124	107.2494	2.987
Dual	57.0123	1.207	102.8123	1.113

convert this challenge optimization problem into a series of sub-problem with closed-form solutions. We compare our method with above three methods.

On above four datasets, we compare the performance of above four method. There are three key parameters: (1) feature number p; (2) group number G; (3) regularization weights w. We set $w_i = w$ to the same value, even for different group regularization.

Experiment results on synthetic datasets.

We show the comparison results w.r.t different group number (G), different feature number (p), different regularization weights (w) using group structure, hierarchical tree structure on synthetic datasets in Fig. 4.6. We see that, our method achieves good performance in different parameter settings, and scales very well even when p, g is very large. Our method is better than the other methods, in terms of the computation time, at different

Table 4.3: Comparison of different algorithms for overlapping lasso computation (Obj, CPU time, iteration number) on *pathway* and *edge* gene-expression dataset. Involved genes $p = 1000$, convergence criteria: 10^{-4}.

method	pathway			edges		
	Obj.	CPU/s	Iteration	Obj.	CPU/s	Iteration
Our method	**88.11**	**9.24**	108	**117.73**	**12.30**	91
Re-weighted	88.12	12.33	123	118.34	13.29	92
FoGLasso	87.89	9.75	116	117.75	12.34	121
SLasso	88.93	11.17	142	117.98	17.81	102
Prox-Grad	88.12	12.50	138	118.10	16.53	112
ADMM	89.32	13.72	78	119.32	21.34	83

Table 4.4: Comparison of different algorithms for overlapping lasso computation (Obj, CPU time, iteration number) on *pathway* and *edge* gene-expression dataset. Involved genes $p = 2000$, convergence criteria: 10^{-4}.

method	pathway			edges		
	Obj.	CPU/s	Iteration	Obj.	CPU/s	Iteration
Our method	**131.32**	**28.92**	148	**235.75**	**29.51**	163
Re-weighted	132.46	38.65	163	235.90	39.52	201
FoGLasso	131.80	29.14	156	235.76	29.63	187
SLasso	131.34	32.74	174	236.01	31.34	198
Prox-Grad	132.57	35.02	185	235.97	32.93	212
ADMM	133.09	48.74	107	237.35	44.31	150

group number G, feature number p and regularization weight w. For Dykstra method and ADMM method, it is more sensitive to group number (G) and feature number (p).

Experiment results on gene-expression datasets.

We show the comparison results w.r.t different regularization weights (w) in Table. 4.1, Table. 4.2. Clearly, our method outperforms Dykstra method and ADMM method, and is also better than dual method, in terms of both objective function values and computational time. Moreover, our method is much simpler than dual method.

Comparisons of different computation methods for overlapping group lasso.

On the gene expression dataset, we follow [119] and use the least square loss $f(x) = \frac{1}{2}||\mathbf{y} - X\mathbf{b}||^2$, and $\lambda_1 = \lambda_2 = \gamma \times \lambda_1^{max}$, and $\lambda_1^{max} = ||X^T\mathbf{y}||_{\inf}$, and γ is chosen from the

set $\{5 \times 10^{-1}, 2 \times 10^{-1}, 1 \times 10^{-1}, ..., 1 \times 10^{-1}\}$, $w_i = 1$. We have two methods to solve the overlapping group lasso problem, (1) proximal method proposed in §2,3,4 (Our method) ; (2) general method proposed in §5 (shown as "Re-weighted" in Tables 4.3, 4.4.

We compare above two methods against other 4 methods: (1) SLasso method [9]; (2) FoGLaaso method [119]; (3) Prox-Grad method [115]; (4) Alternating Direction Method of Multipliers (ADMM) [116].

For the given γ, we run SLasso until a certain precision is reached, and then we run the other algorithms until all methods can reach to that of the SLasso. Tables 4.3, 4.4 summarize the comparison of different algorithms for overlapping lasso computation (Obj, CPU time, iteration number) when involved genes $p = 1000, p = 2000$, and convergence criteria is 10^{-4}. We make several important observations from experiment results.

(1) Our algorithm is efficient, as compared to the other methods, including FoGLasso and SLasso, in terms of CPU time.

(2) Our algorithm converges relatively faster than other methods, and always achieves smaller objective, as compared to other methods.

(3) The efficiency on *edge* dataset is greatly improved, as compared with that on *pathways*. Moreover, as the number of genes increases, the efficiency of our algorithm is also greatly improved.

(4) The results indicate that this algorithm is slower than the proximal gradient method, although our method for L_2 norm does not need a proximal step. Most of time for computation is spent on Eq.(4.23), which requires the computation of inverse of a matrix of $p \times p$.

We will further investigate how to make our algorithm more scalable and efficient for large-size problems.

Experiment results on breast cancer prediction.

Table 4.5: Classification accuracy, number of selected genes, number of selected pathways using our method (overlapping group lasso of Eq.4.2), standard lasso using 3-fold cross validation.

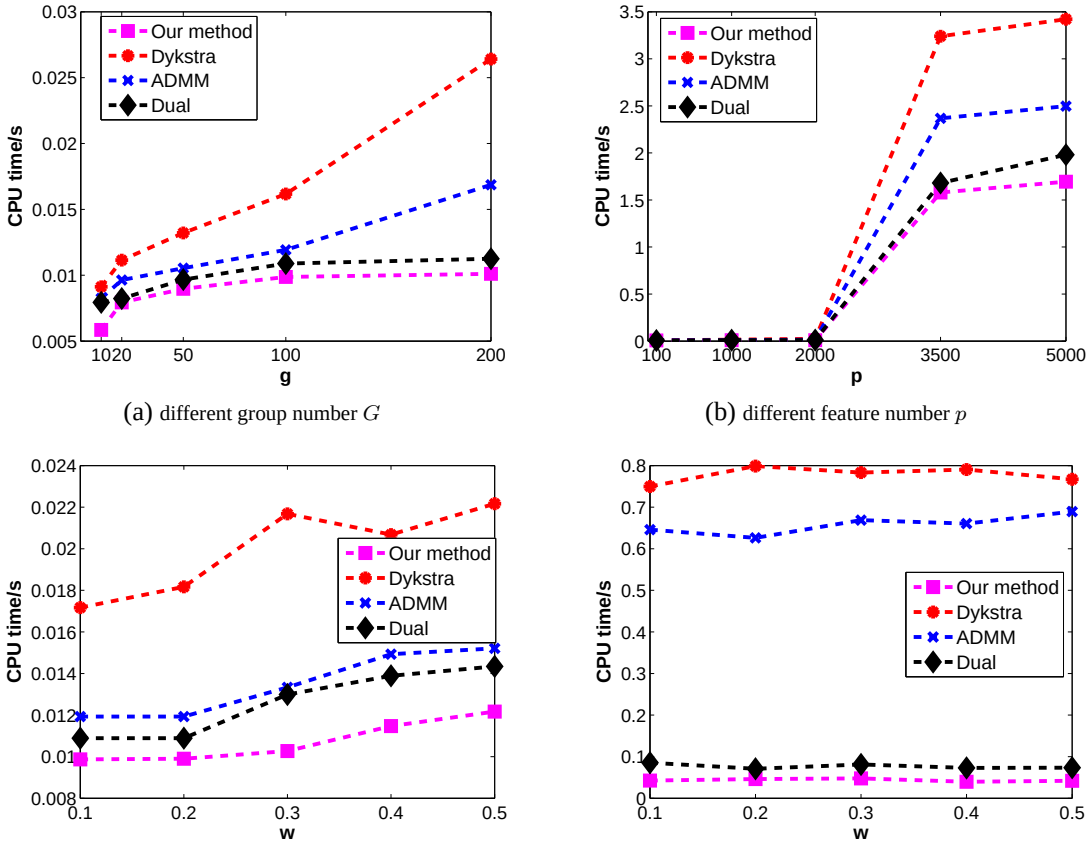
Metric	Our method	standard lasso
Classification accuracy	$65\% \pm 2\%$	$62\% \pm 2\%$
# of genes in each fold	48/57/70	109/112/143
# of pathways in each fold	238/242/167	247/267/278

Another important application of overlapping group lasso is to perform gene selection for breast cancer prediction. For example, for the breast cancer gene expression dataset [134] as shown in §4.1, we are interested in finding a group of genes, which can distinguish metastatic (78 samples) from non-metastatic (217 samples) cancer tumors. The pathway information, which involves a group of genes, conveys functionality information for cancer discovery. Different genes could be involved in different groups, and these groups may even overlap with each other. Overlap group lasso provides an easy and natural way to incorporate these prior information into cancer prediction.

On pathway dataset (see §4.1 for more details), we analyze the 3510 genes, which are in at least one pathway. Before making classification, we keep the 300 genes most correlated with the output [132]. Parameter λ is set according to cross validation. Since the dataset is very unbalanced, we balance the dataset, by making a replication of each metastasis patient in each fold in cross validation. Table 4.5 shows the comparison results of overlapping group lasso (Eq.4.2) and standard Lasso method [11]. Clearly, our method (overlapping group lasso) gives better classification accuracy. As compared to standard lasso, the solution of overlapping group lasso is more sparse at both gene and pathway level. This makes the relations between identified genes and pathways easier to interpret, which suggests using overlapping group lasso model for cancer prediction.

4.9 Lessons learned

In this paper, we propose an efficient algorithm to solve the general overlapping group lasso problem. We present an efficient algorithm to solve the associated proximal operator. Different structures can be integrated into this framework. Numerical experiments on both synthetic and gene expression datasets demonstrate the effectiveness of the proposed algorithm. An interesting future direction is to adaptive this algorithm to solve problems, which involve much richer structures among the variables (e.g., more complicated hierarchical tree structure, forest structure).



(a) different group number G

(b) different feature number p

(c) different regularization weight w

(d) different regularization weight w

Figure 4.6: Time comparison (y-axis: CPU time) for computing the proximal operators on synthetic datasets. w.r.t different p, w, G. (a-c) overlapping group structure as shown in Fig. 2; (d) tree structure which has similar hierarchical structure as shown in Fig. 1. (a) feature size $p = 1000$, regularization parameter $w = 0.1$; (b) group number $G = 100$, regularization parameter $w = 0.1$; (c) feature size $p = 1000$, group size $G = 100$; (d) feature size $p = 2000$, group size $G = 200$.

CHAPTER 5

Maximum Consistency Preferential Random Walks

Random walk plays a significant role in computer science. The popular PageRank algorithm uses random walk. Personalized random walks force random walk to "personalized views" of the graph according to users' preferences. In this paper, we show the close relations between different preferential random walks and label propagation methods used in semi-supervised learning. We further present a maximum consistency algorithm on these preferential random walk/label propagation methods to ensure maximum consistency from labeled data to unlabeled data. Extensive experimental results on 9 datasets provide performance comparisons of different preferential random walks/label propagation methods. They also indicate that our maximum consistency algorithm clearly improves the classification accuracy over existing methods.

5.1 Background of random walk

Random walk model [137] is a mathematical formalization of the paths that consists of taking successive random steps, i.e., at each step the walk jumps to another site according to some probability distribution. The random walk model plays an important role in computer science, and it has many applications in information retrieval [138], social network [139], etc. PageRank [140] is a link analysis algorithm, which uses the idea of random walk to measure the webpage's relative importance. Personalized Page Rank [141] is presented to create "personalized views" of the web searching results based on redefining importance according to users' preferences.

