A NEW INTEGRATIVE DATA MINING FRAMEWORK FOR ANALYZING THE CANCER GENOME ATLAS DATA

by

DIJUN LUO

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
August 2012

To my mother.

ACKNOWLEDGEMENTS

I would like to thank my supervising Professors Dr. Heng Huang and Dr. Chris Ding, who opened a door to my Ph.D study 5 years ago. Professor Huang gave me the opportunity to study in The University of Texas at Arlington and provided a suitable environment for study and research.

I am particular thankful to Professor Ding, who educated me in the past 5 years about how to become a good researcher. One sentence from Professor Ding that I remember clearly and will remember for ever is that “for whatever we have learned, we need to understand them in our own ways”. Professor Ding changed my understanding about understanding and the way to study.

I would like to thank all members in my dissertation committee, including Professors Yu Lei, Nan Zhang, and Junzhou Huang, for their help in developing my dissertation and their helpful comments on my research work.

I would also like to extend my appreciation to Dr. Jimeng Sun, Dr. Jianying Hu, Dr. Fei Wang, and Dr. Marianthi Markatou at IBM T. J. Watson Research Center for their help in my internship in IBM.

I am grateful to all the teachers who taught me during the years I spent in the University.

Finally, I would like to express my deep gratitude to my wife who have encouraged and inspired me in my whole PhD study.

July 10, 2012

ABSTRACT

A NEW INTEGRATIVE DATA MINING FRAMEWORK FOR ANALYZING THE CANCER GENOME ATLAS DATA

DIJUN LUO, Ph.D.

The University of Texas at Arlington, 2012

Supervising Professor: Heng Huang

Besides accuracy and efficiency, understandability is another key issue of predictive modeling in real-world applications, especially in biomedical and healthcare data analysis. We develop a new integrative framework to enhance the interpretability of data by sparsity-based learning. We proposed several novel sparsity-based learning models, emphasizing different understandable properties of data, such as explicit sparsity, low redundancy, and low rank, and apply to The Cancer Genome Atlas (TCGA) data analysis. Results indicate that the proposed methods provide more insights from TCGA data while maintaining stable and competitive performances in predictive modeling. To further enhance the interpretability of biological processes and disease mechanisms, we also develop a novel visualization tool by considering heterogeneous relationships among genomics elements. By applying the novel learning models and the visualization tools, pathways of several important cancer diseases are revisited and a series of novel potential bio-markers are discovered which improves our ability to diagnosis, treat and prevent cancer.

TABLE OF CONTENTS

Appendix

LIST OF ILLUSTRATIONS

Figure Page

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Background and Motivation

Cancer has been developed to be one of the leading causes of death in the United States and many other countries. Currently, one in four people will die of cancer in the United States, since the fact that no effect treatment has been discover for cancer diseases. In total, 1,638,910 new incidents and 577,190 deaths from cancer are projected to occur in the United States in 2012. The chance of being diagnosed with an malignant cancer in a lifetime is 45% and 38% for men and women respectively [1].

Cancer is now one of the major threats to public health and is in desperate need for a cure. The Cancer Genome Atlas (TCGA) project The Cancer Genome Atlas (TCGA) began as pilot in 2006 with an investment of $50 million each from the National Cancer Institute (NCI) and National Human Genome Research Institute (NHGRI). The project was dedicated to analyze and discover genome alterations in tumors by introducing the integrated multi-dimensional/multi-view analysis which provides a unique opportunity to conduct in *silico* scientific research where multiple measurements of clinical subjects are simultaneously considered. The mission is to put comprehensive and coordinated effort to accelerate the understanding of the molecular basis of cancer through the application of genome analysis technologies, including large-scale genome sequencing. The final goal of the TCGA project it to improve our ability to diagnosis, treat and prevent cancer.

We are interested in analyzing the TCGA data in a data mining and machine learning way, especially for the purpose of deeper understanding from the biological data and providing feedback with interpretable learning models. We aim to developed integrative framework to enhance the interpretability of learning models by making use of the structures of data and by visualization.

1.2 Main Contributions

1.2.1 Multi-subspace Learning for Linear Modeling of TCGA Data

We present several important techniques in data analysis of TCGA data, include multi-subspace discovery problem and provides a theoretical solution which is guaranteed to recover the number of subspaces, the dimensions of each subspace, and the members of data points of each subspace simultaneously. We further propose a data representation model to handle noisy real world data. We develop a novel optimization approach to learn the presented model which is guaranteed to converge to global optimizers. As applications of our models, we first apply our solutions as preprocessing in a series of machine learning problems, including clustering, classification, and semi-supervised learning. We found that our method automatically obtains robust data presentation which preserves the affine subspace structures of high dimensional data and generate more accurate results in the learning tasks. We also establish a robust standalone classifier which directly utilizes our sparse and low rank representation model. Experimental results indicate the proposed methods improve the quality of data by preprocessing and the standalone classifier outperforms some state-of-the-art learning approaches. The proposed multi-subspace method is also applied to the TCGA data and interesting and consistent patterns are discovered.

1.2.2 Social Diffusion Process for Clustering

In the dissertation, a new stochastic process, called as Social Diffusion Process (SDP), is also presented to address the graph modeling. Based on this model, we derive a graph evolution algorithm and a series of graph-based approaches to solve machine learning problems, including clustering and semi-supervised learning. SDP can be viewed as a special case of *Matthew effect*, which is a general phenomenon in nature and societies. We use social event as a metaphor of the intrinsic stochastic process for broad range of data. We evaluate our approaches in a large number of frequently used datasets and compare our approaches to other state-of-the-art techniques. Results show that our algorithm outperforms the existing methods in most cases. We also applying our algorithm into the functionality analysis of microRNA and discover biologically interesting cliques. Due to the broad availability of graph-based data, our new model and algorithm potentially have applications in wide range.

1.2.3 Explicit Structured Sparse Learning for Bio-marker Identification

The dissertation enhance the interpretability of structured learning by introducing the ℓ_2/ℓ_0 norm optimization. As powerful tools, machine learning and data mining techniques have been widely applied in various areas. However, in many real world applications, besides establishing accurate *black box* predictors, we are also interested in *white box* mechanisms, such as discovering predictive patterns in data which enhance our understanding of underlying physical, biological and other natural processes. For these purposes, sparse representation and its variations have been one of the focuses. More recently, structural sparsity has attracted increasing attentions. In previous research structural sparsity was often achieved by imposing convex but non-smooth norms such as ℓ_2/ℓ_1 and group ℓ_2/ℓ_1 norms. In this dissertation, we present the explicit ℓ_2/ℓ_0 and group ℓ_2/ℓ_0 norm to directly approach structural sparsity. To tackle the problem of intractable ℓ_2/ℓ_0 optimizations, we develop a general Lipschitz auxiliary function which leads to simple iterative algorithms. In each iteration, optimal solution is achieved for the induced sub-problem and a guarantee of convergence is provided. Further more, the local convergent rate is also theoretically bounded. We test our optimization techniques in the multi-task feature learning problem.

Experimental results suggest that our approaches outperform other approaches in both synthetic and real world data sets.

1.2.4 Sparse Learning with Low-redundancy

We also developed a scalable model for risk factor and bio-marker identification. As diverse clinical information become available for analysis, a large number of features can be constructed and leveraged for predictive modeling. Feature selection is a classic analytic component that faces new challenges due to the new applications: How to handle a diverse set of high dimensional features? How to select features with high predictive power, but low redundant information? How to design methods that can select globally optimal features with theoretical guarantee? How to incorporate and extend existing knowledge driven approach? In this dissertation, we present Scalable Orthogonal Regression (SOR), an optimization-based feature selection method with the following novelties: 1) Scalable: SOR achieves nearly linear scale-up with respect to the number of input features and the number of samples; 2) Optimal: SOR is formulated as an alternative convex optimization problem with theoretical convergence and global optimality guarantee; 3) Non-redundant: thanks to the orthogonality objective, SOR is designed specifically to select less redundant features without sacrificing the quality; 4) Extensible: SOR can enhance an existing set of preselected features by adding additional features that are complement to the existing set but still with strong predictive power. In the evaluation SOR consistently outperforms several other state of the art feature selection methods in several quality metrics on several real datasets. Finally, we demonstrate a case study of a large-scale clinical application for predicting early onset of Heart Failure (HF) using real Electronic Health Records (EHRs) data of over 10K patient for over 7 years. Leveraging SOR, we are able to construct accurate and robust predictive models and derive potential clinical insights.

1.2.5 Regulatory Elements Visualization

Regulatory elements in cell, such as microRNAs (miRNAs), play important roles. Extensive efforts have been made by both biological experiments and *in silico* studies. As low-cost alternatives of biological experiments, several computational approaches have been developed to facilitate the discovery of mechanisms of these elements.

We develop novel approaches for regulatory elements analysis, including visualization and prediction. More specifically, we formalized the problem into a partial differential equation framework, and employed Green's function approach and the corresponding Direchlet boundary conditions to solve the problem. We discover a novel miRNA pattern in *H. Sapiens* in the visualization results. RNAPred achieves 100% precision by using few number of known of miRNAs. By applying RNAPred, we discover novel miRNAs in *D. Melanogaster*, which are conserved in other species.

1.3 Overview of The Rest of the Dissertation

The rest of the dissertation will be organized as follows. A multi-subspace learning algorithm will be introduced in Chapter 2, and the explicit sparsity learning will be introduced in Chapter 3. Then the low redundancy property of sparsity will be emphasized in Chapter 4 and two visualization tools will be introduced in Chapters 5 and 6. Finally a conclusion will be made in Chapter 7.

CHAPTER 2

MULTI-SUBSPACE LEARNING WITH CONVEX OPTIMIZATION

2.1 Motivations of Multi-subspace Learning

The linear sparse representation approaches recently attract attentions from the researchers in statistics and machine learning. By providing robustness, simpleness, and sound theoretical foundations, sparse representation models have been widely considered in various applications [2, 3, 4, 5].

In most previous models, we impose on the data an assumption that the data points can be linearly represented by other data points in the same class or data points nearby [6, 7]. This assumption will further lead to another assumption that subspace of each class has to include the original point. Our major argument in this chapter is that this assumption is too loose in real world applications. For this reason, we further impose the affine properties of the subspaces and present a challenging affine subspace discovery problem. To be more specific, given a set of data points, which lie on multiple unknown spaces, we want to recover the membership of data points to subspaces, *i.e.* which data point belongs to which subspace. The major challenge here is that not only the subspaces and membership are unknown, but also the number of subspaces and the dimensions of the subspaces are unknown.

In this chapter [1] we will (1) present a sparse representation learning model to obtain the solutions automatically, which is theoretically guaranteed to recover all the unknown information listed above, (2) extended our model to handle noisy data and apply the sparse representation as a preprocessing in various machine learning tasks, such as unsupervised

[1] Most of the major results in this chapter have been published in paper [8].

learning, classification and semi-supervised learning, and (3) develop a standalone classifier directly based on the sparse representation model. To handle the noisy data with robust performance, we introduce a mixed-norm optimization problem which involves trace, ℓ_2/ℓ_1, and ℓ_1 norms. We further develop an efficient algorithm to optimize the induced problem which is guaranteed to converge to a global optimizer.

Our model explicitly imposes both sparse and low rank requirements on the data presentation. We apply our model as preprocessing in various machine learning applications. The extensive and sound empirical results suggest that one might benefit from taking sparsity and low rank into consideration simultaneously.

2.2 Problem Description and Our Solution

Consider K groups data points $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_K]$ and assume that there are $n_1, n_2, \cdots, n_K$ data points in each group, respectively ($\sum_{k=1}^{K} n_k = n$). We assume that for each group, the data points belong to independent affine subspaces. And the dimensions of the affine subspaces are $d_1, d_2, \cdots, d_K$. To be more specific, for each affine subspace $\mathbf{X}_k$, there exist $d_k + 1$ bases $\mathbf{U}^k = [\mathbf{u}_1^k, \mathbf{u}_2^k, \cdots, \mathbf{u}_{d_k}^k, \mathbf{u}_{d_k+1}^k]$ and for each data point $\mathbf{x} \in \mathbf{X}_k$, there exists $\boldsymbol{\beta}$ such that $\mathbf{x} = \mathbf{U}^k \boldsymbol{\beta}^k$ and that $\boldsymbol{\beta}^\mathsf{T} \mathbf{1} = 1$. In this chapter, by the dimension of the affine subspace, we mean the characteristic dimension, *i.e.* from the manifold point of view. Even though there are $d_k + 1$ bases in $\mathbf{U}^k$, we still consider that $\mathbf{U}^k$ defines a d_k-dimensional affine subspace.

2.2.1 Multi-Subspace Discovery Problem

The problem of *Multi-Subspace Discovery* is given $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_K]$ to recover (1) the number of affine space K, (2) the dimension of each subspace d_k, and (3) the membership of the data points to the affine subspaces. The challenge in this problem is that the only known information is the input **X**, where the data points are typically disordered, and all other information is unknown.

Will illustrate the Multi-Subspace Discovery problem in Figure 2.1[2]. In this chapter, we first derive a solution of this problem and provide several theoretical analysis of our solution on non-noisy data, then extend our model to handle noisy real-world case by adding ℓ_2/ℓ_1 norms which are convex but non-smooth regularizations. We develop an efficient algorithm to solve the problem.

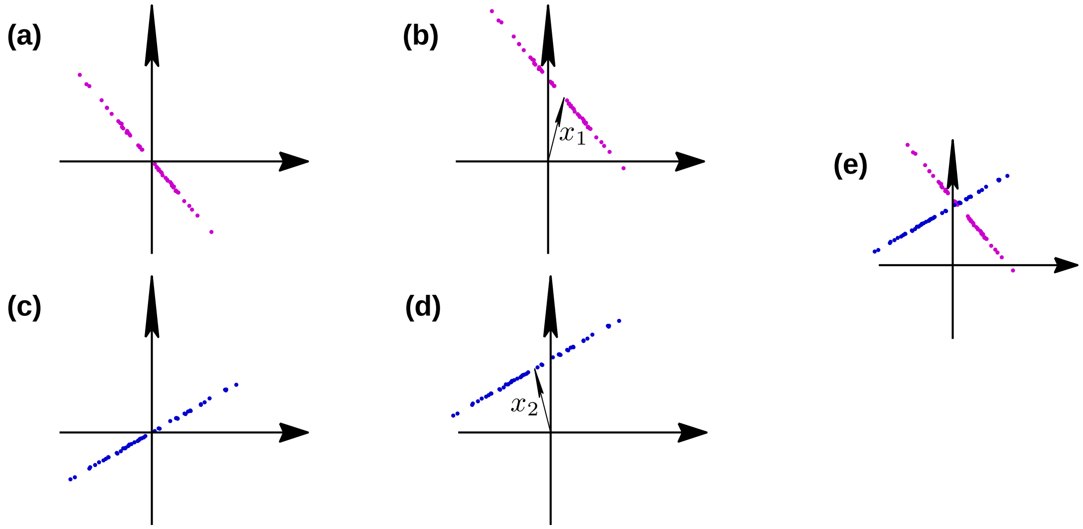


Figure 2.1. A demonstration of the Multi-Subspace Discovery problem. (a) and (c): Two groups of data points lying on two 1-dimension subspaces. (b): All data points shifted by x_1 from (a). (d): All data points shifted by x_2 from (c). (e): A mixture of data points from (b) and (d). The affine subspace clustering problem is to recover the number of subspaces (2 in this case), the membership of the data points to the subspaces (indicated by the color of the data points in (e), the dimensions of the subspaces (1 for both of the subspace in this cases)..

[2] This figure has been published in paper [8].

2.2.2 A Constructive Solution

We cast the multi-subspace discovery problem into a trace norm optimization, in which the optimizer directly gives the number of affine subspace and the membership of the clustering. The results are theoretically guaranteed.

Representation of One Subspace

In order to introduce our solution in a more interpretable way, we first solve a simple problem in which there is only one affine subspace. Let $\mathbf{X}_1 = (\mathbf{x}_1, \cdots, \mathbf{x}_{n_1})$ be in a d_1-dimensional affine subspace spanned by the basis $\mathbf{U}_1$, $d_1 + 1 < n_1$, *i.e.* for each data points $\mathbf{x}_i$, there exists $\boldsymbol{\alpha}_i$,

$$\mathbf{x}_i = \mathbf{U}_1\boldsymbol{\alpha}_i,\ \boldsymbol{\alpha}_i \in \mathbb{R}^{d_1+1},\ \boldsymbol{\alpha}_i^\intercal \mathbf{1} = 1,\ 1 \le i \le n_1 \tag{2.1}$$

or more compactly, $\mathbf{X}_1 = \mathbf{U}_1\mathbf{A},\ \mathbf{A}^\intercal\mathbf{1} = \mathbf{1}$, where $\mathbf{1}$ is a column vector with all elements one in proper size and $\mathbf{A} = (\boldsymbol{\alpha}_1, \cdots, \boldsymbol{\alpha}_{n_1})$. We define

$$\tilde{\mathbf{X}}_1 = \begin{pmatrix} \mathbf{U}_1\mathbf{A} \\ \mathbf{1}^\intercal \end{pmatrix} \tag{2.2}$$

Then we have,

Lemma 2.2.1 *If* $\mathbf{X}_1$ *satisfies Eq. (2.1) and let*

$$\mathbf{Z}_1 = \tilde{\mathbf{X}}_1^+\tilde{\mathbf{X}}_1 \tag{2.3}$$

where $\tilde{\mathbf{X}}_1$ *is defined in Eq. (2.2) and* $\tilde{\mathbf{X}}_1^+$ *is the* Moore-Penrose *pseudo inverse of* $\tilde{\mathbf{X}}_1$*, then*

$$\mathbf{X}_1 = \mathbf{X}_1\mathbf{Z}_1,\ \mathbf{1}^\intercal\mathbf{Z}_1 = \mathbf{1}^\intercal, \tag{2.4}$$

and rank$(\mathbf{Z}_1) = d_1 + 1$.