Semi-supervised learning (SSL) has connections with random walks on graphs. In SSL, only a small number of data points are labeled while a large number of data points are unlabeled. The goal of SSL is to classify the unlabeled data based on labeled data. SSL has attracted more attention because the acquisition of labeled data is quite expensive and time-consuming, while large amount of unlabeled data are easier to obtain. Many different approaches have been proposed to solve SSL problems [142, 143], e.g., classification-based approach [144], clustering-based approach [145], graph-based approaches [27, 28, 29], etc. Among all these approaches, graph-based approach is the most popular way to model the whole dataset as undirected weighted graph with pairwise similarities ($\mathbf{W}$), and the semi-supervised learning can be viewed as label propagation from labeled data to unlabeled data, like a random walk on a similarity-graph $\mathbf{W}$. Our work is inspired by previous graph-based semi-supervised methods, especially by the work of consistency labeling [28] and Green's function [29].

In this paper, we first show the close relationships between different preferential random walks and label propagation methods. We show that the labeled data points act as the preferential/personalized bias vectors in the personalized random walks. This provides much insight to the existing label propagation methods, and suggest ways to improve these methods. We also perform extensive experiments to compare the performance of different methods used in preferential random walks.

Furthermore, we observe that current label-propagation approach may not achieve best available results especially when the propagation operator do not exactly reveal the intrinsic structure collected from both labeled and unlabeled data points. Many label propagation approach is done in one shot from source (labeled data) to all unlabeled data. This can not guarantee many newly-labeled data, which lie far-away in the data manifold with the labeled data, are labeled reliably. Motivated by this observation, in this paper, we present a novel maximum consistency approach to improve the performance of existing

label propagation methods. Our approach first allows the label propagation from source to reliably newly-labeled data only, and progressively expands to all unlabeled data, to ensure maximum consistency from labeled data to unlabeled data. The key idea of our approach is to leverage the existing propagation operator and repeatedly utilizes it, which has almost the same computational complexity as the existing propagation methods.

Specifically, it is worthwhile to emphasize the contribution of our paper.

- We first show the relations between different preferential random walks and existing label propagation methods. Extensive experiments on 9 datasets are performed to demonstrate the performance of different methods.
- We present a maximum consistency algorithm to improve existing label-propagation methods. Extensive experiments performed on 9 datasets indicate clear performance improvement.

The rest of this paper is organized as follows. §2 gives a brief overview of personalized random walk. Next in §3, we establish the connections between the preferential random walks and label propagation methods. In §4, we emphasize the concept of score distribution in semi-supervised learning methods. In §5, we propose our maximum consistency label propagation method. §6 reviews the related work to our paper. In §7, extensive experiments on 9 datasets are performed to provide performance comparisons of different preferential random walks/label propagation methods, and demonstrate our maximum consistency algorithm results. Finally, we conclude the paper.

5.2 A brief overview of personalized random walk

On a graph with edge weights $\mathbf{W}$, $\mathbf{D}$ is a diagonal matrix with $\mathbf{D} = \mathrm{diag}(\mathbf{We}), \mathbf{e} = (1, \ldots, 1)^T$, $\mathbf{P} = [\mathbf{P}_{ij}]$ is the transition matrix from node i to node j,

$$\mathbf{P} = \mathbf{D}^{-1}\mathbf{W} \tag{5.1}$$

Let $\mathbf{f}_i$ be the probability of one walker on site i, thus

$$\mathbf{f} = (1-\alpha)\mathbf{y} + \alpha \mathbf{P}^T \mathbf{f}, \tag{5.2}$$

In PageRank([140]), $\mathbf{y} = (1, \ldots, 1)^T/n$, $\alpha = 0.9$.

In personalized random walk [141], $\mathbf{y}$ is the personalized probability (a vector) encoding the personalized preferences. For example, for a random walker, he prefers to visit sites i_1, i_2. Then $\mathbf{y}_i = 1$ if $i = i_1, i_2$; $\mathbf{y}_i = 0$ otherwise.

Generalized Preferential Random Walks

In multi-person random walks, there are K random walkers. Each random walker $k(1 \leq k \leq K)$has a distribution vector $\mathbf{f}_k$ and a personalized preference vector $\mathbf{y}_k$,

$$\mathbf{f}_k = (1-\alpha)\mathbf{y}_K + \alpha \mathbf{P}^T \mathbf{f}_K. \tag{5.3}$$

Let $\mathbf{F} = (\mathbf{f}_1, \cdots, \mathbf{f}_K)$ and $\mathbf{Y} = (\mathbf{y}_1, \cdots, \mathbf{y}_K)$, from Eq.(2), we obtain the transition

$$\mathbf{F} = (1-\alpha)\mathbf{Y} + \alpha \mathbf{P}^T \mathbf{F}. \tag{5.4}$$

The solution for the final stationary distributions of the K random walkers are

$$\mathbf{F} = \frac{1-\alpha}{\mathbf{I} - \alpha \mathbf{P}^T}\mathbf{Y}. \tag{5.5}$$

Method 1

Here we use standard random walk transition probability of Eq.(1) and obtain the stationary distributions of the K random walkers:

$$\mathbf{F} = \frac{1-\alpha}{\mathbf{I} - \alpha \mathbf{W}\mathbf{D}^{-1}}\mathbf{Y} = \frac{1-\alpha}{(\mathbf{D} - \alpha \mathbf{W})\mathbf{D}^{-1}}\mathbf{Y} = \mathbf{D}\frac{1-\alpha}{(\mathbf{D} - \alpha \mathbf{W})}\mathbf{Y}. \tag{5.6}$$

Method 2

If we use the "pseudo transition probability" $\mathbf{P} = \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$, we obtain the stationary distributions of the K random walkers as:

$$\mathbf{F} = \frac{1-\alpha}{\mathbf{I} - \alpha \mathbf{D}^{-\frac{1}{2}} \mathbf{W} \mathbf{D}^{-\frac{1}{2}}} \mathbf{Y}. \tag{5.7}$$

Method3

If we use another "pseudo transition probability" $\mathbf{P} = \mathbf{W}\mathbf{D}^{-1}$, we obtain the stationary distributions of the K random walkers as:

$$\mathbf{F} = \frac{1-\alpha}{\mathbf{I} - \alpha \mathbf{D}^{-1} \mathbf{W}} \mathbf{Y} = \frac{1-\alpha}{\mathbf{D}^{-1}(\mathbf{D} - \alpha \mathbf{W})} \mathbf{Y} = \frac{1-\alpha}{(\mathbf{D} - \alpha \mathbf{W})} \mathbf{D}\mathbf{Y}. \tag{5.8}$$

So far, we have discussed random walks on a graph. Next, we make connections to semi-supervised learning. The significance of relation analysis between preferential random walks and label propagations is to help to capture the essence of these different algorithms and better interpret the experiment results. To our knowledge, so far there is a lack of systematic study to explore the commonalities and differences of these algorithms, as well as their intrinsic relationships.

5.3 Relations between preferential random walks and Label Propagations

In semi-supervised learning, we have $n = n_\ell + n_u$ data points $\{\mathbf{x}_i\}_{i=1}^{n}$, where first n_ℓ data points are already labeled with $\{y_i\}_{i=1}^{n_\ell}$ for c target classes. Here, $\mathbf{x}_i \in \Re^p$ and $y_i \in 1, 2, ..., K$, such that $y_i = k$ if x_i belongs to the k-th class. The last n_u data are unlabeled. The goal of semi-supervised learning is to learn their class labels: $\{y_i\}_{i=n_\ell+1}^{n}$. Let $\mathbf{Y} \in \Re^{n \times K}$ be a class indicator matrix, $\mathbf{Y}_{ij} = 1$ if $\mathbf{x}_i$ is labeled as $y_i = j$ and $\mathbf{Y}_{ij} = 0$ otherwise.

Local - Global Consistency method (LGC)

Local and global consistency(LGC) [32] utilizes sufficiently smooth assumptions with respect to the intrinsic structure collectively revealed by known labeled and unlabeled

data points. Given the graph edge matrix W, LGC constructs the normalized matrix $\mathbf{S} = \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$, where D is a diagonal matrix with $\mathbf{D} = \mathrm{diag}(\mathbf{We})$. Then the predicted label matrix $\mathbf{F}$ is,

$$\mathbf{F} = \mathbf{QY}, \qquad \mathbf{Q} = \beta(\mathbf{I} - \alpha\mathbf{S})^{-1}, \tag{5.9}$$

where $\mathbf{Q}$ is the label propagation operator, $\alpha = \frac{1}{1+\mu}$, $\beta = \frac{\mu}{1+\mu}$,μ is a parameter.

Relations with preferential random walk.

Compared with method 2 in generalized preferential random walk of Eq.(7), we can see LGC is *identical* to it. This is because constant β will not change the classification results.

Green's function method (GF)

Green's function for semi-supervised learning and label propagation is first presented in [29]. GF is defined as the inverse of graph laplacian $\mathcal{L} = \mathbf{D} - \mathbf{W}$ with zero-mode discarded. Using the eigenvectors of $\mathcal{L}$: $\mathcal{L}\mathbf{v}_k = \lambda_k\mathbf{v}_k$, where $0 = \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$ are the eigenvalues. Green's function computes the predicted label matrix $\mathbf{F}$,

$$\mathbf{F} = \mathbf{QY}, \qquad \mathbf{Q} = \mathcal{L}_+^{-1} = \frac{1}{(\mathbf{D} - \mathbf{W})_+} = \sum_{i=2}^{n} \frac{\mathbf{v}_i\mathbf{v}_i^T}{\lambda_i}, \tag{5.10}$$

where $\mathbf{Q}$ is label propagation operator, $(\mathbf{D} - \mathbf{W})_+$ indicates zero eigen-mode is discarded.

Relations with preferential random walk.

From Method 1 of generalized preferential random walk, the stationary distribution $\mathbf{F}$ of Eq.(6) is related to Q in Eq.(10). As $\alpha \longrightarrow 1$, we have

$$(\mathbf{D} - \alpha\mathbf{W})^{-1} \longrightarrow (\mathbf{D} - \alpha\mathbf{W})^{+} = \sum_{i=2}^{n} \frac{\mathbf{v}_i\mathbf{v}_i^T}{\lambda_i}. \tag{5.11}$$

Indeed, for classification purpose, the GF approach is the limit of Method 1 of generalized preferential random walk of Eq.(6). This is because of the following three reasons.

(1) In semi-supervised learning, the classification result for object i is determined by the location of the largest element in i-th row(See Eq.(12)).

(2) Given a distribution $\mathbf{A}$ and a diagonal matrix $\mathbf{D} = \mathrm{diag}(d_1 \cdots d_n)$, $\mathbf{DA}$ will multiply the i-th row of $\mathbf{A}$ by d_i. The relative distribution of this row does not change. Thus $\mathbf{D}$ applied to distribution $\mathbf{A}$ does not change the classification results.

(3) The multiplicative constant $1 - \alpha$ does not change the classification too.