Proof By making use of the property of *Moore-Penrose* pseudo inverse, we immediately have

$$\tilde{\mathbf{X}}_1 = \tilde{\mathbf{X}}_1 \tilde{\mathbf{X}}_1^+ \tilde{\mathbf{X}}_1,$$

Thus,

$$\begin{pmatrix} \mathbf{U}_1\mathbf{A} \\ \mathbf{1}^\top \end{pmatrix} = \begin{pmatrix} \mathbf{U}_1\mathbf{A} \\ \mathbf{1}^\top \end{pmatrix} \mathbf{Z},$$

which is equivalent to two equations of

$$\mathbf{X}_1 = \mathbf{X}_1 \mathbf{Z}_1,$$

$$\mathbf{1}^\top \mathbf{Z}_1 = \mathbf{1}^\top.$$

It is obvious that $\mathrm{rank}(\mathbf{Z}_1) = \mathrm{rank}(\tilde{\mathbf{X}}_1)$. On the other hand, by the definition of $\mathbf{A}$ in Eq. (2.2), we have $\mathbf{1}^\top \mathbf{A} = \mathbf{1}^\top$, thus

$$\tilde{\mathbf{X}}_1 = \begin{pmatrix} \mathbf{U}_1\mathbf{A} \\ \mathbf{1}^\top \end{pmatrix} = \begin{pmatrix} \mathbf{U}_1\mathbf{A} \\ \mathbf{1}^\top\mathbf{A} \end{pmatrix} = \begin{pmatrix} \mathbf{U}_1 \\ \mathbf{1}^\top \end{pmatrix} \mathbf{A} \tag{2.5}$$

From Eq. (2.2) we have

$$\mathrm{rank}(\tilde{\mathbf{X}}_1) \geq \mathrm{rank}(\mathbf{U}_1\mathbf{A}) = \mathrm{rank}(\mathbf{X}_1) = d_1 + 1$$

But from Eq. (2.5) we have

$$\mathrm{rank}(\tilde{\mathbf{X}}_1) \leq \mathrm{rank}(\mathbf{A}) = d_1 + 1.$$

Thus $\mathrm{rank}(\mathbf{Z}_1) = \mathrm{rank}(\tilde{\mathbf{X}}_1) = d_1 + 1$.

Since $d_1 + 1 < n_1$, $\mathbf{Z}_1$ is low rank. Interestingly, this low-rank affine subspace presentation of Eqs. (2.1, 2.4) can be reformulated as a trace norm optimization problem:

$$\min_{\mathbf{Z}_1} \|\mathbf{Z}_1\|_* \quad \text{s.t.} \quad \mathbf{X}_1 = \mathbf{X}_1\mathbf{Z}_1, \ \mathbf{1}^\mathsf{T}\mathbf{Z}_1 = \mathbf{1}^\mathsf{T} \tag{2.6}$$

where $\|\mathbf{Z}_1\|_*$ is the trace norm of $\mathbf{Z}_1$, *i.e.* the sum of singular values, or explicitly,

Lemma 2.2.2 $\mathbf{Z}_1$ *defined in Eq. (2.3) is an optimizer of the problem in Eq. (2.6).*

The proof of Lemma 2.2.2 requires Lemma 2.4.1. We will introduce the proof Lemma 2.2.2 later.

In this chapter, we hope to recover multiple $\mathbf{Z}$ which has diagonal block structure from $\mathbf{X}$ by which we solve the multi-subspace discovery problem.

Constructive Representation of K Subspaces

Now consider the full case where the data points $\mathbf{X}$ belong exactly to K independent subspaces. Assume data points within a subspace are indexed sequentially, $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_K]$. Repeat the above analysis for each subspace, we have

$$\mathbf{X} = [\mathbf{X}_1, \cdots, \mathbf{X}_K] = [\mathbf{X}_1\mathbf{Z}_1, \cdots, \mathbf{X}_K\mathbf{Z}_K] = \mathbf{X}\mathbf{Z}, \tag{2.7}$$

where

$$\mathbf{Z} = \begin{pmatrix} \mathbf{Z}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{Z}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \mathbf{Z}_K \end{pmatrix} \tag{2.8}$$

Thus by construction, we have the following,

Theorem 2.2.3 *If* $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]$ *belong exactly to K subspaces of rank* d_k *respectively, there exists* $\mathbf{Z}$*, such that*

$$\mathbf{X} = \mathbf{X}\mathbf{Z},\ \mathbf{1}^\intercal\mathbf{Z} = \mathbf{1}^\intercal. \tag{2.9}$$

where $\mathbf{Z}$ *has the structure of Eq.(2.8) and rank*$(\mathbf{Z}_k) = d_k + 1, 1 \leq k \leq K.$

Proof Since we have Lemma 2.2.1, the proof of Theorem 1 is straightforward by construction. Let $\mathbf{X} = [\mathbf{X}_1, \cdots, \mathbf{X}_K]$, then by Lemma 1, we can always rewrite as $\mathbf{X} = [\mathbf{X}_1\mathbf{Z}_1, \cdots, \mathbf{X}_K\mathbf{Z}_K]$, or

$$\mathbf{X} = [\mathbf{X}_1, \cdots, \mathbf{X}_K]\begin{pmatrix} \mathbf{Z}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{Z}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \mathbf{Z}_K \end{pmatrix}.$$

Recovery of The Multiple Subspaces

Intuited by Lemma 2.2.2, and Theorem 2.2.3, one might hypothetically consider recovering the block structure by using the following optimization,

$$\min_{\mathbf{Z}} \|\mathbf{Z}\|_* \ \text{ s.t. } \mathbf{X} = \mathbf{X}\mathbf{Z},\ \mathbf{1}^\intercal\mathbf{Z} = \mathbf{1}^\intercal, \tag{2.10}$$

which is a convex problem since the objective function $\|\mathbf{Z}\|_*$ is a convex function *w.r.t* $\mathbf{Z}$ and the domain constraints $\mathbf{X} = \mathbf{X}\mathbf{Z},\ \mathbf{1}^\intercal\mathbf{Z}_1 = \mathbf{1}^\intercal$ is an affine space, which is a convex domain. This is desirable property: if a solution $\mathbf{Z}^*$ is a local solution, $\mathbf{Z}^*$ must be a global solution. However, a convex optimization could have multiple global solutions, *i.e.*, the global solution is not unique.

This optimization indeed has one optimal solution:

Theorem 2.2.4 *The optimization problem of Eq. (2.10) has the optimal solution*

$$\mathbf{Z}^* = \tilde{\mathbf{X}}^+ \tilde{\mathbf{X}} \tag{2.11}$$

where

$$\tilde{\mathbf{X}} = \begin{pmatrix} \mathbf{X} \\ \mathbf{1}^\intercal \end{pmatrix}. \tag{2.12}$$

Theorem 2.2.4 can be directly derived from Lemma 2.2.1, by replacing $\mathbf{X}_1$ with $\mathbf{X}$.

In general, $\mathbf{Z}^*$ is not sparse and does not have the sparse block structure of $\mathbf{Z}$ in Eq. (2.8). Similar data representation model was represented in [9], which suffers from the same problem. Here we extend the model to solve the general multi-subspace problem and provide a proof of the solution.

To recover a solution which has the sparse structure of Eq. (2.8), we add a ℓ_1 term to optimization Eq. (2.10) to promote sparsity of the solution, and optimize the following

$$\begin{aligned} &\min_{\mathbf{Z}} J_1(\mathbf{Z}) = \|\mathbf{Z}\|_* + \delta\|\mathbf{Z}\|_1 \\ &\text{s.t. } \mathbf{X} = \mathbf{X}\mathbf{Z},\ \mathbf{1}^\intercal \mathbf{Z}_1 = \mathbf{1}^\intercal, \end{aligned} \tag{2.13}$$

where $\|\mathbf{Z}\|_1$ is the element-wise ℓ_1 norm: $\|\mathbf{Z}\|_1 = \sum_{ij} |Z_{ij}|$ and δ is model parameter which control the balance between low rank and sparsity. In our theoretical studies, we only require $\delta > 0$.

And fortunately, for problem Eq.(2.13), we have the following theorem,

Proposition 2.2.5 *Assume $\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_K$ are independent affine subspaces. Let $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_K]$, then all the minimizers of problem Eq.(2.13) have the form of Eq.(2.8). Further more, each block $\mathbf{Z}_k$ has only one connected component.*

By *independent*, we mean for any $\mathbf{x}$ in the k-th group, $\mathbf{x}$ can not be represented by all the data points not in the k-th group. Or explicitly,

$$\forall k, \mathbf{P}_k \cap \mathbf{P}_{-k} = \Phi, \tag{S4.1}$$

where Φ is the empty set, $\mathbf{P}_k$ is the space spanned by all the data points in group k, and $\mathbf{P}_{-k}$ is the space spanned by all the data points not in group k.

$$\mathbf{P}_k = \left\{\mathbf{x} : \mathbf{x} = \mathbf{X}_k \boldsymbol{\alpha}_k,\ \boldsymbol{\alpha}_k^\top \mathbf{1} = 1,\ \boldsymbol{\alpha}_k \in \mathbb{R}^{n_k}\right\},$$
$$\mathbf{P}_{-k} = \left\{\mathbf{x} : \mathbf{x} = \mathbf{X}_{-k} \boldsymbol{\alpha}_{-k},\ \boldsymbol{\alpha}_{-k}^\top \mathbf{1} = 1,\ \boldsymbol{\alpha}_{-k} \in \mathbb{R}^{n-n_k}\right\},$$

where

$$\mathbf{X}_{-k} = [\mathbf{X}_1, \cdots, \mathbf{X}_{k-1}, \mathbf{X}_{k+1}, \cdots, \mathbf{X}_K],$$

and n_k is the number of data points in the k-th group $\mathbf{X}_k$.

Proof. We first introduce the following Lemma.

Lemma 2.2.6 *(Lemma 3.1 in paper [9]) Let A D, B and C be matrices of compatible dimension, the following always holds,*

$$\left\| \begin{matrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{matrix} \right\|_* \geq \left\| \begin{matrix} \mathbf{A} & 0 \\ 0 & \mathbf{D} \end{matrix} \right\|_* = \|\mathbf{A}\|_* + \|\mathbf{D}\|_*. \tag{2.15}$$

Notice that this Lemma here is a little different from Lemma 3.1 in paper [9] in the sense that we here do not require $\mathbf{A}$ and $\mathbf{D}$ to be square matrices[3]. We prove the first part of the theorem by showing that for any non-block diagonal matrix $\mathbf{Z}'$, $\mathbf{Z}'$ is not an optimizer of Eq. (13). Assume $\mathbf{Z}'$ is an optimizer of Eq. (13) and is non-block diagonal. Without loss

[3] The proof is similar.

of generality, we assume the t-th column of $\mathbf{Z}'$ in the first block has off-block diagonal elements, *i.e.*

$$\mathbf{Z}' = \begin{pmatrix} & Z'_{1,t} & \\ & Z'_{2,t} & \\ & \vdots & \\ \mathbf{z}'_1, \mathbf{z}'_2, \cdots, & Z'_{n_1,t} & , \mathbf{z}'_{t+1}, \cdots, \mathbf{z}'_n \\ & Z'_{n_1+1,t} & \\ & \vdots & \\ & Z'_{n,t} & \end{pmatrix},$$

where $\mathbf{z}'_j$ is the j-th column of $\mathbf{Z}'$ and Z_{ij} is the i-th element of $\mathbf{z}'_j$. Let

$$\mathbf{z}'_t = \mathbf{z}^b + \mathbf{z}^o,$$

where

$$\mathbf{z}^b = \begin{pmatrix} Z'_{1,t} \\ Z'_{2,t} \\ \vdots \\ Z'_{n_1,t} \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \mathbf{z}^o = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ Z'_{n_1+1,t} \\ \vdots \\ Z'_{n,t} \end{pmatrix}.$$

$\mathbf{z}^o \neq \mathbf{0}$ here. Since $\mathbf{Z}'$ is an optimizer of Eq. (10), $\mathbf{X} = \mathbf{X}\mathbf{Z}'$, or $\mathbf{x}_j = \mathbf{X}\mathbf{z}'_j$, $j = 1, \cdots, n$. Obviously,

$$\mathbf{x}_t = \mathbf{X}\mathbf{z}'_t \in \mathbf{P}_1,$$

Due to Eq. (S4.1), one must have

$$\mathbf{X}\mathbf{z}^b \in \mathbf{P}_1,\ (\mathbf{z}^b)^\intercal \mathbf{1} = 1.$$

Then

$$(\mathbf{z}^o)^\intercal \mathbf{1} = 0,\ \mathbf{X}\mathbf{z}^o = 0,$$

or

$$\mathbf{x}_t = \mathbf{X}\mathbf{z}^b.$$

Let $\mathbf{Z}'_b = [\mathbf{z}'_1, \cdots, \mathbf{z}^b, \mathbf{z}'_{t+1}, \cdots, \mathbf{z}'_n]$. Then $\mathbf{X} = \mathbf{X}\mathbf{Z}'_b$. However, since $\mathbf{z}^o \neq \mathbf{0}$, we have

$$\|\mathbf{Z}'\|_1 = \sum_{j=1, j\neq t,}^{n} \|\mathbf{z}'_j\|_1 + \|\mathbf{z}'_t\|_1 > \sum_{j=1, j\neq t,}^{n} \|\mathbf{z}'_j\|_1 + \|\mathbf{z}'_b\|_1 = \|\mathbf{Z}'_b\|_1.$$

And by the Lemma 2.2.6, we have

$$\|\mathbf{Z}'\|_* \geq \|\mathbf{Z}'_b\|_*,$$

or

$$\|\mathbf{Z}'\|_* + \delta\|\mathbf{Z}'\|_1 > \|\mathbf{Z}'_b\|_* + \delta\|\mathbf{Z}'_b\|_1$$

(remember $\delta > 0$) indicating $\mathbf{Z}'_b$ satisfies the constraint in Eq. (13) and has a lower objective function value than $\mathbf{Z}'$. Thus $\mathbf{Z}'$ is impossible to be the optimizer of Eq. (13). This completes the proof of the first of the theorem.

The second part of the theorem is obvious. Without loss of generalization, let us consider the first block $\mathbf{X}_1$. Assume all the data points in $\mathbf{X}_1$ share the same base $\mathbf{U}^1$ with rank d_1 and $\mathbf{Z}_1$ can be further separated into $\mathbf{Z}_1^1$, and $\mathbf{Z}_1^2$, then the corresponding $\mathbf{X}_1^1$ and $\mathbf{X}_1^2$ must have dimension d_1^1 and d_1^2 and $d_1^1 + d_1^2 = d_1$. Then Lemmas 1 and 2, the total rank of $\mathbf{Z} = \mathrm{rank}(\mathbf{Z}_1^1) + \mathrm{rank}(\mathbf{Z}_1^2) = d_1^1 + 1 + d_1^2 + 1 = d_1 + 2$. However, still by Lemmas 1, and 2, such $\mathbf{Z}_1$ is not possible to be the optimizer of Eq. (13). Thus $\mathbf{Z}_1$ has to be a single connected component.

Since each block $\mathbf{Z}_k$ has only one connected component and all the whole $\mathbf{Z}$ is block diagonal, the number of affine subspaces is trivial to recovered, which is the number of connected components of $\mathbf{Z}$. The membership of each data points to the affine spaces is also guaranteed to be recovered.

2.2.3 More Theoretical Analysis

In previous research by Liu *et al.* [9], the theoretical properties of low-rank representation have been discussed. Here we investigate more surprising results on these representations.

First we have similar result on the following problem.

$$\min_{\mathbf{Z}} \|\mathbf{Z}\|_* \quad \text{s.t.} \quad \mathbf{X} = \mathbf{X}\mathbf{Z}, \tag{2.16}$$

Theorem 2.2.7 *The optimization problem of Eq. (2.16) has the optimal solution*

$$\mathbf{Z}^* = \mathbf{X}^+\mathbf{X} \tag{2.17}$$

and $\|\mathbf{Z}^*\|_* = rank(\mathbf{Z}^*) = rank(\mathbf{X})$.

The proof is similar to Lemma 2.2.2 and we omit it here.

Surprisingly for the following problem,

$$\min_{\mathbf{Z}} \frac{1}{2}||\mathbf{Z}||_F^2 \quad \text{s.t.} \quad \mathbf{X} = \mathbf{X}\mathbf{Z}, \tag{2.18}$$

where $||\mathbf{Z}||_F$ denotes the Frobenius norm $||\mathbf{Z}||_F = \sqrt{\sum_{ij} Z_{ij}^2}$, we also have

Theorem 2.2.8 *The optimization problem of Eq. (2.18) has the unique optimal solution*

$$\mathbf{Z}^* = \mathbf{X}^+\mathbf{X} \tag{2.19}$$

and $||\mathbf{Z}^*||_F^2 = rank(\mathbf{Z}^*) = rank(\mathbf{X})$.