Comparison of preferential random walk results.

In label propagation of Eq.(9) or Eq.(10), once the distribution score (a.k.a propagation score) $\mathbf{F}$ are obtained, each unlabeled data point $\mathbf{x}_i$ is assign a class label according to

$$k = \arg\max_{1 \le j \le c} \mathbf{F}_{ij} \tag{5.12}$$

Note the key difference of LGC with GF is the computation of propagation operator $\mathbf{Q}$: LGC uses Eq.(5.9) while GF uses Eq.(5.10), which leads to different label propagation results. Another popular label propagation method is Harmonic function [27], which emphasizes harmonic nature of the label diffusive process. It is very different from LGC and Green's function, thus we did not discuss it here.

We have done extensive experiments to compare the above discussed methods for semi-supervised learning. We defer the presentation of these results in the experiment §7. We next discuss another contribution of this paper, i.e., the maximum consistency algorithm on these preferential random walk/label propagation methods.

5.4 Score Distribution: Confidence of Label Assignment

We begin the presentation of our maximum consistency approach with analysis of the distribution score of the propagation. Our maximum consistency algorithm is not designed in an ad-hoc way, but motivated by the insight obtained from experiments: although label propagation methods are effective, they may not achieve best possible label propagation results. Next we illustrate the reasons.

In both LGC (Eq.9) and GF (Eq.10) methods, the propagation is done in one shot. All unlabeled data obtain their class labels immediately. However, some unlabeled data points may lie nearby in the data manifold (embedding subspace) with the labeled data, while many other unlabeled data lie far-away from the labeled data. Therefore, the reliability or confidence of the class labels obtained in propagation vary from high (for those lie near labeled data) to low (for those lie far-away from labeled data).

However, in the class assignment procedure of Eq.(12), the class decision is simply the largest one among the c classes in the propagation score distribution. For example, for $\mathbf{x}_i$, the score distribution maybe

$$(\mathbf{F}_{i1} \cdots \mathbf{F}_{ic}) = (0.1, 0.2, 0.8, 0.3, 0.05),$$

in a data with $c = 5$. For $\mathbf{x}_j$, the score distribution maybe

$$(\mathbf{F}_{j1} \cdots \mathbf{F}_{jc}) = (0.2, 0.35, 0.38, 0.05, 0.3).$$

Even though both $\mathbf{x}_i, \mathbf{x}_j$ are assigned class label=3, the confidence of the assignments are different. Clearly, $\mathbf{x}_i$ is assigned with higher confidence because $\mathbf{F}_{i3} = 0.8$ is much higher than other classes. $\mathbf{x}_j$ is assigned with lower confidence because $\mathbf{F}_{j3} = 0.38$ is marginally higher than some other classes. In other words, for $\mathbf{x}_i$ the propagation score distribution has a sharp peak while for $\mathbf{x}_j$ the propagation score distribution has a rather flat peak.

There could be many reasons that $\mathbf{x}_i$'s score distribution is much sharper than the score distribution for $\mathbf{x}_j$. $\mathbf{x}_i$ could lie much closer to class$= 3$ labeled data point than $\mathbf{x}_j$. It could also be that there are more class$= 3$ labeled data near $\mathbf{x}_i$ than near $\mathbf{x}_j$. It is also possible that there are many unlabeled points near $\mathbf{x}_i$ such that they mutually enhance the class$= 3$ probability than those near $\mathbf{x}_j$. More possibilities exist. Fortunately, it is not necessary to dig out these details — they are collectively reflected in the propagation score distribution. Consider the existing label propagation approach. Both $\mathbf{x}_i, \mathbf{x}_j$ are assigned labels in the propagation. $\mathbf{x}_j$ obtains class$= 3$ label, although it is done with low confidence.

Now let us consider a different approach where we break the actual label assignment into several rounds. We first assign class label for $\mathbf{x}_i$ and move it to the pool of already-labeled data, while defer the decision for $\mathbf{x}_j$ in later rounds. As the pool of already-labeled data expands to the neighborhood of $\mathbf{x}_j$, the propagation score distribution for $\mathbf{x}_j$ is likely to become sharper. At this time/round, we assign class label to $\mathbf{x}_j$. Thus the class label assignment is always occurring at the situation where the assignment is done with high confidences, i.e., the assignment is done such that the data point is the most consistent with other members of the same class, both globally and locally, as reflected by the sharp score distribution. From these observations and discussions, we design a maximum consistent(MC) label propagation algorithm, which uses the label propagation operator $\mathbf{Q}$ defined in both LGC and GF methods. We call our approach as MC-LGC and MC-GF. Detailed algorithm is presented in next section.

Motivation of Maximum consistency label propagation.

To summarize, semi-supervised learning methods such as random walk did not consider the distributions of scores of multiple labeled but just judged the label of nodes from the maximal score in all the classes of nodes. On the other hand, the unlabeled nodes far away from the labeled nodes can hardly receive the flows. To handle these two issues above, next we present maximum consistency label propagation algorithm.

5.5 Maximum Consistency Label Propagation

Design of the algorithm

Our algorithm design is guided by maximum consistency assumption, which consists of multiple label propagations,

$$\begin{aligned} \mathbf{F}^1 &= \mathbf{Q}\mathbf{Y}^0, \\ \mathbf{F}^2 &= \mathbf{Q}\mathbf{Y}^1, \\ &\cdots \\ \mathbf{F}^t &= \mathbf{Q}\mathbf{Y}^{t-1}, \end{aligned} \tag{5.13}$$

where $\mathbf{Q}$ is the propagation operator which can be computed from Eq.(9) or Eq.(10), and $\mathbf{F}^t$ is the label prediction matrix during each propagation. In each label propagation process, we use the current labeled data matrix $\mathbf{Y}^t$ to update the label prediction matrix $\mathbf{F}^t$.

At the end of each propagation, only those unlabeled data points whose class labels are reliably predicted are actually assigned class labels and moved into the pool of labeled data(Lpool). The rest of unlabeled data points remain in the pool of unlabeled data (Upool). Thus the pool of unlabeled data decreases with each propagation, and the pool of labeled data expands with each propagation. At last propagation, all remaining unlabeled data are assigned class labels.

Because of class balance consideration, the pool of labeled data should get approximately the same number of new members for each class. In our algorithm, each class gets one new member after each propagation. We call this procedure as "balanced class expansion (BCE)". The number of unlabeled data are shrinking while the number of labeled data are increasing during this repeated BCE procedure. The critical issue in this BCE procedure is how to select this new member for each class. i.e., how to decide "reliably predicted" data points in each BCE. As analyzed in above section, the reliability of label propagation is reflected in score distribution. Thus, in our algorithm, we use the score distribution to decide the most "reliable predicted" data points from the data points in Upool in each BCE. We will illustrate more details in the next section.

Discussion.

If we add different number of new members to different classes, it will produce unbalance. Even if the discriminant scores of one class are much higher than that of another class, we also consider add one number for each class. Although it is inefficient, we believe this conservative way will result in selecting more "reliable" data points.

Normalization on the distribution score.

Although data in Lpool expands in a class-balanced way, there are always the situation where classes become unbalanced. In the label propagation, we need to properly normalize the contributions from each class.

Suppose, a subset of data are labeled and there exists a class prior probability π_k. Let $\pi = \mathrm{diag}(\pi_1 \cdots \pi_k)$, and $\mathbf{Z}$ be the multi-class label assignment matrix from labeled data, i.e.,

$$\mathbf{Z}_{ik} = \begin{cases} 1, & \text{if } \mathbf{x}_i \text{ belongs to class k} \\ 0, & \text{otherwise} \end{cases} \tag{5.14}$$

then the balanced source of propagation is defined as

$$\mathbf{Y} = \mathbf{Z}\pi = \begin{pmatrix} \pi_1\mathbf{Z}_{1,1} & \cdots & \pi_c\mathbf{Z}_{1,c} \\ \cdots & \cdots & \cdots \\ \pi_1\mathbf{Z}_{n,1} & \cdots & \pi_c\mathbf{Z}_{n,c} \end{pmatrix}. \tag{5.15}$$

In our algorithm, we set the prior to $\pi_k = \frac{1}{\sum_i \mathbf{Z}_{ik}}$. therefore, each class contributes the same total weight to the propagation: $\sum_i \mathbf{Y}_{ik} = \sum_i \mathbf{Y}_{i\ell}$ for any two class k, ℓ. In our algorithm the initial label matrix $\mathbf{Y}^0$ is constructed as

$$\mathbf{Y}^0 = \mathbf{Z}^0\pi^0, \tag{5.16}$$

where $\mathbf{Z}^0$ is the initial label assignment matrix constructed as Eq.(5.14) from the initially labeled data in Lpool. In the t-th iteration, let $\mathbf{Z}^t$ be the label assignment matrix constructed from current data in Lpool,

$$\mathbf{Y}^t = \mathbf{Z}^t\pi^t. \tag{5.17}$$

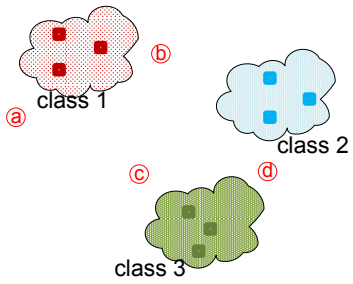


Figure 5.1: Selection of discriminative data in balanced class expansion. Data points: a, b, c, d.

Reliable assigning class labels with score distribution.

After obtaining the assignment score $\mathbf{F}_{ik}$ for all data in Upool, our goal is to pick up the "reliable" assigned data points, one for each class, and add them to the Lpool whereas remove them from the Upool. Afterwards in the actual label assignment for each class, we (1) find out all the currently unlabeled data assigned to this class, (2) pick the one with the highest discriminative score and assign it to this class.

A Motivating example to illustrate discriminant score.

Fig.(5.1) illustrates the idea of selecting discriminative unlabeled data points. Class 1 selects data a instead of data b, because a is far away from classes 2 and 3; although b is slightly closer to class 1, but b is also closer to class 2. In other words, a is more class discriminative than b. Similarly, class 2 selects data c instead of d, because c is more discriminative than d.