Proof We write the Lagrangian function of Eq. (2.18) as

$$\mathcal{L}(\mathbf{Z}, \Lambda) = \frac{1}{2}||\mathbf{Z}||_F^2 - \mathbf{tr}(\mathbf{X} - \mathbf{X}\mathbf{Z})^\intercal \Lambda \tag{2.20}$$

We prove the theorem by showing that there exist Λ^* such that both of the following hold,

1. $\partial \mathcal{L}/\partial \mathbf{Z} = 0$ where $\mathbf{Z} = \mathbf{Z}^*$ and $\Lambda = \Lambda^*$
2. $\mathbf{X} = \mathbf{X}\mathbf{Z}^*$

One can find in the proof Lemma 2.2.2 that the second condition $\mathbf{X} = \mathbf{X}\mathbf{Z}^*$ automatically holds. And now show the first condition.

$$\frac{\partial \mathcal{L}}{\partial \mathbf{Z}} = \mathbf{Z} - \mathbf{X}^\intercal \Lambda = 0. \tag{2.21}$$

Let $\mathbf{Z} = \mathbf{Z}^* = \mathbf{X}^+\mathbf{X}$ and $\Lambda = \Lambda^* = \mathbf{U}\Sigma^{-1}\mathbf{V}$ where $\mathbf{U}\Sigma V^\intercal = \mathbf{X}$ is the SVD decomposition of $\mathbf{X}$. Then

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \mathbf{Z}} &= \mathbf{Z} - \mathbf{X}^\intercal \Lambda && (2.22)\\
&= \mathbf{V}\Sigma^{-1}\mathbf{U}^\intercal\mathbf{U}\Sigma\mathbf{V}^\intercal - \mathbf{V}\Sigma\mathbf{U}^\intercal\mathbf{U}\Sigma^{-1}\mathbf{V}^\intercal && (2.23)\\
&= \mathbf{V}\mathbf{V}^\intercal - \mathbf{V}\mathbf{V}^\intercal && (2.24)\\
&= 0 && (2.25)
\end{aligned}
$$

Theorems 2.2.4 and 2.2.8 indicate that choosing the smallest Frobenius norm of Eq. (2.16) gives the same results [4].

2.3 Multi-Subspace Representation With Noise

Typically data are drawn from multiple subspaces but with noise. Thus $\mathbf{X} = \mathbf{X}\mathbf{Z}$ does not hold anymore for any low rank $\mathbf{Z}$. On the other hand, we can combine the two constraints in Eq. (2.13) as,

$$\begin{pmatrix} \mathbf{X} \\ \mathbf{1}^\intercal \end{pmatrix} = \begin{pmatrix} \mathbf{X} \\ \mathbf{1}^\intercal \end{pmatrix} \mathbf{Z}. \qquad (2.26)$$

With the notation of $\tilde{\mathbf{X}}$ in Eq. (2.12), we have $\tilde{\mathbf{X}} = \tilde{\mathbf{X}}\mathbf{Z}$. We may express the relationship as $\tilde{\mathbf{X}} = \tilde{\mathbf{X}}\mathbf{Z} + \mathbf{E}$, where $\mathbf{E}$ represents noise. To handle such noise case, in the optimization objective of Eq.(2.13), we add the term

$$\|\mathbf{E}\|_{\ell_2/\ell_1} = \sum_j \sqrt{\sum_i \mathbf{E}_{ij}^2} = \sum_{j=1}^n \left\| \begin{pmatrix} \mathbf{x}_j \\ 1 \end{pmatrix} - \begin{pmatrix} \mathbf{X} \\ \mathbf{1}^\intercal \end{pmatrix} \mathbf{z}_j \right\|.$$

[4] Notice that this theoretical results is different from Liu *et al.*'s statement in Section 3.2 of paper [9].

This is the ℓ_2/ℓ_1-norm of matrix of $\mathbf{E}$. This norm is more robust against outliers than the usual Frobenius norm. With this noise correction term, we solve,

$$\min_{\mathbf{Z}} \|\tilde{\mathbf{X}} - \tilde{\mathbf{X}}\mathbf{Z}\|_{\ell_2/\ell_1} + \lambda\|\mathbf{Z}\|_* + \delta\|\mathbf{Z}\|_1, \tag{2.27}$$

where λ and δ are parameters which control the importance of $\|\mathbf{Z}\|_*$ and $\mathbf{Z}_1$, respectively.

2.3.1 Multi-Subspace Representation

Notice that if the data contain noise and the constraints in Proposition 2.2.5 do not hold, we lose the guarantee of the block diagonal structure of $\mathbf{Z}$. However, since the low rank and sparsity regularizer of Eq. (2.27), the final solution $\mathbf{Z}$ can be interpreted as representation coefficient of $\mathbf{X}$. We call such representation as Multi-Subspace Representation (MSR).

In summary, MSR representation of data $\mathbf{X}$ is given by the following:
(1) From input data $\mathbf{X}$, solve the optimization Eq.(2.27) to obtain $\mathbf{Z}$;
(2) The MSR representation of $\mathbf{X}$ is $\mathbf{X}\mathbf{Z}$, i.e., the representation of $\mathbf{x}_i$ is $\mathbf{X}\mathbf{z}_i$.

In §4, we develop an algorithm to solve Eq. (2.27) and in §5, some applications of our model in machine learning are given.

2.3.2 Relation to Previous Work

The MSR representation here is motivated by the affine subspace clustering problem. However, some properties of the representation have been investigated in previous work by other researchers. First notice that $\mathbf{Z}$ is sparse, the representation of $\mathbf{x}_i \approx \mathbf{Z}\mathbf{z}_i$ is similar to the one in sparse coding [10, 11]. Interestingly, research in other communities suggests that in the natural process and even in human cognition, information is often organized in a sparse way, *e.g.* Vinge *et al.* discover that primary visual cortex (area *V1*) uses a sparse code to efficiently represent natural scenes [12].

In the sparse representation model, for each testing object, we seek a sparse representation of the testing object by all objects in training data set. Such learning mechanisms implicitly learn the structure, under the assumption that the sparse representation coefficients are imbalanced among groups. To be more specific, given a set of training data $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]$ ($p \times n$ matrix, where p is the dimension of the data) and a testing data point $\mathbf{x}_t$, they solve the following optimization problem

$$\min_{\alpha_t} ||\mathbf{x}_t - \mathbf{X}\alpha_t||^2 + \lambda||\alpha_t||_1, \tag{2.28}$$

where α_t ($n \times 1$ vector) has the reconstruction coefficients of x_t using all the training data objects $\mathbf{X}$, λ is the model parameter, and $||\cdot||_1$ is the ℓ_1 norm: $||a||_1 = \sum_{i=1} |a_i|$.

Wright *et al* introduce the Sparse Represented-Based Classification method [6], which uses the following strategy for class prediction,

$$\arg\min_k r_k = ||\mathbf{x}_t - \mathbf{X}\alpha_t^k||, \tag{2.29}$$

where r_k is the representation error using the training samples in group k and α_t^k is obtained by setting the coefficients in α_t, corresponding to training samples not in class k, to zero, *i.e.*

$$\alpha_t^k(i) = \begin{cases} \alpha_t(i) & \text{if } i \in C_k, \\ 0 & \text{otherwise}, \end{cases}$$

where C_k is a set of all data points in class $k, k = 1, 2, \cdots, K$, and K is the number of classes.

On the other hand, **Z** in our model is also low rank, which is a natural requirement of most of data representation techniques, such as the low rank kernel methods [13] and robust Principle Component Analysis [14]. One can easily find literacy of the low rank representation in real world applications in various domains which indicates that low rank is one of the intrinsic properties of the data we observe, *e.g.* the missing value recover of DNA microarrays [15].

By combining the two basic properties (sparsity and low rank), our model naturally captures a proper representation of the data. We will demonstrate the quality of such representation using comprehensive empirical evidences in the experimental section.

Here **Z** can be viewed as the similarity between objects. And there also exists approaches for directly similarity learning[16, 17].

2.4 An Efficient Algorithm and Analysis

2.4.1 Outline of The Algorithm

Assume we are solving a general problem of

$$J(\mathbf{x}) = f(\mathbf{x}) + \phi(\mathbf{x}), \tag{2.30}$$

where $f(\mathbf{x})$ is smooth and $\phi(\mathbf{x})$ is non-smooth and convex. If one of the elements in subgradient of $\phi(\mathbf{x})$ can be written as product of $g(\mathbf{x})$ and $h(\mathbf{x})$, *i.e.*,

$$g(\mathbf{x})h(\mathbf{x}) \in \partial\phi(\mathbf{x}),$$

where $h(\mathbf{x})$ is smooth and $\partial\phi(\mathbf{x})$ is the subgradient of $\phi(x)$, then instead of solving Eq. (2.30), we iteratively solve the following,

$$\mathbf{x}^{t+1} = \arg\min_{\mathbf{x}} \tilde{J}(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x}^t) \int h(\mathbf{x}) d\mathbf{x}. \tag{2.31}$$

Notice that $\partial\tilde{J}(\mathbf{x})/\partial\mathbf{x} \in \partial J(\mathbf{x})$ when $\mathbf{x} = \mathbf{x}^t$. Hopefully, at convergence, $\mathbf{x}^{t+1} = \mathbf{x}^t$, then $\mathbf{0} \in \partial J(\mathbf{x})$ at $\mathbf{x}^t$, which means $\mathbf{x}^t$ is an optimizer of $J(\mathbf{x})$.

In general, the iterative steps in Eq. (2.31) cannot guarantee the convergence of $\mathbf{x}$ (*i.e.* $\mathbf{x}^{t+1} = \mathbf{x}^t$), and even the convergence of $J(\mathbf{x})$ (*i.e.* $J(\mathbf{x}^{t+1}) = J(\mathbf{x}^t)$). Fortunately, in our case of Eq. (2.27), our optimization technique guarantees both, and thus our algorithm guarantees to be an optimizer. Further more, in our algorithm, optimization problem in Eq. (2.31) has a close form solution, thus our algorithm is efficient.

2.4.2 Optimization Algorithm

Here we first present the optimization algorithm of Eq.(2.27), and then present theoretical analysis of the algorithm.

The algorithm is summarized in Algorithm 1. In the algorithm, $\mathbf{z}_i$ denotes the i-th column of $\mathbf{Z}$. The converged optimal solution is only weakly dependent on parameter. We set δ to $\delta = 1$. ϵ is an auxiliary constant for improving numerical stability in computing trace norm. We set $\epsilon = 10^{-8}$ in all experiments.

In the third line of the *for* loop, we are actually solving the problem in Eq. (2.31). In practice, we do not explicitly compute the inverse. Instead, we solve the following linear equation to obtain $\mathbf{z}_i$,

$$\left[\tilde{\mathbf{X}}^\intercal\tilde{\mathbf{X}} + \lambda\mathbf{d}_i\left(\mathbf{B} + \delta\mathbf{D}\right)\right]\mathbf{z}_i = \tilde{\mathbf{X}}^\intercal\tilde{\mathbf{x}}_i. \tag{2.32}$$

The algorithm is simple which involves no other optimization procedures. The algorithm generally converges in about 10 iterations in our experiments.

Algorithm 1 $(\mathbf{X}, \lambda, \delta)$

```
Input: Data $\mathbf{X}$, model parameters $\lambda, \delta$
Output: $\mathbf{Z}$ which optimizes Eq.(2.27).
Initialization: Compute $\tilde{\mathbf{X}}$ using Eq. (2.12), $\mathbf{Z} = \mathbf{0}$.
while not converged do
    $\mathbf{B} = (\mathbf{Z}\mathbf{Z}^\intercal + \epsilon\mathbf{I})^{-1/2}$
    for $i = 1 : n$ do
        $\mathbf{d}_i = \|\tilde{\mathbf{x}}_i - \tilde{\mathbf{X}}\mathbf{z}_i\|$,
        $\mathbf{D}_i = \mathbf{diag}\left(Z_{1i}^{-1}, Z_{2i}^{-1}, \cdots, Z_{ni}^{-1}\right)$,
        $\mathbf{z}_i = \left[\tilde{\mathbf{X}}^\intercal\tilde{\mathbf{X}} + \lambda\mathbf{d}_i\left(\mathbf{B} + \delta\mathbf{D}\right)\right]^{-1}\tilde{\mathbf{X}}^\intercal\tilde{\mathbf{x}}_i$,
    end for
end while
Output: $\mathbf{Z}$
```

We have developed theoretical analysis for this algorithm, convering three properties for this algorithm: convergence, objective function value decreasing monotonically, and converging to global solution.

2.4.3 Theoretical Analysis of Algorithm 1

Before presenting the main theories for Algorithm 1, we first introduce two useful lemmas here.

Lemma 2.4.1

$$\|\mathbf{Z}\|_* = \lim_{\epsilon\to 0} \mathbf{tr}\left(\mathbf{Z}\mathbf{Z}^\intercal + \epsilon\mathbf{I}\right)^{1/2}, \tag{2.33}$$

and

$$\lim_{\epsilon\to 0}\left(\mathbf{Z}\mathbf{Z}^\intercal + \epsilon\mathbf{I}\right)^{-1/2}\mathbf{Z} \in \partial\|\mathbf{Z}\|_*, \tag{2.34}$$

where $\partial\|\mathbf{Z}\|_*$ is the subgradient of trace norm.

Here $\epsilon\mathbf{I}$ is introduced for numerical stability.

Proof

Let

$$\mathbf{Z} = \mathbf{U}\begin{pmatrix} \sigma_1 & \cdots & 0 & \\ \vdots & \ddots & \vdots & \mathbf{0} \\ 0 & \cdots & \sigma_m & \\ & \mathbf{0} & & \mathbf{0} \end{pmatrix}\mathbf{V}^\intercal, \mathbf{U}\mathbf{U}^\intercal = \mathbf{I}, \mathbf{V}\mathbf{V}^\intercal = \mathbf{I},$$

be the SVD decomposition of $\mathbf{Z}$. Then

$$\begin{aligned}
&\lim_{\epsilon\to 0}\mathbf{tr}\left(\mathbf{Z}\mathbf{Z}^\intercal + \epsilon\mathbf{I}\right)^{1/2} \\
=&\lim_{\epsilon\to 0}\mathbf{tr}\mathbf{U}\begin{pmatrix} \sigma_1^2+\epsilon & \cdots & 0 & \\ \vdots & \ddots & \vdots & \mathbf{0} \\ 0 & \cdots & \sigma_m^2+\epsilon & \\ & \mathbf{0} & & \epsilon\mathbf{I} \end{pmatrix}^{1/2}\mathbf{U}^\intercal \\
=&\lim_{\epsilon\to 0}\mathbf{tr}\begin{pmatrix} \sqrt{\sigma_1^2+\epsilon} & \cdots & 0 & \\ \vdots & \ddots & \vdots & \mathbf{0} \\ 0 & \cdots & \sqrt{\sigma_m^2+\epsilon} & \\ & \mathbf{0} & & \sqrt{\epsilon}\mathbf{I} \end{pmatrix}\mathbf{U}^\intercal\mathbf{U} \\
=&\lim_{\epsilon\to 0}\sum_{i=1}^{m}\sqrt{\sigma_i^2+\epsilon} + (n-m)\sqrt{\epsilon} \\
=&\sum_{i=1}^{m}\sigma_i \\
=&\|Z\|_*.
\end{aligned}$$

On the other hand,

$$\lim_{\epsilon\to 0} (\mathbf{Z}\mathbf{Z}^\top + \epsilon\mathbf{I})^{-1/2}\,\mathbf{Z} = \lim_{\epsilon\to 0} \mathbf{U} \begin{pmatrix} \frac{\sigma_1}{\sqrt{\sigma_1^2+\epsilon}} & \cdots & 0 & \\ \vdots & \ddots & \vdots & \mathbf{0} \\ 0 & \cdots & \frac{\sigma_1}{\sqrt{\sigma_1^2+\epsilon}} & \\ & \mathbf{0} & & \mathbf{0} \end{pmatrix}^{1/2} \mathbf{V}^\top = \mathbf{U} \begin{pmatrix} 1 & \cdots & 0 & \\ \vdots & \ddots & \vdots & \mathbf{0} \\ 0 & \cdots & 1 & \\ & \mathbf{0} & & \mathbf{0} \end{pmatrix} \mathbf{V}^\top$$

Denote the above by $\mathbf{E}$. By following papers [18, 19] or explicitly Eq. (3.2) in paper [20], for all $\mathbf{Z}$, we have

$$\partial\|\mathbf{Z}\|_* = \left\{\mathbf{E} + \mathbf{W} : \mathbf{W} \in \mathbb{R}^{n\times n}, \mathbf{P}_\mathbf{U}\mathbf{W} = 0, \mathbf{W}\mathbf{P}_\mathbf{V} = 0, \|\mathbf{W}\| \leq 1\right\},$$

where

$$\mathbf{P}_\mathbf{U} = \mathbf{U} \begin{pmatrix} 1 & \cdots & 0 & \\ \vdots & \ddots & \vdots & \mathbf{0} \\ 0 & \cdots & 1 & \\ & \mathbf{0} & & \mathbf{0} \end{pmatrix} \mathbf{U}^\top,$$

and

$$\mathbf{P}_\mathbf{V} = \mathbf{V} \begin{pmatrix} 1 & \cdots & 0 & \\ \vdots & \ddots & \vdots & \mathbf{0} \\ 0 & \cdots & 1 & \\ & \mathbf{0} & & \mathbf{0} \end{pmatrix} \mathbf{V}^\top,$$

Obviously, $\mathbf{W} = 0$ is a special case in $\partial\|\mathbf{Z}\|_*$, or

$$\mathbf{E} \in \partial\|\mathbf{Z}\|_*,$$

which completes the proof.