Now we discuss the discriminative score computation. For each unlabeled data point $\mathbf{x}_i$, it has been assigned to k scores($\mathbf{F}_{ik}, 1 \leq k \leq c$). The c scores are then sorted as,

$$\mathbf{F}_{ik_1} \geq \mathbf{F}_{ik_2} \geq \mathbf{F}_{ik_3} \geq ... \tag{5.18}$$

3 classes with the highest scores are recorded as the three closest classes for $\mathbf{x}_i$: $\mathbf{F}_{k_1}$; $\mathbf{F}_{k_2}$, $\mathbf{F}_{k_3}$. As discussed above, even two data points $\mathbf{x}_i$ and $\mathbf{x}_j$ have been assigned to the same class c_k, they may have different discriminant scores depending on the scores which how $\mathbf{x}_i, \mathbf{x}_j$ may be assigned to other classes. Here we consider the target class the data points

will be assigned to and other two competing classes which we wish to be discriminant against. The discriminative scores for the 1st choice target class are defined as (if there is only 2 classes, we do not need c_{k3}),

$$\mathbf{D}(i, c_{k1}) = \mathbf{F}^2_{ic_{k1}} \frac{|\mathbf{F}_{ic_{k1}} - \mathbf{F}_{ic_{k2}}| + |\mathbf{F}_{ic_{k1}} - \mathbf{F}_{ic_{k3}}|}{\sqrt{\mathbf{F}_{ic_{k1}} + \mathbf{F}_{ic_{k2}} + \mathbf{F}_{ic_{k3}}}}. \tag{5.19}$$

The score difference achieves the discriminative affects. The denominator provides a mild scale normalization. Without this term, the class with largest $\mathbf{F}_{ik}$ scale may dominate the score computation process. Note that these scores are computed once for each balanced class expansion. For each unlabeled data point $\mathbf{x}_i$ in Upool, it is assigned to class k, which has the largest $\mathbf{D}(x_i, c_k)$ scores among all class k. For each class k, we select the data points $\mathbf{x}_i$, which has the largest discriminative score $\mathbf{D}(x_i, c_k)$ among all data points in Upool assigned to class k. This procedure is designed to maximize the label assignment consistency, which is consistent with LGC/GF approach.

Discussion on the discriminant score.

Actually, we can define other formulations of discriminant score. (1) Without the denominator of Eq.(5.19), discriminant score can be written as,

$$\mathbf{D}_2(i, c_{k1}) = \mathbf{F}^2_{ic_{k1}}(|\mathbf{F}_{ic_{k1}} - \mathbf{F}_{ic_{k2}}| + |\mathbf{F}_{ic_{k1}} - \mathbf{F}_{ic_{k3}}|). \tag{5.20}$$

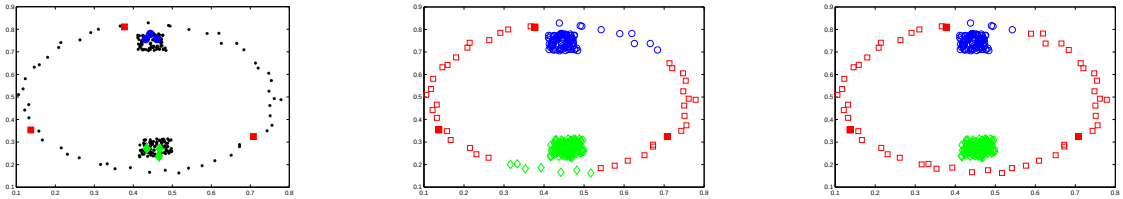
(2) Without the square for the 1st term of Eq.(5.19), discriminant score can be written as,

$$\mathbf{D}_3(i, c_{k1}) = \mathbf{F}_{ic_{k1}} \frac{|\mathbf{F}_{ic_{k1}} - \mathbf{F}_{ic_{k2}}| + |\mathbf{F}_{ic_{k1}} - \mathbf{F}_{ic_{k3}}|}{\sqrt{\mathbf{F}_{ic_{k1}} + \mathbf{F}_{ic_{k2}} + \mathbf{F}_{ic_{k3}}}}. \tag{5.21}$$

(3) Select more top (e.g., $4, 5, 6, 7, \cdot\cdot$) classes to compute the discriminant score, then discriminant score for T classes is given by,

$$\mathbf{D}_4(i, c_{k1}) = \mathbf{F}^2_{ic_{k1}} \frac{\sum_{t=1}^{T} |\mathbf{F}_{ic_{k1}} - \mathbf{F}_{ic_{kt}}|}{\sqrt{\sum_{t=1}^{T} \mathbf{F}_{ic_{kt}}}}. \tag{5.22}$$

Our experiments results(see §7.4) show Eq.(5.19) achieves slightly better results than other discriminant scores defined in Eqs.(5.20,5.21,5.22). For Eq.(5.20), the denominator is removed. When some $\mathbf{F}_{ic_k}$ has very large values, it may dominator the score. For Eq.(5.21),



(a) Data distribution. (b) LGC result (c) MC-LGC result

Figure 5.2: Illustration of maximum consistency approach on a synthetic dataset. Labeled data shown in thick symbols: red squares, green diamonds, blue circles for 3 classes. Initially unlabeled data are shown in black stars and, after obtaining labels, shown in open symbols.

square of score $\mathbf{F}_{ic_k}$ is removed, which makes the score less sharper than that of Eq.(5.19). For Eq.(5.22), more top classes are fetched to achieve discriminant effect. In our experiments, we find when we select 3 classes, we can get very good results. When we select more classes, the results change slightly, but sometimes even worse.

Demonstration of algorithm performance on toy data.

Here we use a toy data example to illustrate the advantage of the MC approach (on LGC methods) in Fig.2. A 3-class synthetic dataset is displayed in Fig.(2a). For each class, three data points are labeled while the rest of data points are unlabeled. Results of standard LGC methods and MC-LGC methods are shown in Figs.(2b, 2c). It is clear that MC approaches achieves better results. One can get similar results if making the comparisons of GF against MC-GF methods.

Complete algorithm.

Above we have given a detailed discussion on the rational of the algorithm. The whole algorithm is listed in Algorithm 1. This algorithm wraps around the label propagation operator $\mathbf{Q}$, and it can also use other label propagation operators.

Time complexity analysis.

Note we only need to compute propagation operator $\mathbf{Q}$ (through Eq.5.10 or Eq.5.9) once as in standard LGC or GF, and the extra time cost is the iteration cost in balanced

Algorithm 6 Maximum consistency label propagation algorithm (MC algorithm)

Input: labeled data $L = \{(\mathbf{x}_i, y_i)\}_{i=1}^{\ell}$, unlabeled data $U = \{\mathbf{x}_j\}_{j=\ell+1}^{\ell+u}$, MaxIter

Output: predicted class labels for unlabeled data

Procedure:

1: compute propagation operator $\mathbf{Q}$ with Eq.(9) or Eq.(10), compute initial label matrix $\mathbf{Y}^0$ using Eq.(16), $t = 1$
2: **while** $t <$ MaxIter & U is not empty **do**
3: $\mathbf{F}^t = \mathbf{Q}\mathbf{Y}^{t-1}$
4: **for** each unlabeled data **do**
5: compute its corresponding discriminative score using Eq.(5.19)
6: **end for**
7: **for** $k = 1$ to c **do**
8: search all unlabeled data whose 1st choice target class is k. {Balanced class expansion}
9: **if** not empty **then**
10: pick the one with the largest discriminative score, add it to class k, remove it from U
11: **end if**
12: **end for**
13: Update $\mathbf{Y}^t$ with Eq.(17) using current label assignment $\mathbf{Z}^t$ {new labeled data added to Lpool}
14: $t = t + 1$
15: **end while**

class expansion(BCE) process, which includes (1)the iteration time of BCE process which is proportional to number of iteration t; (2) the discriminant score table computation in lines $7 - 13$ of Algorithm 1, which is proportional to the number of *current* unlabeled data points n_l and the number of class label c. In our experiment, we find that the extra time cost is very limited as compared to the propagation operator computation in step 1.

5.6 Connection to Related Works

In this section, to make our contribution more clear, we discuss the related works highly related to our algorithm. The related methods can be roughly divided into three categories, (1) personalized random walk (RW); (2) semi-supervised learning (SSL); (3) belief propagation (BP).

Random Walk is a very popular technique widely used for PageRank algorithm [140]. Many variations of random walk methods are proposed, including personalized page rank [141], lazy random walks [146], fast random walk with restart [147], center-piece subgraph discovery [148], using ghost edge for classification in sparsely labeled networks [149] and so on.

Semi-Supervised Learning methods are widely used in real applications. Graph-based semi-supervised methods are the most popular and effective methods in semi-supervised learning. The key-idea of graph-based semi-supervised methods is to estimate a (label propagation) function on a graph, which maximizes (1) consistency with the label information; (2) the smoothness over the whole graph. Several representative methods include harmonic function [27], local and global consistency [28] and Green's function [29].

Belief Propagation [150] is widely used for inference in probability graphical model. Belief propagation methods can be used for collective classification for network data [151], grouping nodes into regions for graphs [152] and so on. However, the computational cost for BP method is usually very high.

In this paper, we mainly focus on (1) discuss the relations between semi-supervised learning and random walks; (2) propose a maximum consistency label propagation methods. We note there is few (if not many) papers to discuss the relations between different kinds of random walks and semi-supervised learning. In our paper, we try to answer the following questions: (1) whether these random walks methods are related or not; (2) whether they are identical or not; (3) which method can produce the best results; (4) is there one method consistently performing better than the others? To our knowledge, the relations have not been well investigated so far. One similar work to our paper is about unifying guilt-by-association approach [153], which discusses the relations between random walk, semi-supervised learning and belief propagation. For maximum consistency random walk method, it is an improvement of state-of-the-art semi-supervised learning methods, which

Table 5.1: Descriptions of datasets

Dataset	#Size	#Dimension	#Class
AT&T	400	644	40
Caltech	600	432	20
MSRC	210	432	7
Binalpha	1014	320	36
Mnist	150	784	10
Umist	360	644	20
Newsgroup	499	500	5
Reuters	900	1000	10
digit	1500	241	2

extends the works of local and global consistency [28] and Green's function [29]. Our method can be extended to be used for collective classification [151] and community detection [154]. Due to space limit, we omit the discussions here.

5.7 Experiments

In this section, we perform two groups of experiments. One group is to compare three different methods in preferential random walks of Eqs.(6-8), and the other group is to evaluate the effectiveness of maximum consistency (MC) algorithm. First we discuss the datasets used in our experiments.

Datasets

We adopt 9 data sets in our experiments, including two face datasets AT&T and umist, three digit datasets mnist([73]), binalpha and digit[1], two image scene datasets Caltec101 [74, 75] and MSRC [75], and two text datasets Newsgroup[2] and Reuters[3]. Table 5.1 summarizes the characteristics of the datasets.

Experiments results on 3 methods of Generalized Preferential Random Walks of Eqs.(6-8)

[1]http://www.kyb.tuebingen.mpg.de/ssl-book/ benchmarks.html

[2]http://people.csail.mit.edu/jrennie/20Newsgroups/

[3]http://www.daviddlewis.com/resources/testcollections/reuters21578/

In §2, we give three methods for generalized preferential random walks. We also show method 2 is equivalent to LGC method. When $\alpha = 0.1$, method 1 is equivalent to GF method. In all the methods except in GF, parameter α will influence the semi-supervised classification results. For image datasets, we use Gaussian kernel to construct the graph edge weights $\mathbf{W}_{ij} = e^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{\sigma^2}}$, where γ is fine tuned according to [27]. For text datasets, we use linear kernel to compute the graph similarity. We randomly select 20% of all data as the training data. In Fig.3, we show the average classification results on 4 methods (GF, method1, method2(=LGC), method3) by using 5-fold cross-validation. In Fig.3, x-axis represents the different α settings($\alpha = 0.1, 0.3, 0.5, 0.7, 0.9$), y-axis is the average classification accuracy over 10 independent runs.

Experiment result analysis.

From Fig. 3, we can observe: (1) method 1 and GF perform well on all the datasets; (2) parameter α does not influence very much for the classification results obtained from method 1; (3) method 2 and 3 perform reasonably well when $\alpha \leq 0.5$, but their performances degrade much when α is approaching 1.

Experiment results on maximum consistency algorithm.

We compare maximum consistency algorithm with standard LGC and GF methods. The α in LGC and MC-LGC methods are set to $\alpha = 0.5$ as suggested in [32]. We use Eq.(5.19) as the discriminant score in the balanced class expansion process. The maximum iteration time T is set according to the number of data points in the unlabeled pool. If there are more than $\theta = 90\%$ of the whole data labeled, we stop our maximum consistency algorithm, and do one-shot label propagation.

We show the classification results of 4 methods (LGC, MC-LGC, GF, MC-GF) by randomly selecting different percentage of labeled data in Fig.4, where x-axis represents different percentages of labeled data (i.e., $10\%, 20\%, \cdots$), and y-axis is the average classification accuracy over 10 independent runs.

Experiment results analysis.

From Fig. 4, we can observe, (1) MC-LGC consistently performs better than LGC especially when the percentage of labeled data is very small(e.g., 10%); (2) MC-GF performs much better than GF; (3) on text dataset, MC-GF's superiority is more significant(more than 5% improvement).

Next, we discuss our maximum consistency algorithm experiment results when using different parameter settings.

Discussion on discriminant score computation.

Discriminant score computation is very important for the decision of data to be propagated. The first issue is how to compute the discriminant score. Here we show the experiment results of classification when alternative discriminant score computation formulations of Eq.(5.20), Eq.(5.21) are used. The other settings of the experiments are the same as those described in §7.3. Fig. 5 shows the classification results of 4 methods of label propagation(GF, MC-GF, LGC, MC-LGC) by using different discriminant score computations of Eqs.(5.19,5.20,5.21) on datasets MSRC and binalpha. We can observe, most of the time, the classification results obtained from Eq.(5.19) are slightly better on both datasets for both MC-GF and MC-LGC methods. These experiment results suggest us to use Eq.(5.19) for discriminant score in our algorithm.

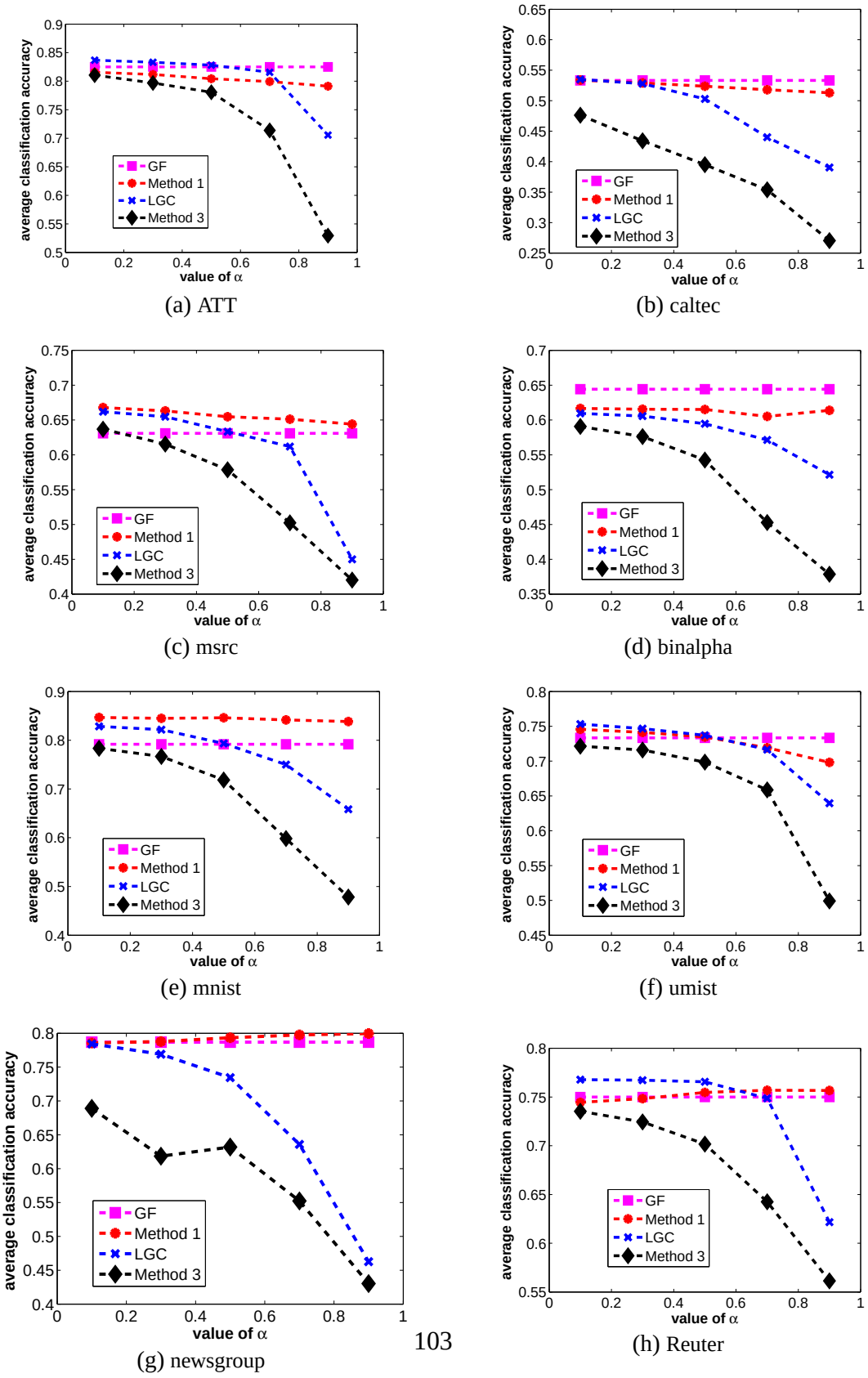
Discussion on the iteration number.

Another key parameter is related to what extent is the procedure designed to maximize the label assignment consistency. As described in §7.3, we use the number of labeled data points in labeled pool as a criteria to stop our algorithm. We use parameter θ to represent the percentage of *currently* labeled data of the whole dataset. In §7.3, we set $\theta = 0.9$. We try different settings of $\theta = \{60\%, 70\%, 80\%, 90\%, 100\%\}$ and report the experiment results on dataset Caltec in Fig. 6. The other settings of the experiments are the same as those described in §7.3. We find, on most of the datasets, if we set $\theta = 90\%$, we can achieve

the best results. Thus we set $\theta = 90\%$ as the default setting for our maximum consistency algorithm.

5.8 Lessons learned

We analyze the relations between 3 methods of generalized preferential random walks and label propagation methods. A maximum consistency algorithm is presented to improve the current label propagation methods. Extensive experiments on 9 datasets show the effectiveness of MC algorithm and different generalized preferential random walks. We will explore the opportunities of algorithm improvement on other semi-supervised learning models, e.g., support vector machine, k-nearest neighbors, etc. Also, we will apply the proposed random walk model for medical image segmentation tasks.



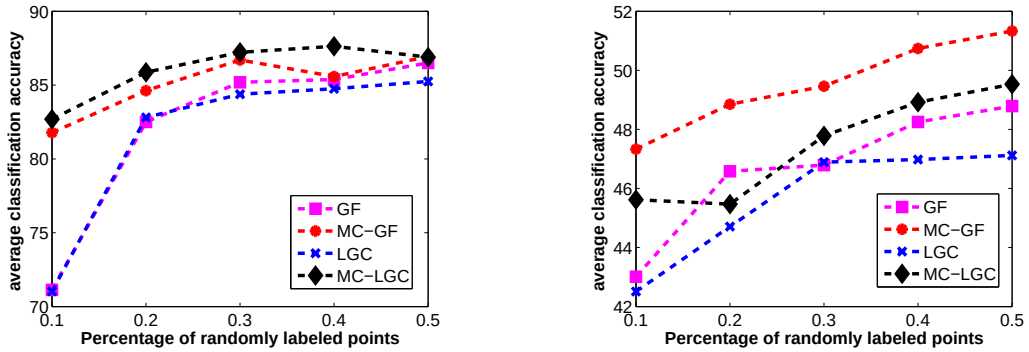
(a) ATT (b) caltec

(c) msrc (d) binalpha

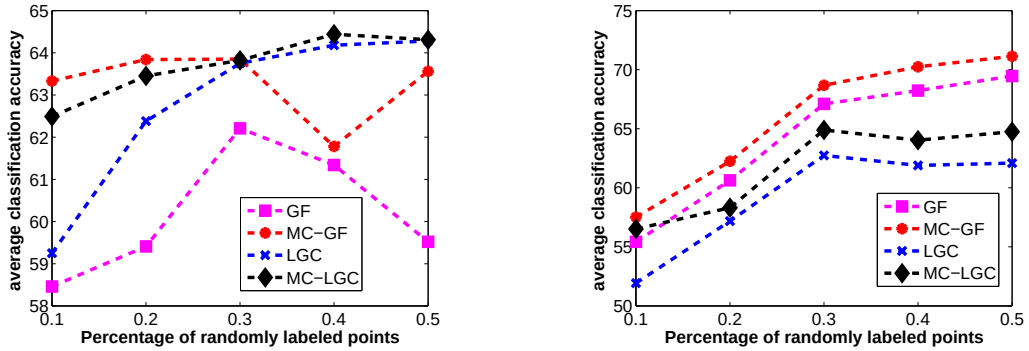
(e) mnist (f) umist

(g) newsgroup (h) Reuter

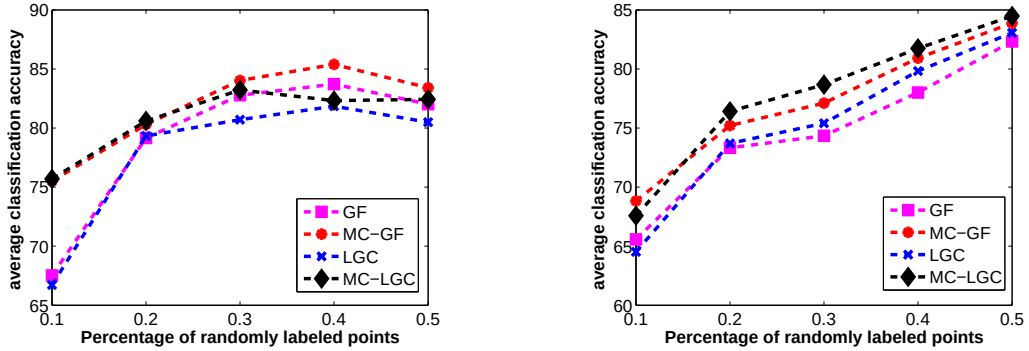
Figure 5.3: Experiments results on 4 methods of Generalized Preferential Random Walks: GF, method1, method2(=LGC), method3. x-axis represents the different α settings($\alpha = 0.1, 0.3, 0.5, 0.7, 0.9$), y-axis is the average classification accuracy over 10 independent runs.