Lemma 2.4.2 *Assume matrices* $\mathbf{Z}$ *and* $\mathbf{Y}$ *have the same size. Let* $\mathbf{A} = (\mathbf{Y}\mathbf{Y}^\intercal + \epsilon\mathbf{I})^{1/2}$ *and* $\mathbf{B} = (\mathbf{Z}\mathbf{Z}^\intercal + \epsilon\mathbf{I})^{1/2}$*. Then the following holds*

$$\mathbf{trA} - \mathbf{trB} + \frac{1}{2}\mathbf{tr}\mathbf{Z}^\intercal\mathbf{B}^{-1}\mathbf{Z} - \frac{1}{2}\mathbf{tr}\mathbf{Y}^\intercal\mathbf{B}^{-1}\mathbf{Y} \leq 0. \tag{2.35}$$

Proof

$$\begin{aligned}
&\mathbf{trA} - \mathbf{trB} + \frac{1}{2}\mathbf{tr}\mathbf{Z}^\intercal\mathbf{B}^{-1}\mathbf{Z} - \frac{1}{2}\mathbf{tr}\mathbf{Y}^\intercal\mathbf{B}^{-1}\mathbf{Y}\\
=&\mathbf{trA} - \mathbf{trB} + \frac{1}{2}\mathbf{tr}\mathbf{B}^{-1}\left(\mathbf{Z}\mathbf{Z}^\intercal - \mathbf{Y}\mathbf{Y}^\intercal\right)\\
=&\frac{1}{2}\mathbf{tr}\mathbf{B}^{-1}\left(2\mathbf{BA} - 2\mathbf{B}^2 + \mathbf{Z}\mathbf{Z}^\intercal - \mathbf{Y}\mathbf{Y}^\intercal\right)\\
=&\frac{1}{2}\mathbf{tr}\mathbf{B}^{-1}\left(2\mathbf{BA} - 2\mathbf{B}^2 + \mathbf{Z}\mathbf{Z}^\intercal + \epsilon\mathbf{I} - \mathbf{Y}\mathbf{Y}^\intercal - \epsilon\mathbf{I}\right)\\
=&\frac{1}{2}\mathbf{tr}\mathbf{B}^{-1}\left(2\mathbf{BA} - \mathbf{B}^2 - \mathbf{A}^2\right)\\
=&-\frac{1}{2}\mathbf{tr}\mathbf{B}^{-1/2}\left(\mathbf{A} - \mathbf{B}\right)^2\mathbf{B}^{-1/2} \leq 0.
\end{aligned}$$

One should notice that here $\mathbf{A}$ and $\mathbf{B}$ are symmetric full rank matrices.

Lemma 2.4.2 serves as a crucial part of our main theorem, which is stated as follows,

Theorem 2.4.3 *Algorithm 1 monotonically decreases the following objective,*

$$\min_{\mathbf{Z}} J(\mathbf{Z}) = ||\tilde{\mathbf{X}} - \tilde{\mathbf{X}}\mathbf{Z}||_{\ell_2/\ell_1} + \lambda\mathbf{tr}\left(\mathbf{Z}\mathbf{Z}^\intercal + \epsilon\mathbf{I}\right)^{\frac{1}{2}} + \delta||\mathbf{Z}||_1, \tag{2.36}$$

i.e. $J(\mathbf{Z}_{t+1}) \leq J(\mathbf{Z}_t)$*, where* $\mathbf{Z}_t$ *is the solution of* $\mathbf{Z}$ *in the t-th iteration.*

Proof We first consider the following optimization problem.

$$\begin{aligned}
&\min_{\mathbf{Z}} \tilde{J}(\mathbf{Z})\\
=&\sum_{i=1}^{n}\left(\frac{||\tilde{\mathbf{x}}_i - \tilde{\mathbf{X}}\mathbf{z}_i||^2}{\mathbf{d}_i} + \lambda\delta\mathbf{z}_i^\intercal\mathbf{D}^{-1}\mathbf{z}_i\right) + \lambda\mathbf{tr}\mathbf{Z}^\intercal\mathbf{B}^{-1}\mathbf{Z},
\end{aligned}$$

where $\mathbf{B} = (\mathbf{Z}_t\mathbf{Z}_t^\top + \epsilon\mathbf{I})^{1/2}$, $\mathbf{d}_i = \|\tilde{\mathbf{x}}_i - \tilde{\mathbf{X}}\mathbf{z}_i^t\|, \mathbf{D}_i^{-1} = \mathbf{diag}\left(|Z_{1i}^t|, |Z_{2i}^t|, \cdots, |Z_{ni}^t|\right)$, $\mathbf{Z}_t$ is the solution of Algorithm 1 at iteration t and $\mathbf{z}_i^t$ is the i-th column of $\mathbf{Z}_t$, and Z_{ij}^t is the (i, j) element of $\mathbf{Z}_t$. We will show that (1) the solution at iteration $t + 1$ of Algorithm 1 $\mathbf{Z}_{t+1}$ is a global minimizer of $\tilde{J}(\mathbf{Z})$, *i.e.* $\mathbf{Z}_{t+1} = \arg\min_{\mathbf{Z}} \tilde{J}(\mathbf{Z})$ (thus $\tilde{J}(\mathbf{Z}_{t+1}) \le \tilde{J}(\mathbf{Z}_t)$), and (2) $J(\mathbf{Z}_{t+1}) - J(\mathbf{Z}_t) + \frac{1}{2}\tilde{J}(\mathbf{Z}_t) - \frac{1}{2}\tilde{J}(\mathbf{Z}_{t+1}) \le 0$. Then the proof will be completed by following $J(\mathbf{Z}_{t+1}) \le J(\mathbf{Z}_t) + \frac{1}{2}\left(\tilde{J}(\mathbf{Z}_{t+1}) - \tilde{J}(\mathbf{Z}_t)\right) \le J(\mathbf{Z}_t)$.

(1)One can check that

$$\tilde{J}(\mathbf{Z}) = \sum_{i=1}^{n}\left(\frac{\|\tilde{\mathbf{x}}_i - \tilde{\mathbf{X}}\mathbf{z}_i\|^2}{\mathbf{d}_i} + \lambda\mathbf{z}_i^\top\mathbf{B}^{-1}\mathbf{z}_i + \lambda\delta\mathbf{z}_i^\top\mathbf{D}^{-1}\mathbf{z}_i\right),$$

and

$$\frac{\partial\tilde{J}}{\partial\mathbf{z}_i} = \left(\tilde{\mathbf{X}}^\top\tilde{\mathbf{X}}/\mathbf{d}_i + \lambda\delta\mathbf{D}^{-1} + \lambda\mathbf{B}^{-1}\right)\mathbf{z}_i - \frac{\tilde{\mathbf{X}}^\top\tilde{\mathbf{x}}_i}{\mathbf{d}_i}.$$

Setting the above to be 0, we obtain,

$$\mathbf{z}_i = \left[\tilde{\mathbf{X}}^\top\tilde{\mathbf{X}}/\mathbf{d}_i + \lambda\delta\mathbf{D}^{-1} + \lambda\mathbf{B}^{-1}\right]^{-1}\tilde{\mathbf{X}}^\top\tilde{\mathbf{x}}_i/\mathbf{d}_i.$$

Notice that the above solution is exactly $\mathbf{Z}_{t+1}$. And because $\tilde{J}(\mathbf{Z})$ is convex, we know that $\tilde{J}(\mathbf{Z}_{t+1}) \le \tilde{J}(\mathbf{Z}_t)$.

(2) Let $\mathbf{A} = \left(\mathbf{Z}_{t+1}\mathbf{Z}_{t+1}^{\intercal} + \epsilon\mathbf{I}\right)^{1/2}$, then

$$
\begin{aligned}
& J(\mathbf{Z}_{t+1}) - J(\mathbf{Z}_t) + \frac{1}{2}\tilde{J}(\mathbf{Z}_t) - \frac{1}{2}\tilde{J}(\mathbf{Z}_{t+1}) \\
=& \|\tilde{\mathbf{X}} - \tilde{\mathbf{X}}\mathbf{Z}_{t+1}\|_{2,1} - \|\tilde{\mathbf{X}} - \tilde{\mathbf{X}}\mathbf{Z}_t\|_{2,1} + \lambda\mathbf{tr}(\mathbf{A} - \mathbf{B}) \\
& + \lambda\delta\sum_{ij}\left(|Z_{ij}^{t+1}| - |Z_{ij}^{t}|\right) - \sum_{i=1}^{n}\frac{\|\tilde{\mathbf{x}}_i - \tilde{\mathbf{X}}\mathbf{z}_i^{t+1}\|^2 - \mathbf{d}_i^2}{2\mathbf{d}_i} \\
& -\frac{\lambda}{2}\mathbf{tr}\left(\mathbf{Z}_{t+1}^{\intercal}\mathbf{B}^{-1}\mathbf{Z}_{t+1}^{\intercal} - \mathbf{Z}_t^{\intercal}\mathbf{B}^{-1}\mathbf{Z}_t^{\intercal}\right) \\
& - \frac{\lambda\delta}{2}\sum_{ij}\left(\frac{|Z_{ij}^{t+1}|^2}{|Z_{ij}^{t}|} - |Z_{ij}^{t}|\right) \\
=& \sum_{i=1}^{n}\left(\|\tilde{\mathbf{x}}_i - \tilde{\mathbf{X}}\mathbf{z}_i^{t+1}\| - \mathbf{d}_i - \frac{\|\tilde{\mathbf{x}}_i - \tilde{\mathbf{X}}\mathbf{z}_i^{t+1}\|^2 - \mathbf{d}_i^2}{2\mathbf{d}_i}\right) \\
& + \frac{\lambda\delta}{2}\sum_{ij}\left(2|Z_{ij}^{t+1}| - |Z_{ij}^{t}| - \frac{|Z_{ij}^{t+1}|^2}{|Z_{ij}^{t}|}\right) \\
& + \frac{\lambda}{2}\mathbf{tr}\left(\mathbf{Z}_t^{\intercal}\mathbf{B}^{-1}\mathbf{Z}_t^{\intercal} - \mathbf{Z}_{t+1}^{\intercal}\mathbf{B}^{-1}\mathbf{Z}_{t+1}^{\intercal} + (\mathbf{A} - \mathbf{B})\right)
\end{aligned}
$$

By applying Lemma 4 (with $\mathbf{Z} = \mathbf{Z}_t$ and $\mathbf{Y} = \mathbf{Z}_{t+1}$), we have

$$\begin{aligned}
& J(\mathbf{Z}_{t+1}) - J(\mathbf{Z}_t) + \frac{1}{2}\tilde{J}(\mathbf{Z}_t) - \frac{1}{2}\tilde{J}(\mathbf{Z}_{t+1}) \\
\leq & \sum_{i=1}^{n}\left(\|\tilde{\mathbf{x}}_i - \tilde{\mathbf{X}}\mathbf{z}_i^{t+1}\| - \mathbf{d}_i - \frac{\|\tilde{\mathbf{x}}_i - \tilde{\mathbf{X}}\mathbf{z}_i^{t+1}\|^2 - \mathbf{d}_i^2}{2\mathbf{d}_i} \right) \\
& + \frac{\lambda\delta}{2}\sum_{ij}\left(2|Z_{ij}^{t+1}| - |Z_{ij}^t| - \frac{|Z_{ij}^{t+1}|^2}{|Z_{ij}^t|} \right) \\
= & \sum_{i=1}^{n}\frac{1}{2\mathbf{d}_i}\left(2\mathbf{d}_i\|\tilde{\mathbf{x}}_i - \tilde{\mathbf{X}}\mathbf{z}_i^{t+1}\| - \mathbf{d}_i^2 - \|\tilde{\mathbf{x}}_i - \tilde{\mathbf{X}}\mathbf{z}_i^{t+1}\|^2 \right) \\
& + \frac{\lambda\delta}{2}\sum_{ij}\frac{2|Z_{ij}^t||Z_{ij}^{t+1}| - |Z_{ij}^t|^2 - |Z_{ij}^{t+1}|^2}{|Z_{ij}^t|} \\
= & -\sum_{i=1}^{n}\frac{1}{2\mathbf{d}_i}\left(\|\tilde{\mathbf{x}}_i - \tilde{\mathbf{X}}\mathbf{z}_i^{t+1}\| - \mathbf{d}_i \right)^2 \\
& - \frac{\lambda\delta}{2}\sum_{ij}\frac{\left(|Z_{ij}^t| - |Z_{ij}^{t+1}| \right)^2}{|Z_{ij}^t|} \\
\leq & 0,
\end{aligned}$$

which completes the proof.

Since the objective in Eq.(2.36) is lower bounded by 0, Theorem 2.4.3 guarantees the convergence of the objective value. Further more, we have

And according to Lemma 2.4.1, we know that the above solution is also the optimal solution of Eq.(2.27) when $\epsilon \to 0$.

Now we are ready to proof Lemma 2.2.2.

Proof Lemma 2.2.2

We write the Lagrangian function of Eq. (6) as

$$\mathcal{L}(\mathbf{Z}, \Lambda) = \|\mathbf{Z}\|_* - \mathbf{tr}(\tilde{\mathbf{X}}_1 - \tilde{\mathbf{X}}_1\mathbf{Z})^\intercal\Lambda,$$

where $\tilde{\mathbf{X}}_1$ is defined Eq. (2). We complete the proof by showing that if $\mathbf{Z}$ is given by Eq. (3), there exists a Lagrangian multiplier Λ which satisfies the following,

$$\mathbf{0} \in \partial\mathcal{L}(\mathbf{Z}, \Lambda), \tag{S1.1}$$

where $\partial\mathcal{L}(\mathbf{Z}, \Lambda)$ is the subgradient of $\mathcal{L}(\mathbf{Z}, \Lambda)$. Since Eq. (6) is convex, we then conclude that Eq. (3) gives the optimal solution of Eq. (6).

Now we proof Eq. (S1.1). Assume the SVD decomposition of $\tilde{\mathbf{X}}_1$ is

$$\tilde{\mathbf{X}}_1 = \mathbf{U}\begin{pmatrix} \sigma_1 & \cdots & 0 & \\ \vdots & \ddots & \vdots & \mathbf{0} \\ 0 & \cdots & \sigma_m & \end{pmatrix}[\mathbf{V}, \tilde{\mathbf{V}}]^{\intercal},$$

where $\sigma_1, \cdots, \sigma_m > 0$ and m is the rank of $\tilde{\mathbf{X}}_1$. Notice that $[\mathbf{V}, \tilde{\mathbf{V}}]^{\intercal}[\mathbf{V}, \tilde{\mathbf{V}}] = \mathbf{I}$, then we have,

$$\tilde{\mathbf{X}}_1\tilde{\mathbf{X}}_1^{\intercal} = \mathbf{U}\begin{pmatrix} \sigma_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_m^2 \end{pmatrix}U^{\intercal},$$

or

$$\begin{aligned}\mathbf{Z}^* =& \tilde{\mathbf{X}}_1^{\intercal}(\tilde{\mathbf{X}}_1\tilde{\mathbf{X}}_1^{\intercal})^{-1}\tilde{\mathbf{X}}_1 \\ =& [\mathbf{V}, \tilde{\mathbf{V}}]\begin{pmatrix} 1 & \cdots & 0 & \\ \vdots & \ddots & \vdots & \mathbf{0} \\ 0 & \cdots & 1 & \\ & \mathbf{0} & & \mathbf{0} \end{pmatrix}[\mathbf{V}, \tilde{\mathbf{V}}]^{\intercal}.\end{aligned}$$

From Lemma 2.4.1, we have

$$\lim_{\epsilon\to 0}\left(\mathbf{Z}^*\mathbf{Z}^{*T}+\epsilon\mathbf{I}\right)^{-1/2}\mathbf{Z}^*\in\partial\|\mathbf{Z}^*\|_*,$$

And

$$\begin{aligned}
&\lim_{\epsilon\to 0}\left(\mathbf{Z}^*\mathbf{Z}^{*T}+\epsilon\mathbf{I}\right)^{-1/2}\mathbf{Z}^*\\
&=\lim_{\epsilon\to 0}\left([\mathbf{V},\tilde{\mathbf{V}}]\begin{pmatrix}1&\cdots&0&\\\vdots&\ddots&\vdots&\mathbf{0}\\0&\cdots&1&\\&\mathbf{0}&&\mathbf{0}\end{pmatrix}[\mathbf{V},\tilde{\mathbf{V}}]^\intercal+\epsilon\mathbf{I}\right)^{-1/2}\mathbf{Z}^*\\
&=\lim_{\epsilon\to 0}\left([\mathbf{V},\tilde{\mathbf{V}}]\begin{pmatrix}1+\epsilon&\cdots&0&\\\vdots&\ddots&\vdots&\mathbf{0}\\0&\cdots&1+\epsilon&\\&\mathbf{0}&&\epsilon\mathbf{I}\end{pmatrix}[\mathbf{V},\tilde{\mathbf{V}}]^\intercal\right)^{-1/2}\mathbf{Z}^*\\
&=\lim_{\epsilon\to 0}[\mathbf{V},\tilde{\mathbf{V}}]\begin{pmatrix}\frac{1}{\sqrt{1+\epsilon}}&\cdots&0&\\\vdots&\ddots&\vdots&\mathbf{0}\\0&\cdots&\frac{1}{\sqrt{1+\epsilon}}&\\&\mathbf{0}&&\frac{1}{\sqrt{\epsilon}}\mathbf{I}\end{pmatrix}[\mathbf{V},\tilde{\mathbf{V}}]^\intercal\mathbf{Z}^*\\
&=\lim_{\epsilon\to 0}\mathbf{V}\begin{pmatrix}\frac{1}{\sqrt{1+\epsilon}}&\cdots&0\\\vdots&\ddots&\vdots\\0&\cdots&\frac{1}{\sqrt{1+\epsilon}}\end{pmatrix}\mathbf{V}^\intercal\\
&=\mathbf{V}\mathbf{V}^\intercal,
\end{aligned}$$

which leads to

$$\mathbf{V}\mathbf{V}^\intercal-\frac{\partial\mathbf{tr}(\tilde{\mathbf{X}}_1-\tilde{\mathbf{X}}_1\mathbf{Z})^\intercal\Lambda}{\mathbf{Z}}\in\partial\mathcal{L}(\mathbf{Z},\Lambda)|_{\mathbf{Z}=\mathbf{Z}^*},$$

or

$$\mathbf{V}\mathbf{V}^{\intercal} - \tilde{\mathbf{X}}_1^{\intercal}\Lambda \in \partial\mathcal{L}(\mathbf{Z}, \Lambda)|_{\mathbf{Z}=\mathbf{Z}^*},$$

To complete the proof, one only need to find Λ such that $\mathbf{V}\mathbf{V}^{\intercal} - \tilde{\mathbf{X}}_1^{\intercal}\Lambda = \mathbf{0}$, or equivalently,

$$\mathbf{V}\mathbf{V}^{\intercal} - \mathbf{V}\Sigma\mathbf{U}^{\intercal}\Lambda = \mathbf{0},$$

where

$$\Sigma = \begin{pmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_m \end{pmatrix}$$

We let

$$\Lambda = \mathbf{U}\Sigma^{-1}\mathbf{V}^{\intercal}.$$

This leads to

$$\begin{aligned} &\mathbf{V}\mathbf{V}^{\intercal} - \mathbf{V}\Sigma\mathbf{U}^{\intercal}\Lambda \\ =&\mathbf{V}\mathbf{V}^{\intercal} - \mathbf{V}\Sigma\mathbf{U}^{\intercal}\mathbf{U}\Sigma^{-1}\mathbf{V}^{\intercal} \\ =&\mathbf{0}, \end{aligned}$$

which completes the proof.