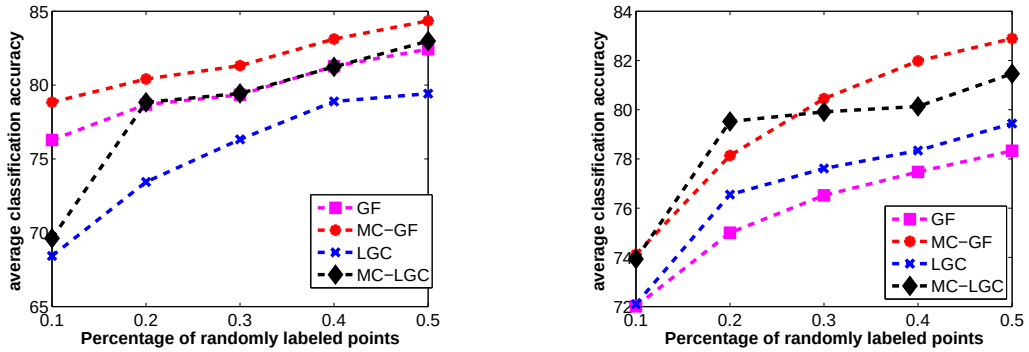
(a) ATT (b) caltec



(c) msrc (d) binalpha

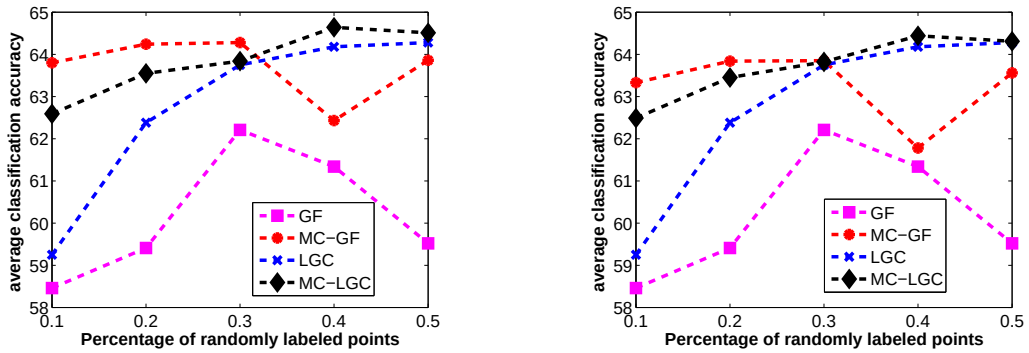


(e) mnist (f) umist



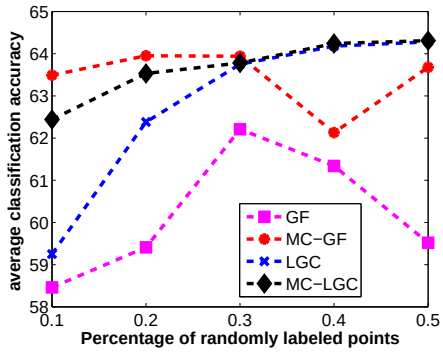
(g) newsgroup (h) Reuter

Figure 5.4: Experiments results on 4 methods of label propagation: GF, MC-GF, LGC, MC-LGC. x-axis represents the different percentage of labeled data, y-axis is the average classification accuracy over 10 independent runs

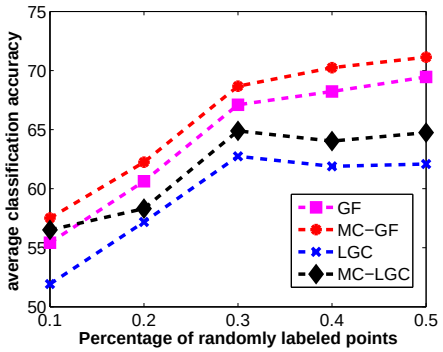


(a) MSRC with Eq.(5.19)

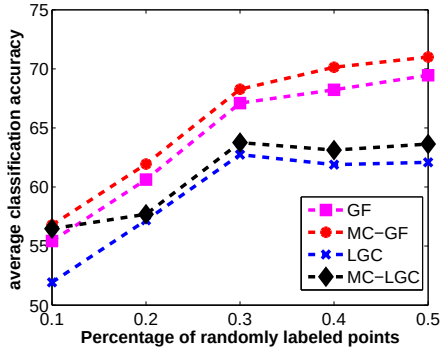
(b) MSRC with Eq.(5.20)



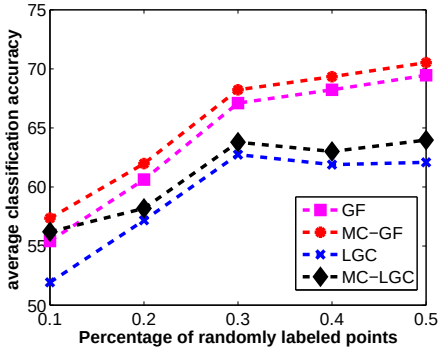
(c) MSRC with Eq.(5.21)



(d) binalpha with Eq.(5.19)

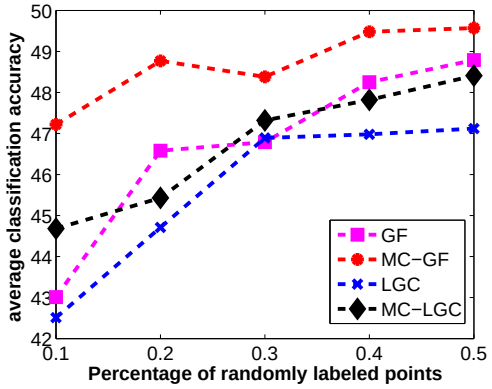


(e) binalpha with Eq.(5.20)

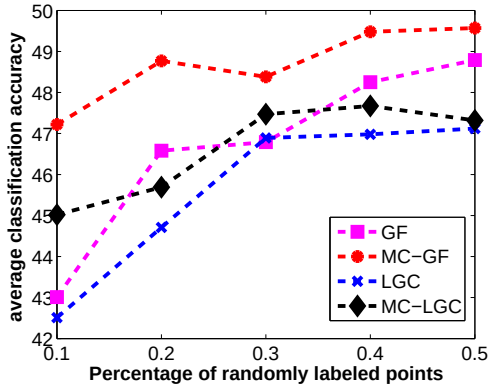


(f) binalpha with Eq.(5.21)

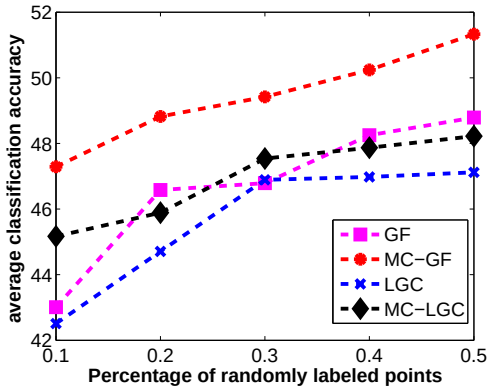
Figure 5.5: Experiments results on 4 methods of label propagation: GF, MC-GF, LGC, MC-LGC using different discriminant score computations of Eqs.(5.19,5.20 and 5.21) on datasets MSRC and binalpha. x-axis represents the different percentage of labeled data, y-axis is the average classification accuracy over 10 independent runs



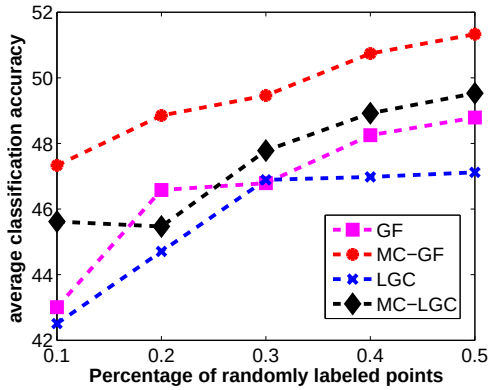
(a) Caltec ($\theta = 60\%$)



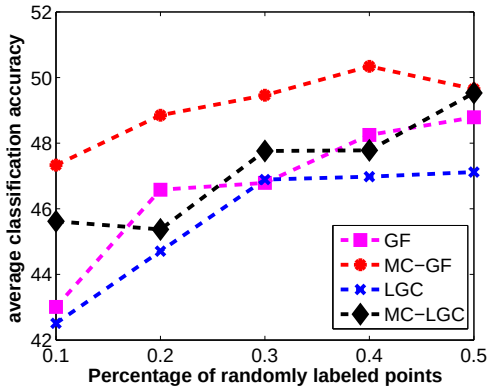
(b) Caltec ($\theta = 70\%$)



(c) Caltec ($\theta = 80\%$)



(d) Caltec ($\theta = 90\%$)



(e) Caltec ($\theta = 100\%$)

Figure 5.6: Experiments results on 4 methods of label propagation: GF, MC-GF, LGC, MC-LGC by using different parameter θ on dataset Caltec. x-axis represents the different percentage of labeled data, y-axis is the average classification accuracy over 10 independent runs

CHAPTER 6

Conclusion

In this thesis, we provide different methods for dimension reduction, feature engineering and label propagation. We apply the proposed methods image classification/annotation tasks. Throughout the paper, we have the following contributions.

- This thesis proposes an efficient iterative locally linear embedding algorithm.
- This thesis presents two low-rank data recovery models through replacing the rank constraint by a Schatten p norm, for data recovery purpose.
- This thesis proposes an iteratively re-weighted method to solve the generic group lasso problem, where an arbitrary structure can be enforced on feature space.
- This thesis proposes a maximum consistency algorithm for preferential random walks and label propagation.
- Extensive experiments results indicate the good performance of proposed algorithms.

6.1 Future work

According to our current research findings, there are many possible extensions of current approaches, in terms of different topics.

- In terms of Locally Linear embedding (LLE), besides its capability for dimension reduction, we are curious about the de-noising power of this method. In real world, there are noisy data with missing values. How to get the correct embedding results for these noisy data remains a challenging issue. We will explore robust methods for noisy data embedding. Meanwhile, LLE has close relations with subspace learning

and multi-subspace learning. This is also closely connected with motion segmentation. Can we explore methods for better understanding the subspace?

- Schatten p norm model can be used for data recovery purpose. A direct application of this model is for recommendation systems. Can it be exploited for recommendation system analysis? How to incorporate different prior knowledge into this model? Can Schatten p norm model extended for tensor analysis (e.g., video analysis)?
- In terms of constraint feature selection using group lasso regularization term, can we exploit method using exclusive lasso for feature selection? This is helpful because we can enforce exclusiveness constraint according to the exclusiveness of features. Further, can we explore both group lasso and exclusive lasso for feature learning purpose, especially for multi-view feature learning?
- There are many applications of random walk in different domain. We will consider more practical problems, which involves medical image segmentation, social media link predication. For multi-stage semi-supervised learning algorithms, the similar idea can be applied to many models. We will explore the opportunities of algorithm improvement on other semi-supervised learning models, e.g., support vector machine, k-nearest neighbors, etc?
- Extension to large scale data. Nowadays, a large amount of data is available for data analysis. How to speed up our algorithm for large-scale computational purpose becomes a more and more important question then ever before. Distributed computing framework provides us opportunities to explore the system-level optimization using Mapreduce (Hadoop) framework. Paralleling our algorithms to distributed framework and adapting them for large-scale computing framework will be a solution. However, most existing methods cannot directly fit into the current system paradigm, and necessary modifications or extensions of algorithms are needed for large-scale computing purpose.

6.2 Summary

In this thesis, we provide a comprehensive studying for some key technologies in feature engineering, dimension reduction and label propagation. We provide improvements of current state-of-the-art algorithms. Extensive experiments are performed to validate the effectiveness of proposed approach. Our findings and discoveries are expected to be helpful for many practical applications, e.g., image categorization/annotation, text classification, graph link analysis, etc.

REFERENCES

[1] A. Raftery, "Bayesian model selection in social research," 1995.

[2] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, vol. 41, no. 6, pp. 391–407, 1990.

[3] D. M. Blei, A. Ng, M. Jordan, and J. Lafferty, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, p. 2003, 2003.

[4] K. Kreutz-Delgado, J. Murray, B. Rao, K. Engan, T.-W. Lee, and T. J. Sejnowski, "Dictionary learning algorithms for sparse representation," *Neural Computation*, vol. 15, no. 2, pp. 349–396, 2003.

[5] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *NIPS*, 2006, pp. 801–808.

[6] Y. Bengio, "Learning deep architectures for ai," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[7] B. Yao, A. Khosla, and F.-F. Li, "Combining randomization and discrimination for fine-grained image categorization," in *CVPR*, 2011, pp. 1577–1584.

[8] I. Guyon, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[9] R. Jenatton, J. Audibert, and F. Bach, "Structured variable selection with sparsity-inducing norms," *Technical report*, vol. arXiv:0904.3523, 2009.

[10] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski, "Structured sparsity through convex optimization," *CoRR*, vol. abs/1109.2397, 2011.

[11] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society, Series B*, vol. 58, pp. 267–288, 1994.

[12] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, pp. 1289–1306, 2006.

[13] R. Memisevic, "On multi-view feature learning," *CoRR*, vol. abs/1206.4609, 2012.

[14] R. Caruana, "Multitask learning," in *Machine Learning*, 1997, pp. 41–75.

[15] A. M. Martinez and A. C. Kak, "Pca versus lda," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 228–233, 2001.

[16] T. Henri and C. Fan, "Relations between two sets of variates: The bits of information provided by each variate in each set," *Statistics & Probability Letters*, vol. 6, no. 3, pp. 137–139, Feb. 1988.

[17] B. Scholkopf, A. Smola, E. Smola, and K. Muller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, pp. 1299–1319, 1998.

[18] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *SCIENCE*, vol. 290, pp. 2323–2326, 2000.

[19] L. de, "Modern multidimensional scaling: Theory and applications (second edition)," *Journal of Statistical Software*, vol. 14, no. b04, 2005.

[20] J. Tenenbaum, V. Silva, and J. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[21] B. Kulis, "Fast low-rank semidefinite programming for embedding and clustering," in *AISTATS 2007*, 2007, p. 2007.

[22] T. Kolda and B. Bader, "Tensor decompositions and applications," *SIAM REVIEW*, vol. 51, no. 3, pp. 455–500, 2009.

[23] Z. Xu, F. Yan, and Y. Qi, "Inftucker: t-process based infinite tensor decomposition," *CoRR*, vol. abs/1108.6296, 2011.

[24] M. Datar and P. Indyk, "Locality-sensitive hashing scheme based on p-stable distributions," in *In SCG04: Proceedings of the twentieth annual symposium on Computational geometry*. ACM Press, 2004, pp. 253–262.

[25] X. Zhu and A. Goldberg, *Introduction to Semi-Supervised Learning*, ser. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2009.

[26] Y. Bengio, O. Delalleau, and N. Le Roux, "Label Propagation and Quadratic Criterion," pp. 193–216, 2006.

[27] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Proceedings of the 20th International Conference on Machine Learning*, 2003, pp. 912–919.

[28] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Scholkopf, "Learning with local and global consistency," *Advances in Neural Information Processing Systems*, vol. 16, pp. 321–328, 2004.

[29] C. Ding, R. Jin, T. Li, and H. D. Simon, "A learning framework using Green's function and kernel regularization with application to recommender system," in *KDD*, 2007, pp. 260–269.

[30] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *IN ICML*, 2003, pp. 912–919.

[31] X. Zhu, J. Lafferty, and Z. Ghahramani, "Combining active learning and semi-supervised learning using gaussian fields and harmonic functions," in *ICML 2003 workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, 2003, pp. 58–65.

[32] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Scholkopf, "Learning with local and global consistency," in *Advances in Neural Information Processing Systems 16*. MIT Press, 2004, pp. 321–328.

[33] L. Lovsz, "Random walks on graphs: A survey," 1993.

[34] C. Q. Ding, R.Jin, T.Li, and H. Simon, "A learning framework using green s function and kernel regularization with application to recommender system," in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California, USA, August 12-15, 2007*, 2007, pp. 260–269.

[35] S. C. H. Hoi, R. Jin, and M. Lyu, "Learning nonparametric kernel matrices from pairwise constraints," in *ICML '07: Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 361–368.

[36] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Advances in Neural Information Processing Systems 14*. MIT Press, 2001, pp. 585–591.

[37] H. Sahbi, P. Etyngier, J.-Y. Audibert, and R. Keriven, "Manifold learning using robust graph laplacian for interactive image search," *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 0, pp. 1–8, 2008.

[38] L. Meier, S. Geer, and P. Bhlmann, "The group lasso for logistic regression," *Journal of the Royal Statistical Society, Series B*, 2008.

[39] Y. Zhou, R. Jin, and S. C. H. Hoi, "Exclusive lasso for multi-task feature selection." *Journal of Machine Learning Research - Proceedings Track*, vol. 9, pp. 988–995, 2010.

[40] V.Roth, "The generalized lasso: a wrapper approach to gene selection for microarray data," Proceedings 14th International Conference on Automated Deduction (CADE-14), 252–255, Tech. Rep., 2002.

[41] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *Journal of the Royal Statistical Society Series B*, pp. 91–108, 2005.

[42] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 171–184, 2013.

[43] J. Herlocker, J. Konstan, L. Terveen, and T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems*, vol. 22, pp. 5–53, 2004.

[44] H. Cheng, Z. Liu, and J. Yang, "Sparsity induced similarity measure for label propagation," in *ICCV*, 2009, pp. 317–324.

[45] N. S. Aybat and G. Iyengar, "A first-order augmented lagrangian method for compressed sensing," 2010, sIAM J. Optim.

[46] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web." Stanford InfoLab, Technical Report 1999-66, November 1999, previous number = SIDL-WP-1999-0120.

[47] L. Backstrom and J. Leskovec, "Supervised random walks: Predicting and recommending links in social networks," *CoRR*, vol. abs/1011.4071, 2010.

[48] D.Kong and C. Ding, "An iterative locally linear embedding algorithm," in *ICML*, 2012.

[49] C. Ding, D. Kong, and M. Zhang, "Collective kernel construction in noisy environment," in *SDM*, 2013, pp. 64–72.

[50] D. Kong, M. Zhang, and C. Ding, "Minimal shrinkage for noisy data recovery using schatten-p norm objective," in *ECML/PKDD*, 2013, pp. 177–193.

[51] D. Kong and C. Ding, "Efficient algorithms for selecting features with arbitrary group constraints via group lasso," in *ICDM*, 2013.

[52] D. Kong and C. H. Q. Ding, "Maximum consistency preferential random walks," in *ECML/PKDD*, 2012, pp. 339–354.

[53] D. Kong, C. Ding, H. Huang, and H. Zhao, "Multi-label relieff and f-statistic feature selections for image annotation," in *CVPR*, 2012, pp. 2352–2359.

[54] C. Ding and D. Kong, "Nonnegative matrix factorization using a robust error function," in *ICASSP*, 2012, pp. 2033–2036.

[55] D. Kong and C. Ding, "A semi-definite positive linear discriminant analysis and its applications," in *ICDM*, 2012, pp. 942–947.

[56] D. Kong, C. Ding, and H. Huang, "Robust nonnegative matrix factorization using l21-norm," in *CIKM*, 2011, pp. 673–682.

[57] G. Yan, N. Brown, and D. Kong, "Exploring discriminatory features for automated malware classification," in *DIMVA*, 2013, pp. 41–61.

[58] D. Kong and G. Yan, "Discriminant malware distance learning on structural information for automated malware classification," in *KDD*, 2013, pp. 1357–1365.

[59] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *science*, vol. 290, pp. 2219–2323, 2000.

[60] J. Ham, D. Lee, S. Mika, and B. Scholkopf, "A kernel view of the dimensionality reduction of manifolds," 2004.

[61] D. L. Donoho and C. Grimes, "Hessian eigenmaps: New locally linear embedding techniques for high-dimensional data," *Proceedings of National Academy of Science*, pp. 5591–5596, 2003.

[62] Z. Zhang and Z. Zha, "Principal manifolds and nonlinear dimensionality reduction via tangent space alignment," *SIAM J. Scientific Computing*, vol. 26, pp. 313–338, 2004.

[63] K. M. Hall, "R-dimensional quadratic placement algorithm," *Management Science*, vol. 17, pp. 219–229, 1971.

[64] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," *NIPS*, 2001.

[65] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *NIPS*. MIT Press, 2001, pp. 849–856.

[66] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 888–905, 1997.

[67] P. Chan, M.Schlag, and J. Zien, "Spectral k-way ratio-cut partitioning and clustering," *IEEE Trans. CAD-Integrated Circuits and Systems*, vol. 13, pp. 1088–1096, 1994.

[68] F. Wang and C. Zhang, "Label propagation through linear neighborhoods," in *ICML*, 2006.

[69] H. Zha, C. Ding, M. Gu, X. He, and H. Simon, "Spectral relaxation for K-means clustering," *NIPS*, pp. 1057–1064, 2001.

[70] C. Ding and X. He, "K-means clustering and principal component analysis," *Int'l Conf. Machine Learning (ICML)*, 2004.

[71] C. Ding, T. Li, and M. Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2010.

[72] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 643–660, 2001.

[73] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, 1998, pp. 2278–2324.

[74] D. Dueck and B. J. Frey, "Non-metric affinity propagation for unsupervised image categorization," in *ICCV*, 2007, pp. 1–8.

[75] Y. Lee and K. Grauman, "Foreground focus: Unsupervised learning from partially matching images," *International Journal of Computer Vision*, vol. 85, no. 2, pp. 143–166, 2009.

[76] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," *Proc. Int'l Conf. Machine Learning*, 2003.

[77] C. Ding, R. Jin, T. Li, and H. Simon, "A learning framework using green's function and kernel regularization with application to recommender system," in *KDD*, 2007, pp. 260–269.

[78] J. Cai, E. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[79] E. Candès and B. Recht, "Exact matrix completion via convex optimization," *Communications of the ACM*, vol. 55, no. 6, pp. 111–119, 2012.

[80] B. Recht, M. Fazel, and P. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM Review*, vol. 52, no. 3, pp. 471–501, 2010.

[81] J. M. Rennie and N. Srebro, "Fast maximum margin matrix factorization for collaborative prediction," in *ICML*, 2005, pp. 713–719.