2.5 Applications

2.5.1 Using Multi-Subspace Representation as Preprocessing

Since **Z** is low rank, **XZ** is also low rank. And since **Z** is sparse, **XZ** can be interpreted as a sparse coding representation of **X**. According to the analysis in §4.2, we hopefully improve the qualities of the data representation by using **XZ**. In our study, we replace $\mathbf{X}$ by $\mathbf{XZ}$ as a preprocessing step for various machine learning problems, where $\mathbf{Z}$ is the optimal solution of Eq. (2.27).

Notice that the learning of $\mathbf{Z}$ in Eq. (2.27) is unsupervised, which requires no further label information. Thus we can apply it as preprocessing for any machine learning tasks, as long as the data are represented in Euclidean space. In this chapter, we employ MSR for clustering, semi-supervised learning, and classification. We will demonstrate the performance of the preprocessing in the experimental section.

2.5.2 Using Multi-Subspace Representation as Classifier

Here we try to directly make use of our MSR model as a standalone classifier. Assume we have n data points in the data set, $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]$ and the first m data points have discrete class labels $y_1, y_2, \cdots, y_m$ in K classes, $y_i \in \{1, 2, \cdots, K\}$. The classification problem is to determine the class label of $\mathbf{x}_i, i = m+1, \cdots, n$. Let $\mathbf{Z}$ be the optimal solution of Eq.(2.27) for n data points. The MSR representation of each image is $\mathbf{X}\mathbf{z}_i,\ i = 1, \cdots, n$. The class prediction of our model for unlabeled data $\mathbf{x}_t,\ t = m + 1, \cdots, n$, is

$$\arg\min_k r_k = \|\mathbf{X}\mathbf{z}_t - \hat{\mathbf{x}}_t^k\|,\ \hat{\mathbf{x}}_t^k = \sum_{i \in C_k} \mathbf{x}_i Z_{it}. \tag{2.38}$$

Here $\hat{\mathbf{x}}_t^k$ is the representation of testing object $\mathbf{x}_t$ using objects in class C_k, $k = 1, 2, \cdots, K$.

The classification strategy is similar with Wright *et al*'s approach [6]. We will compare the two models in the experimental section.

2.6 Experiment

2.6.1 A Toy Example

We demonstrate with toy example of the affine space recovering by our method in Figure 2.2 [8] (a) shows 100 images from 10 groups used in this example, which are se-

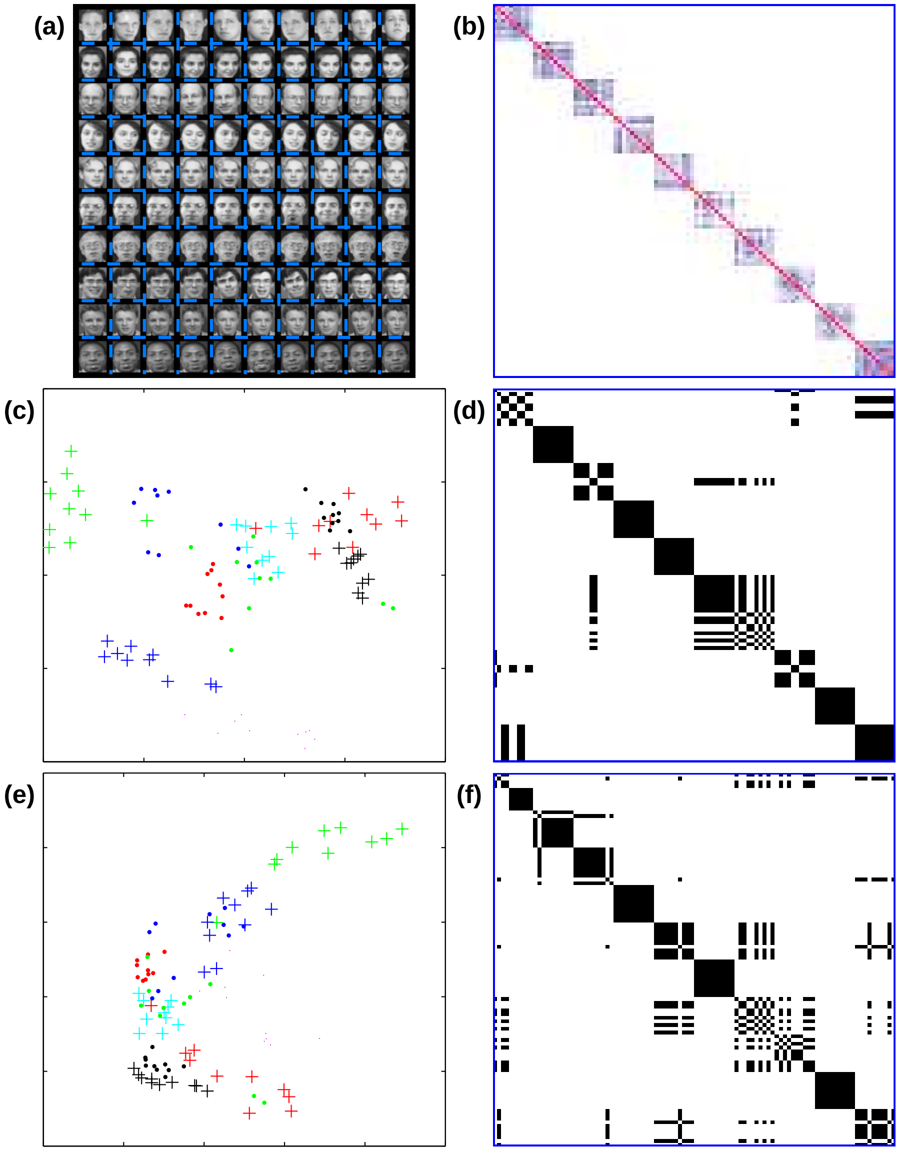


Figure 2.2. A toy example of multi-subspace discovery problem and our solution. (a): 100 images in which the last component has been removed within each group, then the conditions in Proposition 2.2.5 are satisfied. (b): the optimal solution of $\mathbf{Z}$ in Eq. (2.13). White, blue, and red colors represent zeros, negative values, and red positive values, respectively. Within each group, the values of the subgraph represented by $\mathbf{Z}_k$ (defined in Eq. (2.8)) is a single connected component and among the 10 blocks they are disconnected components. (c): PCA visualization of the 100 images where x-axis and y-axis are the first and second principal component, respectively. (d): clustering grouping of K-means. (e): Laplacian Embedding results of the 100 images where x-axis and y-axis are the eigenvectors with the second and third least eigenvectors of graph Laplacian matrix, respectively. (f): clustering grouping of normalized-cut. .

lected from the AT&T data set, details can be found in the experimental section. In order to obtain 10 affine subspaces which satisfy the constraints in Proposition 2.2.5, we remove the last principle component in each group of face images. To be more specific, for each group $\mathbf{X}_k$, we first subtract the data points by the group mean $\mathbf{m}_k : \bar{\mathbf{X}}_k = \mathbf{X}_k - \mathbf{m}_k \mathbf{1}^\intercal$, then perform a PCA (Principle Component Analysis) on the zero-mean data and keep the first 8 principle components and get rid of the 9-th principle component. Then the data is projected back on to the original space and the mean $\mathbf{m}_k$ is added back. Assume the resulting PCA projection is $\mathbf{U}_k$ then the processed data $\mathbf{Y} = \mathbf{U}_k \mathbf{U}_k^\intercal \bar{\mathbf{X}}_k + \mathbf{m}_k$ are used in our example, $k = 1, 2, \cdots, 10$. The images in which the last principle component have been removed are shown in Figure 2.2 (a). Notice that they are visually almost identical to the original image since the energy of the last component is close to zero. Then we solve Eq. (2.13) and the optimal solution is shown in Figure 2.2 (b), in which white color represents zeros, blue colors represent negative values, and red positive values. One can see that within each group, the values of the subgraph represented by $\mathbf{Z}_k$ (defined in Eq. (2.8)) is a single connected component and among the ten $\mathbf{Z}_k, k = 1, 2, \cdots, 10$ they are disconnected components.

For comparison, we also cluster the data using K-means and normalized-cut [21], see Figure 2.2 (d) and (f), the corresponding Principal Component Analysis and Spectral embedding results are also shown in (c) and (e), respectively. One see that both K-means and Normalized-Cut cannot correctly discover the subspaces and group assignments.

2.6.2 Experimental Settings

Datasets. We evaluate the performance of our model on 5 real world datasets, including two face image data bases, *LFW* (Labeled Faces in the Wild)[5], *AT&T*[6], two UCI datasets

[5] http://www.itee.uq.edu.au/~conrad/lfwcrop/

[6] http://people.cs.uchicago.edu/˜dinoj/vis/ORL.zip

Table 2.1. Data descriptions of experiments

Data	Type	sample #	feature #	class #	max #	min #
LFW	Face Image	400	4096	20	20	20
AT&T	Face Image	400	644	40	10	10
Australian	Financial	690	14	2	383	307
BinAlpha	Text Image	1404	320	36	39	39
Dermatology	Disease	366	34	6	112	20

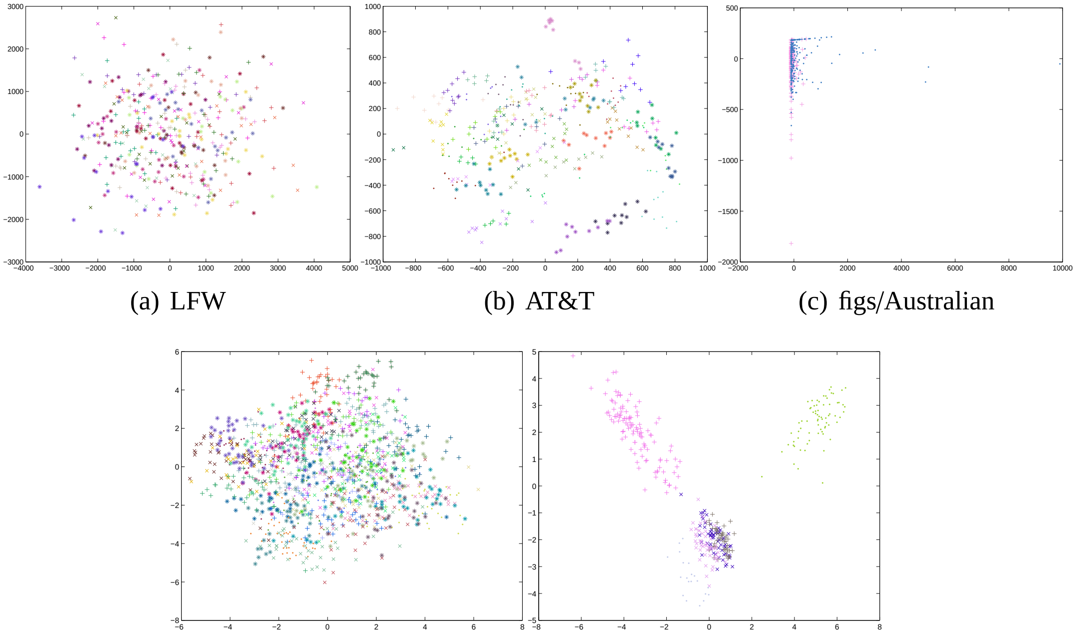
Austrian and *Dermatology* [22], and one handwritten character data BinAlpha[7]. All the data sets are used with the original data, without any further preprocessing.

We summarize the data statistics for these data sets in Table 4.2, where the number of samples, features, classes are listed. The minimum and maximum number samples in classes are also listed to show the balance of the data. We also visualize the data using the first and second principal components for these data in Figure 2.3.

Compared Methods. For the usage of preprocessing of our model, we compare 3 clustering algorithms (Normalized Cut [21] which tends to produce balanced clustering results on manifolds, Spectral Embedding Clustering [23], and K-means, which is the standard clustering algorithm), two standard semi-supervised learning algorithms (Local and Global Consistency by [24], which considers the local and global consistency of data points and Gaussian Fields and Harmonic Functions by [25], which formulates Gaussian random field graphs by harmonic functions using matrix methods or belief propagation), and two standard classification algorithms (linear Support Vector Machine, which is a stable and competitive classification method for high dimensional data, and k-Nearest Neighbor).

For the usage of standalone classifier, we compare our method with Wright *et. al*'s sparse representation based approach [6].

[7] http://www.cs.toronto.edu/˜roweis/data.html



(a) LFW (b) AT&T (c) figs/Australian

(d) BinAlpha (e) Dermatology

Figure 2.3. Data visualization with the first (x-axis) and second (y-axis) principal components..

Validation Settings. All the clustering algorithms compared in our experiments require random initializations. Thus we run the algorithms for 50 random trials and report the averages. For semi-supervised learning, we randomly split the data into 30% and 70% where the 30% of the data points are used as labeled data and 70% are used as unlabeled data. We repeat the random splitting for 50 times, where the average result is reported. For classification, when comparing our method as a preprocessing algorithm, we use the same splitting strategy as in semi-supervised learning, but splitting in to 50% for training and the other half for testing. For classification, when comparing our method as a standalone classifier, we use 30% for training and the rest 70% for testing. The reason is that for some of the datasets, the data points are well separated and the classification accuracy is very high, then the difference between approaches is not obvious. Thus here we use fewer data samples as the training set to enlarge the differences.

Parameter settings. K-means has no parameters. For kNN we use $k = 1$, *i.e.* just use the nearest neighbor classifier. For the Normalized Cut (NCut), Spectral Embedding Clustering (SEC) in clustering, Local and Global Constancy (LGC), and Gaussian Fields and Harmonic Functions (GFHF) in semi-supervised learning, we establish the graph using Gaussian kernel: $W_{ij} = \exp\left(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2/\sigma^2\right)$, where γ is the parameter which is set to be $\gamma = [0.1, 0.5, 1, 2, \cdots, 30]$ and σ is the average of pairwise Euclidian distances among all data points.

For Wright *et. al*'s sparse representation (SR), we use LARS [26] to obtain the full LASSO path solution and use m top ranked coefficients according to the shrinking order in LARS solution path. We choose m from $m = 1, 2, \cdots \min(n, p)$ where n is the number of data points and p is the number of data dimension. The reason we use LARS is that it is more efficient than any other ℓ_1 solver in the sense that LARS computes all the possible solution with different parameters at once and for other solver, we need to retrain the model every time we change the parameter, which is time consuming for the purpose of highly parameter tuning. For our method, we choose λ from $[0.5, 0.6, \cdots, 2.5]$.

2.6.3 Experimental Results

For the usage of preprocessing our model, the results are shown in Figure 3.4. Here we show the average accuracies for both original data without processing (marked as *Orig* in the figure) and the corresponding method on the preprocessed data by our method (marked as *MSR*). We further plot the original accuracy values of all the 50 random trials for each methods to visualize the overall differences of the performance.

One-way ANOVA (Analysis of Variance) is performed to test how significantly our method is better than the original method, and corresponding p value is also shown in the figure. $p \leq \epsilon$ means p is less than any positive values in machine precision, *i.e.* the p value is very close to 0.

Table 2.2. Clustering accuracy comparison of our method as a preprocessing method. *Orig* denotes without any processing, *PCA* denotes the clustering accuracy with PCA dimensional reduction, and *MSR* denotes the clustering accuracy with our method. Best results are highlighted.