[82] J. Abernethy, F. Bach, T. Evgeniou, M.Paristech, and T. Jaakkola, "A new approach to collaborative filtering: Operator estimation with spectral regularization," *Journal of Machine Learning Research*, vol. 10, pp. 803–826, 2009.

[83] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," in *ICCV*, 2009, pp. 2114–2121.

[84] I. Jolliffe, *Principal Component Analysis*. Springer, 1986.

[85] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[86] C. H. Q. Ding, D. Zhou, X. He, and H. Zha, "R_1-pca: rotational invariant l_1-norm principal component analysis for robust subspace factorization," in *ICML*, 2006, pp. 281–288.

[87] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization," in *NIPS*, 2009.

[88] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images," in *CVPR*, 2010, pp. 763–770.

[89] M. Fazel, "Matrix rank minimization with applications," *PhD thesis, Stanford University*, pp. 1–130, 2002.

[90] E. Candès and T. Tao, "The power of convex relaxation: near-optimal matrix completion," *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2053–2080, 2010.

[91] V. Chandrasekaran, S. Sanghavi, P. Parrilo, and A. Willsky, "Sparse and low-rank matrix decompositions," in *Allerton'09 Proceedings of the 47th annual Allerton conference on Communication, control, and computing*, 2009, pp. 962–967.

[92] S. Ma, D. Goldfarb, and L. Chen, "Fixed point and bregman iterative methods for matrix rank minimization," *Mathematical Programming: Series A and B*, vol. 128, no. 1-2, pp. 321–353, 2011.

[93] K. Toh and S. Yun, "An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems," *Pacific Journal of Optimization*, 2010.

[94] T. Pong, P. Tseng, S. Ji, and J. Ye, "Trace norm regularization: Reformulations, algorithms, and multi-task learning," *SIAM Journal on Optimization*, vol. 20, no. 6, pp. 3465–3489, 2010.

[95] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," *Numerische Mathematik*, vol. 14, pp. 403 – 420, 1970.

[96] Z. Lin, M. Chen, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," in *UIUC Tech. Rep., UILU-ENG-09-2214*, 2010.

[97] D. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*. Athena Scientific, 1996.

[98] F. Nie, H. Huang, and C. Ding, "Low-rank matrix recovery via efficient schatten p-norm minimization," in *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[99] A. Argyriou, C. Micchelli, M. Pontil, and Y. Ying, "A spectral regularization framework for multi-task structure learning," in *NIPS*, 2007.

[100] P. Jain, R. Meka, and I. Dhillon, "Guaranteed rank minimization via singular value projection," in *NIPS*, 2010.

[101] K. Mohan and M. Fazel, "Iterative reweighted least squares for matrix rank minimization," in *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*. IEEE, 2010, pp. 653–661.

[102] R. Keshavan, A. Montanari, and S. Oh, "Matrix completion from noisy entries," *The Journal of Machine Learning Research*, vol. 11, pp. 2057–2078, 2010.

[103] D. Wipf, "Non-convex rank minimization via an empirical bayesian approach," in *UAI*, 2012, pp. 914–923.

[104] S. Kim and E. Xing, "Tree-guided group lasso for multi-task regression with structured sparsity," in *ICML*, 2010.

[105] S. Kim, K. Sohn, and E. Xing, "A multivariate regression approach to association analysis of a quantitative trait network," *Bioinformatics*, vol. 25(12):204-212, 2009.

[106] R. Tibshirani, M. Saunders, and S. Rosset, "Sparsity and smoothness via the fused lasso," *Journal of the Royal Statistical Society, Series B*, vol. 67, no. 1, pp. 91–108, 2005.

[107] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society, Series B*, vol. 68, pp. 49–67, 2006.

[108] C. Chen, Y. Li, and J. Huang, "Learning with forest sparsity," *CoRR*, vol. abs/1211.4657, 2012.

[109] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[110] J. Liu, S. Ji, and J. Ye, "Multi-task feature learning via efficient l2,1-norm minimization," in *UAI*, 2009, pp. 339–348.

[111] Y. Nesterov, "Gradient methods for minimizing composite objective function," *ECORE Discussion Paper*, 2007.

[112] J. Huang, T. Zhang, and D. N. Metaxas, "Learning with structured sparsity," *Journal of Machine Learning Research*, vol. 12, pp. 3371–3412, 2011.

[113] J. Liu and J. Ye, "Moreau-yosida regularization for grouped tree structure learning," in *NIPS*, 2010, pp. 1459–1467.

[114] L. Jacob, G. Obozinski, and J.-P. Vert, "Group lasso with overlap and graph lasso," in *ICML*, 2009, p. 55.

[115] X. Chen, Q. Lin, S. Kim, J. Carbonell, and E. Xing, "Smoothing proximal gradient method for general structured sparse learning," in *UAI*, 2011, pp. 105–114.

[116] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[117] J. Mairal, R. Jenatton, G. Obozinski, and F. Bach, "Network flow algorithms for structured sparsity," in *NIPS*, 2010.

[118] A. Argyriou, C. Micchelli, M. Pontil, L. Shen, and Y. Xu, "Efficient first order methods for linear composite regularizers," *Computing Research Repository*, 2011.

[119] L. Yuan, J. Liu, and J. Ye, "Efficient methods for overlapping group lasso," in *NIPS*, 2011, pp. 352–360.

[120] P. Zhao, G. Rocha, and B. Yu, "D the composite absolute penalties family for grouped and hierarchical variable selection," *Annals of Statistics*, vol. 37, no. 6A, pp. 3468–3497, 2009.

[121] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach, "Proximal methods for sparse hierarchical dictionary learning," in *ICML*, 2010.

[122] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[123] R. Chartrand and W. Yin, "Iteratively reweighted algorithms for compressive sensing," in *ICASSP*, 2008, pp. 3869–3872.

[124] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk, "Iteratively re-weighted least squares minimization: Proof of faster than linear rate for sparse recovery," in *CISS*, 2008, pp. 26–29.

[125] F. Nie, H. Huang, X. Cai, and C. Q. Ding, "Efficient and robust feature selection via joint $l_{2,1}$-norms minimization," in *NIPS*, 2010, pp. 1813–1821.

[126] N. S. Rao, R. D. Nowak, S. Wright, and N. Kingsbury, "Convex approaches to model wavelet sparsity patterns," in *ICIP*, 2011, pp. 1917–1920.

[127] K. Kavukcuoglu, M. Ranzato, R. Fergus, and Y. LeCun, "Learning invariant features through topographic filter maps," in *CVPR*, 2009, pp. 1605–1612.

[128] V. Cevher, M. F. Duarte, C. Hegde, and R. G. Baraniuk, "Sparse signal recovery using markov random fields," in *NIPS*, 2008, pp. 257–264.

[129] F. R. Bach, "Consistency of the group lasso and multiple kernel learning," *Journal of Machine Learning Research*, vol. 9, pp. 1179–1225, 2008.

[130] P. Ravikumar, H. Liu, J. Lafferty, and L. Wasserman, "Spam: Sparse additive models," in *NIPS*. MIT Press, 2007.

[131] J. Huang, J. L. Horowitz, and F. Wei, "Variable selection in nonparametric additive models," Tech. Rep., 2010.

[132] J. Yin, X. Chen, and E. P. Xing, "Group sparse additive models," in *ICML*, 2012.

[133] J. Fan, Y. Feng, and R. Song, "Nonparametric independence screening in sparse ultra-high dimensional additive models," *J. Amer. Statist. Assoc.*, vol. 106, pp. 544–557, 2011.

[134] M. Vijver, Y. He, and et al, "A gene-expression signature as a predictor of survival in breast cancer," *New England Journal of Medicine*, vol. 347, no. 25, 2002.

[135] A. Subramanian, P. Tamayo, and et al, "Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles," *Proceedings of the National Academy of Sciences*, vol. 102, no. 43, pp. 15 545–15 550, 2005.

[136] H. Chuang, E. Lee, Y.-T. Liu, D. Lee, and T. Ideker, "Network-based classification of breast cancer metastasis," *Molecular Systems Biology*, vol. 3, no. 140, 2007.

[137] K. Pearson, "The problem of the random walk," in *Nature*, 1905.

[138] N. Craswell and M. Szummer, "Random walks on the click graph." in *SIGIR*, 2007.

[139] L. Backstrom and J. Leskovec, "Supervised random walks: predicting and recommending links in social networks," in *WSDM*, 2011.

[140] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web," in *Technical report, Stanford University Database Group*, 1998.

[141] J. Glen and W. Jennifer, "Scaling personalized web search," in *Technical report, Stanford University Database Group*, 2002.

[142] O. Chapelle, B. Schlkopf, and A. Zien, *Semi-Supervised Learning*. MIT Press, Cambridge, MA, USA, 2006.

[143] X. Zhu, "Semi-supervised learning literature survey," University of Wisconsin-Madison, Tech. Rep. 1530, 2008.

[144] A. Blum and T. M. Mitchell, "Combining labeled and unlabeled sata with co-training," in *COLT*, 1998, pp. 92–100.

[145] T. Joachims, "Transductive learning via spectral graph partitioning," in *ICML*, 2003, pp. 290–297.

[146] E. Minkov and W. W. Cohen, "Learning to rank typed graph walks: Local and global approaches," in *WebKDD and SNA-KDD joint workshop*, 2007.

[147] H. Tong, C. Faloutsos, and J.-Y. Pan, "Fast random walk with restart and its applications," in *ICDM*, 2006.

[148] H. Tong and C. Faloutsos, "Center-piece subgraphs: Problem definition and fast solutions," in *KDD*, 2006.

[149] B. Gallagher, H. Tong, T. Eliassi-Rad, and C. Faloutsos, "Using ghost edges for classification in sparsely labeled networks," in *KDD*, 2008.

[150] J. Pearl, "Reverend bayes on inference engines: A distributed hierarchical approach," in *AAAI*, 1982, pp. 133–136.

[151] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, "Collective classification in network data," *AI magazine*, vol. 29, no. 3, pp. 93–106, 2008.

[152] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Constructing free energy approximations and generalized belief propagation algorithms," *IEEE Transactions on Information Theory*, vol. 51, pp. 2282–2312, 2005.

[153] D. Koutra, T.-Y. Ke, U. Kang, D. H. P. Chau, H.-K. K. Pao, and C. Faloutsos, "Unifying guilt-by-association approaches: Theorems and fast algorithms," in *ECML/PKDD*, 2011, pp. 1–8.

[154] G. W. Flake, S. Lawrence, C. L. Giles, and F. M. Coetzee, "Self-organization and identification of web communities," *IEEE Computer*, vol. 35, no. 3, 2002.

BIOGRAPHICAL STATEMENT

Deguang Kong started his Ph.D studying in Computer Science and Engineering department at University of Texas, Arlington from August, 2010, and defended on November, 2013. His advisor is Chris H.Q. Ding, and his major research direction is data mining, machine learning, and its applications in computer vision, text mining, cyber security, etc. He interned at Los alamos National Lab from June to December, 2012, and interned at NEC lab America (Cupertino, CA) from May to August, 2013. He published first author papers at world-leading top conferences, e.g., ICML, KDD, CVPR, ICDM, ECML/PKDD, CIKM; and SIGMETRICS(poster), INFOCOM, etc.