	Ncut			SEC			*K*-means		
	Orig	PCA	MSR	Orig	PCA	MSR	Orig	PCA	MSR
LFW	0.194	0.193	**0.213**	0.207	0.225	**0.245**	0.193	0.177	**0.198**
AT&T	0.797	0.795	**0.822**	0.805	0.791	**0.810**	0.599	0.583	**0.621**
Australian	0.557	0.564	**0.667**	0.662	0.669	**0.691**	0.562	0.546	**0.665**
BinAlpha	0.357	0.346	**0.388**	0.468	0.486	**0.487**	0.412	0.417	**0.431**
Dermatology	0.829	0.821	**0.891**	0.869	0.874	**0.958**	0.759	0.761	**0.805**

Table 2.3. Semi-supervised learning accuracy comparison of our method as a preprocessing method. *Orig* denotes the accuracy without any processing, *PCA* denotes the accuracy with PCA dimensional reduction, and *MSR* denotes the accuracy with our method. Best results are highlighted.

	GFHF			LGC		
	Orig	PCA	MSR	Orig	PCA	MSR
LFW	0.1636	0.1688	**0.2185**	0.2227	0.2154	**0.2700**
AT&T	0.3458	0.3379	**0.6682**	0.7881	0.7701	**0.8195**
Australian	0.5549	0.5590	**0.6736**	0.5598	0.5487	**0.6778**
BinAlpha	0.5670	0.5813	**0.5968**	0.6198	0.6299	**0.6529**
Dermatology	0.7543	0.7448	**0.8673**	0.9226	0.9293	**0.9446**

Out of the $5 \times 7 = 35$ comparisons, our method significantly outperforms the original methods in 33 comparisons, with $p \leq 0.03$. There is one case (SVM on *AT&T* data set) where our method is better but with no significant evidence. There is also another case in which our method is worse than the original method (kNN on *AT&T*), but the difference is not significant ($p = 0.263$).

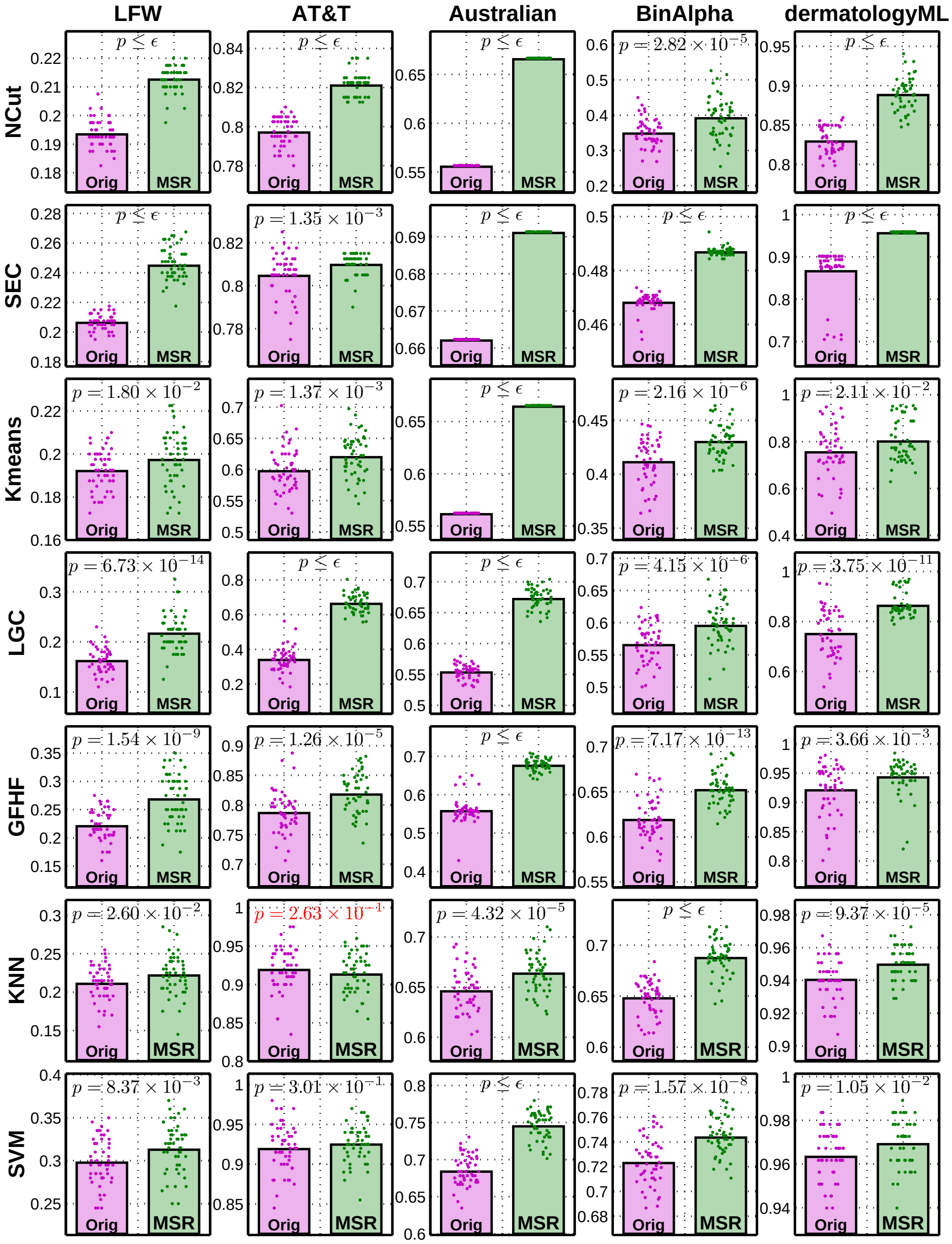


Figure 2.4. Experimental results of our method as a preprocessing method on 7 learning methods and 5 data sets. The scattering dots represent the accuracy values of the methods and bars represent the averages. *Orig* and *MSR* denote the corresponding method on the original data and on the preprocessed by our method, respectively. The p stands for the significance of the one-way ANOVA test (for the hypothesis of "our method is better than the original method"). Out of 35 comparison, our method significantly outperforms the original methods in 33 cases, with $p \leq 0.03$. ϵ is the smallest positive values by machine precision..

Table 2.4. Classification accuracy comparison of our method as a preprocessing method. *Orig* denotes the accuracy without any processing, *PCA* denotes the accuracy with PCA dimensional reduction, and *MSR* denotes the accuracy with our method. Best results are highlighted.

	KNN			SVM		
	Orig	PCA	MSR	Orig	PCA	MSR
LFW	0.2122	0.2316	**0.2231**	0.2990	0.3053	**0.3140**
AT&T	0.9203	0.9093	**0.9143**	0.9204	0.9371	**0.9260**
Australian	0.6469	0.6560	**0.6645**	0.6853	0.6853	**0.7463**
BinAlpha	0.6488	0.6304	**0.6883**	0.7238	0.7339	**0.7444**
Dermatology	0.9410	0.9441	**0.9503**	0.9638	0.9822	**0.9696**

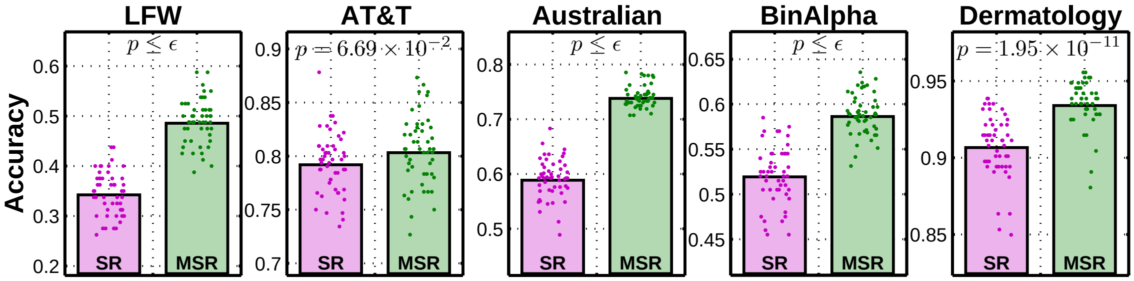


Figure 2.5. A comparison of our model (*MSR*) and the Sparse Representation based method (*SR*) on 5 data sets. The p values represents the significance of one-way ANOVA test of the hypothesis “our method is better than SR”. .

For our model as a standalone classifier, the comparison results with Sparse Representation based method are shown in Figure 2.5 [8]. Out of 5 data sets, our method is significantly better than the Sparse Representation based method in four with $p \leq 0.01$.

Due to the low rank property of our method, one might also be interested in comparing our method with other low rank method as preprocessing. We compare the preprocessing results with Principal Component Analysis (PCA) in Table 2.2 – 2.4[8]. For PCA, the best numbers of dimensions are achieved by tuning and the best results are reported.

[8] These results have been published in paper [8].

Table 2.5. Running time (s) of our algorithm

Dataset	LFW	AT&T	Australian	BinAlpha	Dermatology
Time	0.549 ± 0.061	0.437± 0.114	1.768 ± 0.092	4.447± 0.367	0.393 ± 0.058

2.6.4 Running Time

We test the running time on a Intel Core i7-2670QM CPU @ 2.20 GHz with 8 GB memory and 64-bit operating system. The codes are implemented with Matlab. In the third line of the *for* loop in Algorithm 1, we use the Matlab command “x = A \y” to solve “x = inv(A) * y”. The running time of our algorithm is listed in Table 2.5. The average and standard deviation over 10 times of random trials are reported.

2.6.5 Experimental Results on TCGA Data

The Cancer Genome Atlas (TCGA) project has analyzed mRNA expression, miRNA expression, promoter methylation, and DNA copy number in 489 high-grade serous ovarian adenocarcinomas (HGS-OvCa) and the DNA sequences of exons from coding genes in 316 of these tumors. These results show that HGS-OvCa is characterized by TP53 mutations in almost all tumors (96%); low prevalence but statistically recurrent somatic mutations in 9 additional genes including NF1, BRCA1, BRCA2, RB1, and CDK12; 113 significant focal DNA copy number aberrations; and promoter methylation events involving 168 genes. Analyses delineated four ovarian cancer transcriptional subtypes, three miRNA subtypes, four promoter methylation subtypes, a transcriptional signature associated with survival duration and shed new light on the impact on survival of tumors with BRCA1/2 and CCNE1 aberrations. Pathway analysis suggested that homologous recombination is defective in about half of tumors, and that Notch and FOXM1 signaling are involved in serous ovarian cancer pathophysiology.

Table 2.6. Running time (s) of our algorithm

Dimensions	mRNA	Copy Number	Methylation	miRNA
# of measurements	17814	21942	25149	799
min value	-9.3743	-5.6338	0.0000	-7.1434
max value	10.9291	5.1026	1.0000	9.9827
# patients		455		

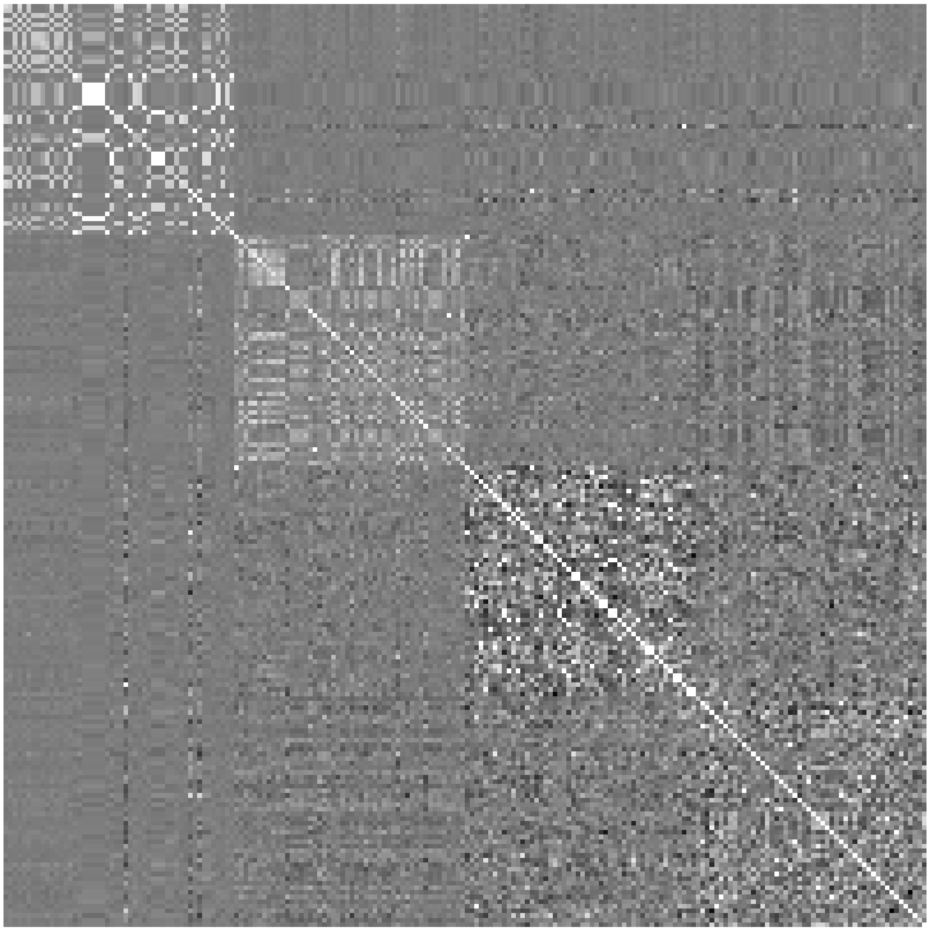


Figure 2.6. Experimental results of **Z** on TCGA data. .

We downloaded the clinical data from website[9] [27] for mRNA expression, DNA copy number variation, promoter methylation, and miRNA expression.

Among these dimensions, we have selected 455 patients which have measurements in all 4 dimensions and have survival records as well. We list the number of measurement of each dimensions as well as the value ranges of the data in Table 2.6.

We selected 50 elements in each view of the data and put them together to form $\mathbf{X}$ and solve the multi-space learning problem in Eq. (2.27) and the optimal solution of $\mathbf{Z}$ is shown in Figure 2.6.

The observation here is that the representative coefficient of $\mathbf{Z}$ between *has-mir-200c* and *PI3K* is significantly high.

[9] https://tcga-data.nci.nih.gov/tcga/dataAccessMatrix.htm?diseaseType=OV

CHAPTER 3

STRUCTURED LEARNING WITH EXPLICIT ℓ_2/ℓ_0-NORM

3.1 Introduction of Structured Learning

Both theoretical and empirical studies have suggested that the sparsity is one of the intrinsic properties of real world data [28, 26, 29, 30]. Sparse representation not only simplifies the data models, but also helps us in discovering predictive patterns in data which enhance our interpretation and understanding of underlying physical, biological and other natural mechanisms [3, 31, 32, 33, 34].

Sparse representations are typically achieved by imposing non-smooth norms, *e.g.* ℓ_1 norm and ℓ_2/ℓ_1-norm (initially called rotational invariant ℓ_1 norm or R_1-norm [35]), as penalties/regularizers in the optimization problems. Applications include LASSO [11], compressive sensing [36, 37], matrix factorization [38], multi-task learning [39]. Related approaches are also successfully developed and applied into other scientific domains, such as genetics analysis [40, 41], neuroscience [42], computer vision [43, 6], and disease studies [44] *etc.*

The optimization problems of these approaches often consist of two components: a convex smooth loss function and a convex non-smooth regularizer. Although the global solutions are guaranteed, the naive approaches are often inefficient and unsuitable for large scale problems. Thus, more efficient algorithms are desired. According to the structure of the constraints, the sparsity can be obtained from three types of regularizers for different purposes:

1. *Flat sparsity*. This type of sparsity is often achieved by ℓ_1-norm regularizer. Optimization techniques include LARS [26], linear gradient search [45], and proximal methods [46].
2. *Structural sparsity*, including group features/covariates detection [47, 48, 49], jointly vector sparsity [50], hierarchical group features [51], *etc.* In the other communities, the structural sparsity is also called block sparsity [52]. The sparsity is often obtained by ℓ_2/ℓ_1-norms, which can be efficiently solved by methods in [53] and [54].
3. *Matrix/tensor sparsity*, such as matrix/tensor completion [55, 43]. The typical regularizer is the trace norm which can be solved by Singular Value Decomposition thresholding [55].

In this chapter[1], we focus on the structural sparsity. For the structural sparsity purpose, we often deal with convex optimization problems (with convex non-smooth norm, like ℓ_2/ℓ_1 norms) and a large number of optimization techniques have developed to tackle the problems, for example [48, 51, 26, 56], *etc*.

3.2 An Illustration of Structural Sparsity

Here we provide a concrete example to illustrate the subtle difference between *structural* sparsity and *flat* sparsity, which show why structural sparsity is useful in machine learning and data mining.

[1] Most of the major results in this chapter have been published in paper [5].

3.2.1 LASSO

Let $\mathbf{X}_{p\times n} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]$ be n data point ($\mathbf{x}_i$ is a p-dimensional vector), $\mathbf{y}_{n\times 1} = (y_1.\cdots, y_n)^T$ be the class labels, and $\boldsymbol{\beta}_{p\times 1} = [\beta_1, \beta_2, \cdots, \beta_p]^T$ be the p-dimensional vector of regression coefficients. Consider the following class prediction problem,

$$\min_{\beta} J(\boldsymbol{\beta}) = \frac{1}{2}||\mathbf{y} - \mathbf{X}^T\boldsymbol{\beta}||^2 + \lambda||\boldsymbol{\beta}||_1, \tag{3.1}$$

where λ is model parameter which controls the sparsity of $\boldsymbol{\alpha}$. This model is known as LASSO [11]. The solution is typically sparse, *i.e.*, the solution of β contains many zero elements.

Assume in the optimal solution of Eq. (3.1) for some $j_0, \beta_{j_0} = 0$.

$$(\mathbf{X}^T\boldsymbol{\beta})_i = \sum_{j=1}^{p} \mathbf{x}_i^j \beta_j = \sum_{1\le j\le p, j\ne j_0} \mathbf{x}_i^j \beta_j, \tag{3.2}$$

where $\mathbf{x}_i^j$ is the j-the component of $\mathbf{x}_i$. Eq. (3.2) indicates the j_0-th component/dimension/feature of all x_i are irrelevant, because they multiply zeros in actual usage. For larger λ, more elements of β are zero, indicating more features/dimensions are eliminated. The remaining features are thus *selected*. The sparse learning is useful for feature selection.

3.2.2 Multi-Task Regression: A Structural Sparsity Example

Now let us consider K linear regression simultaneously, with same data $\mathbf{X}$ but different regression target $\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_K$. Denote $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_K]$, then a naive extension of Eq. (3.1) is

$$\begin{aligned} \min_{\mathbf{B}} J_1(\mathbf{B}) &= \sum_{k=1}^{K} \left(\frac{1}{2}||\mathbf{y}_k - \mathbf{X}^T\boldsymbol{\beta}_k||^2 + \lambda||\boldsymbol{\beta}_k||_1 \right) \\ &= \frac{1}{2}||\mathbf{Y} - \mathbf{X}^T\mathbf{B}||_F^2 + \lambda||\mathbf{B}||_1, \end{aligned} \tag{3.3}$$

where $\mathbf{B} = [\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \cdots, \boldsymbol{\beta}_K]$. Since Eq. (3.3) solve K linear regression problems simultaneously but independently, the sparsity pattern of the elements in $\mathbf{B}$ is not structured. We use a synthetic data to demonstrate what the word "structured" means here. Let

$$\mathbf{X}^T = \begin{pmatrix} 0.463 & 0.319 & -0.100 & 0.526 & 0.535 & 0.329 & 0.475 \\ 0.296 & 0.192 & 0.058 & -0.076 & 0.152 & 0.313 & -0.114 \\ 0.196 & 0.189 & 0.167 & -0.280 & 0.267 & -0.246 & 0.164 \\ 0.330 & 0.357 & 0.027 & -0.001 & 0.118 & 0.058 & 0.191 \\ 0.332 & 0.035 & -0.002 & 0.280 & 0.111 & -0.043 & 0.104 \\ -0.022 & -0.026 & 0.770 & 0.189 & 0.196 & -0.146 & -0.121 \\ -0.217 & 0.028 & 0.404 & 0.359 & 0.335 & -0.282 & -0.235 \\ 0.396 & 0.297 & 0.260 & 0.241 & 0.193 & 0.038 & 0.101 \end{pmatrix}, \mathbf{Y} = \begin{pmatrix} 1 & \mathbf{0} & \mathbf{0} \\ 1 & 1 & \mathbf{0} \\ 1 & \mathbf{0} & 1 \\ 1 & 1 & 1 \\ \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{0} & 1 & 1 \\ \mathbf{0} & \mathbf{0} & 1 \\ \mathbf{0} & \mathbf{0} & 1 \end{pmatrix} \tag{3.4}$$

By solving Eq. (3.3) with $\lambda = 0.3$, we obtain the following global optimal solution,

$$\mathbf{B}_1^* = \begin{pmatrix} 0.350 & 1.262 & \mathbf{0.000} \\ 1.128 & \mathbf{0.000} & 1.866 \\ \mathbf{0.000} & 0.701 & 1.205 \\ -0.749 & \mathbf{0.000} & \mathbf{0.000} \\ 1.156 & \mathbf{0.000} & \mathbf{0.000} \\ 0.151 & \mathbf{0.000} & -0.993 \\ \mathbf{0.000} & -0.450 & \mathbf{0.000} \end{pmatrix}$$

Notice that the solution is sparse, *i.e.* there many zero elements in the solution. However these sparsity patterns are *inconsistent*: For class C_1 label prediction, feature dimensions (3,7) are irrelevant. For class C_2 label prediction, feature dimensions (2,4,5,6) are irrelevant. For class C_3 label prediction, feature dimensions (1,4,5,7) is irrelevant. Therefore, this type of inconsistent feature elimination are not useful. This inconsistent sparsity pattern for different classes are called *flat sparsity*.

Now we consider a *structural sparsity*. In the solution $\mathbf{B}$, if the entire j_0-th row is zero, *i.e.* $B_{j_0,k} = 0, k = 1, 2, 3$. Then

$$\left(\mathbf{X}^T\mathbf{B}\right)_{ik} = \sum_{j=1}^{p} \mathbf{x}_i^j B_{jk} = \sum_{j=1, j\neq j_0}^{p} \mathbf{x}_i^j B_{jk},$$

suggesting that the j_0's component of $\mathbf{x}_i$ is irrelevant in the regression output, *i.e.*

$$\boxed{\text{The } j_0\text{-th row of } \mathbf{B} \text{ is zero} \rightarrow \text{The } j_0\text{-th feature dimension of } \mathbf{X} \text{ is irrelevent.}}$$

There could be several rows $(j_0, j_1, \cdots, j_\sigma)$ of B where the entire row are zeroes. These *consistent* sparse patterns are useful, because feature dimensions are consistently eliminated for all K class label predictions. Thus $\mathbf{B}$ is an indicator to select relevant features.

How do we get structural sparsity? We solve the following problem,

$$\min_{\mathbf{B}} J_{21}(\mathbf{B}) = \frac{1}{2}||\mathbf{Y} - \mathbf{X}^T\mathbf{B}||_F^2 + \lambda \sum_i \sqrt{\sum_j B_{ij}^2} \tag{3.5a}$$

$$= \frac{1}{2}||\mathbf{Y} - \mathbf{X}^T\mathbf{B}||_F^2 + \lambda \sum_i ||\mathbf{b}^i||, \tag{3.5b}$$

where $\mathbf{b}^i$ is the i-th row of $\mathbf{B}$. The term $\sum_i \sqrt{\sum_j B_{ij}^2}$ in Eq. (3.5a) is called the ℓ_2/ℓ_1 norm of matrix $\mathbf{B}$. With its equivalent notation in Eq. (3.5b), the ℓ_2/ℓ_1 norm of $\mathbf{B}$ can be interpreted as the ℓ_1 norm of ℓ_2 norm of its rows, which generates the sparsity on the rows. With the same $\mathbf{X}$ and $\mathbf{Y}$ in Eq. (3.4) and with $\lambda = 0.3$ and $\lambda = 0.5$ we obtain the following global optimal results of Eq. (3.5b),

$$\mathbf{B}^*_{21}|_{\lambda=0.3} = \begin{pmatrix} 0.526 & 0.764 & 0.183 \\ 1.101 & 0.198 & 1.101 \\ -0.027 & 0.859 & 1.253 \\ -0.139 & -0.016 & -0.040 \\ 0.441 & -0.090 & 0.215 \\ 0.144 & 0.188 & -0.490 \\ \mathbf{0.000} & \mathbf{0.000} & \mathbf{0.000} \end{pmatrix}, \mathbf{B}^*_{21}|_{\lambda=0.5} = \begin{pmatrix} 0.677 & 0.660 & 0.254 \\ 0.678 & 0.187 & 0.584 \\ 0.002 & 0.682 & 1.251 \\ \mathbf{0.000} & \mathbf{0.000} & \mathbf{0.000} \\ 0.231 & \mathbf{0.000} & 0.157 \\ \mathbf{0.000} & \mathbf{0.000} & \mathbf{0.000} \\ \mathbf{0.000} & \mathbf{0.000} & \mathbf{0.000} \end{pmatrix}.$$

We can see that the solutions of $\mathbf{B}^*_{21}|_{\lambda=0.3}$ and $\mathbf{B}^*_{21}|_{\lambda=0.5}$ are row-wise sparse, and $\mathbf{B}^*_{21}|_{\lambda=0.5}$ is more sparse than $\mathbf{B}^*_{21}|_{\lambda=0.3}$. With these structural results, we can selects relevant features in the multi-task regression.

3.2.3 Group LASSO

We can also specify the structures by groups. In *group lasso,* we are interested in solving the following problem,

$$\min_{\boldsymbol{\beta}} J(\boldsymbol{\beta}) = ||\mathbf{y} - \sum_{g} \sum_{i \in g} \mathbf{x}^i \beta_i||^2 + \lambda ||\boldsymbol{\beta}^g||, \tag{3.6}$$

where $\mathbf{X} = [\mathbf{x}^1, \mathbf{x}^2, \cdots, \mathbf{x}^p]$. In (3.20), we assume that the features of a data point $\mathbf{x}_i$ ordered in groups,

$$\mathbf{x}_i^T = [\overbrace{\mathbf{x}_i^1, \mathbf{x}_i^2, \cdots, \mathbf{x}_i^{|g_1|}}^{g_1}, \overbrace{\mathbf{x}_i^{|g_1|+1}, \cdots, \mathbf{x}_i^{|g_1|+|g_2|}}^{g_2}, \cdots, \overbrace{\cdots, \mathbf{x}_i^p}^{g_K}], \tag{3.7}$$

where $g_1, g_2, \cdots, g_K$ are K groups of indexes ranging in $[1, 2, \cdots, p]$ and $|g_j|$ is the size of group g_j, $j = 1, 2, \cdots, K$.

We continue to use the same **X** and the first column of **Y** as **y** and the following grouping to specify a group LASSO problem in (3.20).

$$\overbrace{1,2,3,}^{g_1}\ \overbrace{4,5}^{g_2}\ ,\ \overbrace{6,7}^{g_3}$$

and the corresponding results of group lasso with $\lambda = 1$ is

$$\boldsymbol{\beta}^T = [\overbrace{0,0,0,}^{g_1}\overbrace{0.17,0.14,}^{g_2}\ \overbrace{0,0}^{g_3}].$$

The structural sparsity of group lasso is similar to ℓ_2/ℓ_1 norm of multi-task regression discussed in previous subsection, *i.e.* for some of the groups (g_1 and g_3 in this specific case), the entire group are zeros.

3.2.4 $\boldsymbol{\ell}_2/\boldsymbol{\ell}_0$-norm: An Explicit Approach

However, the purpose of the convex norms is to approximate the cardinality. In feature selection problems, the feature we are interested in is a subset of the whole feature space. For this purpose, the most natural constraint is the cardinality constraint.

To directly solve this problem, we propose the explicit ℓ_2/ℓ_0 regularizer in this chapter. For a matrix $\mathbf{A} = (\mathbf{A}_{ij})$, the ℓ_2/ℓ_0 norm is defined as $||\mathbf{A}||_{\ell_2/\ell_0} = \sum_i || \sum_j \mathbf{A}_{ij}^2||_0$, where for a scalar x, $||x||_0 = 1$ if $x \neq 0$, $||x||_0 = 0$ if $x = 0$. For a vector $\mathbf{x}$, the group ℓ_2/ℓ_0 norm is $\sum_g ||||\mathbf{x}^g||||_0$.

Due to the difficulty of ℓ_2/ℓ_0 norm, instead of using convex norm for approximation, we develop a novel general optimization framework to solve the induced problems by introducing an auxiliary function. The major advantage of our auxiliary function method is that it induces an extremely simple optimization problem which can be decoupled as a sum of loss for grouped variables.

3.2.5 Towards Deeper Understanding by Structural Sparsity

One of the basic task in machine learning is to establish *accurate* classifiers, which can be used to predict some unknown knowledge. The accuracy is often estimated by cross-validation, of some other empirical studies. However, such predictors are, in most cases, *black boxes*, *i.e.* the only output is the classification or regression result, nothing else is obtained (in an interpretable way). For example, in an typical machine learning study, we might use the functional MR images or image sequences as input and try to classify whether the person is looking at an image or reading a sentence. One might try to develop more advance techniques to push the accuracy. However, what is the next step? Without the investigation of the hidden mechanisms of the brain functions, few could be done in an meaningful way. By *hidden mechanisms* we mean explore a *white box* understanding of the object we are interested. For example, we might be interested the memory mechanisms of the visual cognitions, such as, which parts of the brain are responsible for the visual memories, which pars are for long term memory, which parts are for short term memory, *etc.* A block classifier would not answer such questions.

However, on the other hand, by studies in previous cognition science, we know that the brain areas have clear and natural structures: different areas exhibit different functions. Thus we can investigate more details mechanisms of human brains by making use of the the functional grouping in brain tissues, which leads to deeper understanding of our human beings.

Notations. In this chapter, we use the following notations. $\langle \mathbf{A}, \mathbf{B} \rangle$ denotes the inner product of two matrix **A** and **B** with the same size: $\langle \mathbf{A}, \mathbf{B} \rangle = \sum_{ij} A_{ij} B_{ij}$. ℓ_0 norm of a scale

x is defined as, $||x||_0 = 1$ if $x \neq 0$, $||x||_0 = 0$ if $x = 0$. $||\mathbf{X}||_F^2 = \sum_{ij} \mathbf{X}_{ij}^2$ is the Frobenius norm. $||\mathbf{x}|| = ||\mathbf{x}||_2 = \sqrt{\sum_i \mathbf{x}_i^2}$. $||\mathbf{x}||_1 = \sum_i |\mathbf{x}_i|$. ℓ_2/ℓ_1-norm of a matrix $\mathbf{A}$ is

$$||\mathbf{A}||_{\ell_2/\ell_1} = \sum_i \sqrt{\sum_j A_{ij}^2}. \tag{3.8}$$

ℓ_2/ℓ_1-norm was first proposed in [35] as rotational invariant ℓ_1 norm for the purpose of robust subspace learning. It is a valid *norm* because it satisfies the triangle inequality $||A||_{\ell_2/\ell_1} + ||B||_{\ell_2/\ell_1} \geq ||A + B||_{\ell_2/\ell_1}$ and two other conditions. On should notice that in literacy (*e.g.* [57]), we also use the $|| \cdot ||_{p,q}$ norm, which is defined as, $||\mathbf{A}||_{p,q} = \max_{\mathbf{x}} ||\mathbf{A}\mathbf{x}||_p/||\mathbf{x}||_q$. And notice that this p, q-norm (with $p = 2, q = 1$) is different from the one we used in our chapter. Thus we use "$\ell_{2,1}$-norm" instead of "$2, 1$-norm" to distinguish them. However, ℓ_2/ℓ_0-norm is not a valid norm because it does not satisfy the positive scalarbility: $||\alpha\mathbf{A}||_{\ell_2/\ell_0} = |\alpha|||\mathbf{A}||_{\ell_2/\ell_0}$ for any scalar α. The term "*norm*" here is for convenience. Another structural norm we are interested in is the group ℓ_2/ℓ_1 norm $\sum_g \sqrt{\sum_{i\in g} \mathbf{x}_i^2} = \sum_g ||\mathbf{x}^g||$, where g here is groups of the indexes of $\mathbf{x}$. The corresponding group ℓ_2/ℓ_0 norm is $\sum_g ||||\mathbf{x}^g||||_0$.

3.3 Related Work

We begin with a brief discussion of the related work.

3.3.1 Related Sparsity Models

LASSO (Least Absolute Shrinkage and Selection Operator) [11] imposes flat ℓ_1 sparsity regularize on the model and is a natural yet simple way to select related variables.

A cousin to the Lasso is the *group Lasso* [47], where the covariates are assumed to be clustered in groups, and instead of summing the absolute values of each individual loading, the sum of Euclidean norms of the loadings in each group is used. When Reproducing Kernel Hilbert Spaces (RKHS) is used to measure the group fitting function, group Lasso turns out to be equivalent to learn the best convex combination of a set of basis kernels, where each kernel corresponds to one Hilbertian norm used for regularization [58].

Along these ℓ_1 norms approaches, some variants have been developed. For example, Bach *et al.* employ bootstrap approach to learn multiple Lasso models on data subsets, then use the intersection of the active variables to maintain a stable feature set [59]. And Meinshausen and Buehlmann use the same bootstrap strategy but select frequent enough variables. Adaptive Lasso [60] approximate the SCAD penalty [61] using data dependent weights with convex constraints.

Another direction of the sparsity is joint covariates selection [48, 50, 51, 53]. These models consider multi-task learning problems in feature selection, which incorporate multiple domain knowledge to learn common covariates.

Besides the linear and convex constraints, other nonlinear penalties are also developed to derive sparse solutions. Zuo *et al.* use elastic net to make the penalty more smooth and to allow the model to select more variable than n (the number of data points) [3]. Tibshirani *et al.* developed the fused Lasso model which favors smoothness along natural ordering of variables [62] and enhance understanding of the active features in many applications [63].

3.3.2 Related Optimization Techniques

In most cases of the previous models, the optimization problems are convex. Yet, simple algorithms, *e.g.* quadratic programming, are not efficient in many real world applications. Extensive techniques have been developed to tackle the optimization problems.

LARS (Least Angle Regression) obtains entire solution path, *i.e.* all solution for all features under all possible regularization parameter λ, by making use of the piecewise linear property of Lasso [26]. Osborne *et al.* made uses of the property of the dual of Lasso problem which leads to new insights into the characteristics of Lasso estimator and to an improved method of estimating the covariance matrix [56]. Mallat and Zhang solved the problem by greedy search [64], while other researchers tried to employ coordinate descend and soft thresholding, *e.g.* [65, 66, 67, 46]. Matching pursuit and orthogonal matching pursuit are also widely used in the sparse optimization problems [68, 69].

3.4 Structural Sparsity via Structural Regularizer

3.4.1 Structural Sparsity Regularizer

A typical sparse learning problem can be written as the following problem,

$$\min_{\mathbf{X}} J(\mathbf{X}) = f(\mathbf{X}) + \lambda\Phi(\mathbf{X}), \tag{3.9}$$

where f is a *convex* fitting function which measures how good the model fits the observation, $\Phi(\mathbf{X})$ is the sparsity regularizer, and λ is the parameter balancing between the fitting function and the regularizer. The regularizer $\Phi(\mathbf{X})$ can be in various forms for different purposes. Here we list 6 of them:

$$\begin{align}
\Phi_1(\mathbf{X}) &= ||\mathbf{X}||_1 = \sum_i \sum_j |\mathbf{X}_{ij}|, \tag{3.10a}\\
\Phi_0(\mathbf{X}) &= ||\mathbf{X}||_0 = \sum_i \sum_j ||\mathbf{X}_{ij}||_0, \tag{3.10b}\\
\Phi_{21}(\mathbf{X}) &= ||\mathbf{X}||_{\ell_2/\ell_1} = \sum_i \sqrt{\sum_j \mathbf{X}_{ij}^2}, \tag{3.10c}\\
\Phi_{20}(\mathbf{X}) &= ||\mathbf{X}||_{\ell_2/\ell_0} = \sum_i || \sqrt{\sum_j \mathbf{X}_{ij}^2}||_0, \tag{3.10d}\\
\Phi_{g21}(\mathbf{x}) &= ||\mathbf{x}||_{g\ell_2/\ell_1} = \sum_g ||\mathbf{x}^g||, \tag{3.10e}\\
\Phi_{g20}(\mathbf{x}) &= ||\mathbf{x}||_{g\ell_2/\ell_0} = \sum_g ||||\mathbf{x}^g||||_0, \tag{3.10f}
\end{align}$$

among which Eqs. (3.10a) and (3.10b) are for the purpose of flat sparsity and Eqs. (3.10c) – (3.10f) are for structural sparsity.

The purpose of the convex norms is to approximate the cardinality. For example, in feature selection problems, the feature we are interested in is a subset of the whole feature space. For this purpose, the most natural constraint is the cardinality constraint as presented Eqs. (3.10d) and (3.10f), which are our contributions on this direction.

These explicit ℓ_2/ℓ_0 and group ℓ_2/ℓ_0 norm problems are NP-hard. Fortunately, in this chapter we develop an optimization technique (the Lipschitz Auxiliary Function Approach) by reducing this problem into tractable sub-problems which can be solved optimally and efficiently. Empirical results show that our approaches outperform the ℓ_2/ℓ_1 and group ℓ_2/ℓ_1 relaxation.

For convenient discussion, without confusion, we sometimes use $\mathbf{X}$ to represent both cases matrix and vector in Eqs. (3.10c) – (3.10f) in the rest of the chapter. When $\mathbf{X}$ is a vector, the Frobenius norm is reduced to the ℓ_2 norm of the vector.

3.4.2 Optimization Overview

We show later that Eq. (3.9) can be reduced to the following problem by our Lipschitz Auxiliary Function approach,

$$\frac{1}{2}||\mathbf{X} - \mathbf{A}||_F^2 + \lambda\Phi(\mathbf{X}). \tag{3.11}$$

And Eq. (3.11) can be further reduced to the following

$$\frac{1}{2}||\mathbf{x} - \mathbf{a}||^2 + \lambda\phi(\mathbf{x}), \tag{3.12}$$

which has close form solutions in all the sparse regularizers listed above. We will show the reduction from Eq. (3.11) to Eq. (3.11) in Sections 3.6.1.2, and 3.6.2.1.

3.5 Lipschitz Auxiliary Function Approach

In machine learning and data mining, auxiliary function method is wide employed, including in the optimization of maximum likelihood estimation [70] and matrix factorization [71, 72]. In this chapter, we first present a novel Lipschitz auxiliary function which is a variant of the proximal [67] and is a general framework to solve the structural sparsity problems.

An auxiliary function for problem

$$\min_{\mathbf{X}} J(\mathbf{X}), \tag{3.13}$$

is a function which satisfies the following,

$$\begin{aligned} Z(\mathbf{X}, \mathbf{X}) &= J(\mathbf{X}) && (3.14) \\ Z(\mathbf{X}, \tilde{\mathbf{X}}) &\geq J(\mathbf{X}), \ \forall \mathbf{X}, \tilde{\mathbf{X}}. && (3.15) \end{aligned}$$

Then the iterative updating algorithm is,

$$\mathbf{X}^{k+1} = \arg\min_{\mathbf{X}} Z(\mathbf{X}, \mathbf{X}^k), k = 0, 1, \cdots, \tag{3.16}$$

where $\mathbf{X}^k$ is the result of the k-th iteration. Using this algorithm one can easily show that the objective function value of $J(\mathbf{X})$ will monotonically decrease:

$$J(\mathbf{X}^{k+1}) = Z(\mathbf{X}^{k+1}, \mathbf{X}^{k+1}) \leq Z(\mathbf{X}^{k+1}, \mathbf{X}^k) \leq Z(\mathbf{X}^k, \mathbf{X}^k) = J(\mathbf{X}^k). \tag{3.17}$$

The first inequality (3.17) comes from the auxiliary function property of (3.15), while the second inequality (3.17) is achieved by the definition of $\mathbf{X}^{k+1}$ in (5.6).

In the rest of this chapter, without further explanation, *valid auxiliary function* means any function which satisfies (3.14) and (3.15). Notice that given any function $J(\mathbf{X})$, the auxiliary functions are not unique.

A function $f : \mathbb{R}^{n\times m} \to \mathbb{R}$ is Lipschitz continuous with constant p if the following holds [73],

$$f(\mathbf{X}) \leq f(\tilde{\mathbf{X}}) + \langle \mathbf{X} - \tilde{\mathbf{X}}, \triangledown f(\tilde{\mathbf{X}}) \rangle + \frac{p}{2}||\mathbf{X} - \tilde{\mathbf{X}}||_F^2, \tag{3.18}$$

In this chapter, we consider optimization the type of objective functions $f(\mathbf{X})$ which is Lipschitz continuous. We propose a valid auxiliary function which can simplify norm regularization to an easier format, and thus the original regularization problem can be solved efficiently. We also provide convergent guarantee of the algorithms. If the norm is convex, we further provide the convergent rate guarantee of the algorithms.

As the foundation of this chapter, we provide the following theorem:

Theorem 3.5.1 *Consider the optimization problem in (3.9), if function $f : \mathbb{R}^{n\times m} \to \mathbb{R}$ is Lipschitz continuous smooth loss function with constant p, then the following function satisfies (3.14) and (3.15),*

$$Z(\mathbf{X}, \tilde{\mathbf{X}}) = \frac{p}{2}||\mathbf{X} - \mathbf{A}||_F^2 + \lambda\Phi(\mathbf{X}) + C, \tag{3.19}$$

where $\mathbf{A} = \tilde{\mathbf{X}} - \frac{1}{p}\triangledown f(\tilde{\mathbf{X}})$, and $C = f(\tilde{\mathbf{X}}) - \frac{1}{2p}||\triangledown f(\tilde{\mathbf{X}})||_F^2$.

We call the auxiliary function in (3.19) as *Lipschitz auxiliary function*.

Proof First we prove inequality Eq.(3.14)

$$\begin{aligned}
Z(\mathbf{X},\mathbf{X}) &= \frac{p}{2}\|\mathbf{X}-(\mathbf{X}-\frac{1}{p}\triangledown f(\mathbf{X}))\|_F^2+\lambda\Phi(\mathbf{X})+C \\
&= \frac{1}{2p}\|\triangledown f(\mathbf{X})\|_F^2+\lambda\Phi(\mathbf{X})+f(\mathbf{X})-\frac{1}{2p}\|\triangledown f(\mathbf{X})\|_F^2 \\
&= f(\mathbf{X})+\lambda\Phi(\mathbf{X}) \\
&= J(\mathbf{X}).
\end{aligned}$$

Second, we prove inequality Eq. (3.15) as

$$\begin{aligned}
Z(\mathbf{X},\mathbf{X}) &= f(\mathbf{X})+\lambda\Phi(\mathbf{X}) \\
&\leq f(\tilde{\mathbf{X}})+\langle\mathbf{X}-\tilde{\mathbf{X}},\triangledown f(\tilde{\mathbf{X}})\rangle+\frac{p}{2}\|\mathbf{X}-\mathbf{A}\|_F^2+\lambda\Phi(\mathbf{X}) \\
&= Z(\mathbf{X},\tilde{\mathbf{X}}),
\end{aligned}$$

where the inequality comes from the Lipschitz continuous condition of Eq. (3.18).

Now setting $\hat{X} = X^k$ in the auxiliary function of Eq. (3.19), the convergence guarantee Eq. (3.17) lead to the algorithm in Algorithm 1. The most important features of the algorithm are (A1) A in Line 3 is readily available because $f(X)$ is differentiable (A2) The *difficult* non-differential regularization term $\Phi(X)$ are now handled in Line 4 together with a *much simplified objective* in Line 4 (as explained in Section 3B), which have *closed-form solutions* and therefore can be easily and efficiently computed.

From this, one can easily develop new algorithms by utilizing the proposed auxiliary function according to the above observations. In this chapter, we provide a series of examples, including ℓ_2/ℓ_1-norms, trace norm, and ℓ_2/ℓ_0-norms.

Further discussions on Algorithm 1.

In general Lipschitz continuous Function is bounded by quadratic function. However, the constant p is not always easy to determine. This is not a problem in designing the algorithm, because we can use an initial guess of p and update it when necessary. we have the following algorithm and theorem.

Algorithm 2 The *GLAF* (General Lipschitz Auxiliary Function) Algorithm.

```
Require: $f(\cdot), \lambda, \Phi(\cdot), p_0, \mathbf{X}_0, \gamma$
 1: $p \leftarrow p_0, \mathbf{X} \leftarrow \mathbf{X}_0, \tilde{\mathbf{X}} \leftarrow \mathbf{X},$
 2: while Not converged do
 3:     $\mathbf{A} \leftarrow \mathbf{X} - \frac{1}{p} \nabla f(\mathbf{X})$
 4:     Solve $\mathbf{X}^b \leftarrow \arg\min_{\mathbf{U}} \|\mathbf{U} - \mathbf{A}\|_F^2 + \lambda \Phi(\mathbf{U}),$
 5:     if $J(\mathbf{X}^b) < J(\mathbf{X})$ then
 6:         $\mathbf{X} \leftarrow \mathbf{X}^b$                    ▷ Lipschitz condition satisfied.
 7:     else
 8:         $p \leftarrow \gamma p$                    ▷ Lipschitz condition not satisfied.
 9:     end if
10: end while
11: return X
```

Here $f(\cdot)$, λ, and $\Phi(\cdot)$ define the learning model, p_0 is the initial guess of the Lipschitz continues constant, and $\mathbf{X}_0$ is the initialization. The only optimization parameter in the algorithm is γ, which is set to 1.1 in all our applications. Since the initial guess of the Lipschitz continues constant p_0 might be too small such that the inequality of (3.18) is not satisfied, which leads to $J(\mathbf{X}^b) > J(\mathbf{X})$ (line 8 in *GLAF* Algorithm). Thus we increase p by a factor γ until (3.18) is satisfied. Hence the parameter γ does not change the converged solution, indicating that our algorithm requires no optimization parameter. Further more, we have the following guarantee of the convergence for our algorithm.

Theorem 3.5.2 *For any lower bounded Lipschitz continues function f, Algorithm 2 converges.*

Notice that we have no requirement on the penalty function of Φ.

Line 4 of the *GLAF* Algorithm is a sub-problem of optimizing (3.19), which needs to be solved. We fucus on this optimization in the rest of the chapter.

Our algorithm makes use of the Lipschitz continues properties of the objective functions in the auxiliary function point of view. In previous related work, researchers have also developed algorithms using Proximal Gradient Method, such as [49].

3.6 Two Examples of Application

We use group lasso and multi-task learning as two examples to illustrate the application of our optimization techniques on non-smooth and non-convex norms for structural learning problems.

We first develop a new algorithm using the auxiliary function approach developed in the previous section to solves the group lasso and multi-task learning problem. Then we employ the auxiliary function approach in a more challenging non-convex version of the corresponding learning problems.

3.6.1 Group Lasso

3.6.1.1 Group Lasso by Lipschitz Auxiliary Function

In *group lasso*, we are interested in solving the following problem,

$$\min_{\boldsymbol{\beta}} J(\boldsymbol{\beta}) = ||\mathbf{y} - \sum_{g} \sum_{i \in g} \mathbf{x}^i \beta_i||^2 + \lambda \sum_{g} ||\boldsymbol{\beta}^g||, \tag{3.20}$$

where $\mathbf{X} = [\mathbf{x}^1, \mathbf{x}^2, \cdots, \mathbf{x}^p]$. One simple algorithm to solve the group LASSO problem is *quadratic programming*, which is not efficient here. Due to the piecewise linearity of the set of solutions as a function of the regularization parameter λ [26]. For the group Lasso, however, the path is only piecewise differentiable, and following such a path is not as efficient as for the Lasso. Recently, researchers have been putting effort on new algorithms to solve (3.20), *e.g.* [47, 58, 74].

As an example of the Lipschitz auxiliary function, we here develop a new algorithm to solve (3.20).

Obviously, since $\cup_{i=1}^{K} g_i = \{1, 2, \cdots, n\}$, (3.20) can be rewritten as,

$$\min_{\boldsymbol{\beta}} J(\boldsymbol{\beta}) = ||\mathbf{y} - \mathbf{X}^T\boldsymbol{\beta}||^2 + \lambda \sum_g ||\boldsymbol{\beta}^g||, \tag{3.21}$$

For (3.21) the Lipschitz auxiliary function is

$$Z(\boldsymbol{\beta}, \tilde{\boldsymbol{\beta}}) = \frac{p}{2}||\boldsymbol{\beta} - \mathbf{a}||^2 + \lambda \sum_g ||\boldsymbol{\beta}^g|| + C, \tag{3.22}$$

where

$$\mathbf{a} = \tilde{\boldsymbol{\beta}} - \frac{2\left(\mathbf{X}\mathbf{X}^T\tilde{\boldsymbol{\beta}} - \mathbf{X}\mathbf{y}\right)}{p}, \tag{3.23}$$

and

$$C = ||\mathbf{y} - X\tilde{\boldsymbol{\beta}}||^2 - \frac{2||\mathbf{X}\mathbf{X}^T\tilde{\boldsymbol{\beta}} - \mathbf{X}\mathbf{y}||^2}{p}. \tag{3.24}$$

Notice that $\mathbf{a}$ and C are constants *w.r.t.* $\boldsymbol{\beta}$. In order to employ the general framework of Algorithm 2, we need to solve the following sub-problem,

$$\min_{\boldsymbol{\beta}} J_Z(\boldsymbol{\beta}) = Z(\boldsymbol{\beta}, \tilde{\boldsymbol{\beta}}) = \frac{p}{2}||\boldsymbol{\beta} - \mathbf{a}||^2 + \lambda \sum_g ||\boldsymbol{\beta}^g||, \tag{3.25}$$

where the constant C is ignored. Notice that both of the ℓ_2-norm and ℓ_2/ℓ_1-norm can be group-wise decoupled:

$$\begin{aligned}\min_{\beta} J_Z(\boldsymbol{\beta}) &= \frac{p}{2}\sum_g ||\boldsymbol{\beta}^g - \mathbf{a}^g||^2 + \lambda \sum_g ||\boldsymbol{\beta}^g|| \\ &= \sum_g \left(\frac{p}{2}||\boldsymbol{\beta}^g - \mathbf{a}^g||^2 + \lambda||\boldsymbol{\beta}^g||\right). \qquad (3.26)\end{aligned}$$

In general, we have the following,

Theorem 3.6.1 *The optimal solution of (3.26) is given by,*

$$\boldsymbol{\beta}_g = \begin{cases} \mathbf{0} & if \quad \lambda \geq p||\mathbf{a}_g|| \\ \frac{p||\mathbf{a}_g||-\lambda}{p||\mathbf{a}_g||}\mathbf{a}_g & if \quad \lambda < p||\mathbf{a}_g|| \end{cases}. \qquad (3.27)$$

The proof utilizes the following lemma (with $\mu = \lambda/p$),

Lemma 3.6.2 *The global optimal solution of*

$$J(\mathbf{u}) = \frac{1}{2}||\mathbf{u} - \mathbf{a}||^2 + \mu||\mathbf{u}|| \qquad (3.28)$$

is given by

$$\mathbf{u} = \begin{cases} \mathbf{0} & if \quad \lambda \geq ||\mathbf{a}|| \\ \frac{||\mathbf{a}||-\mu}{||\mathbf{a}||}\mathbf{a} & if \quad \mu < ||\mathbf{a}|| \end{cases}. \qquad (3.29)$$

The proof of the lemma will be given in the Appendix A.

3.6.1.2 ℓ_2/ℓ_0-norm Group Lasso by Lipschitz Auxiliary Function

In this subsection,we first present the explicit ℓ_2/ℓ_0-norm Group Lasso, then an efficient algorithm is developed to solve the induced optimization problem.

In the ℓ_2/ℓ_0-norm group Lasso, we are interested the group Lasso problem in two forms,

$$\min_{\boldsymbol{\beta}} J(\boldsymbol{\beta}) = ||\mathbf{y} - \sum_g \sum_{i \in g} \mathbf{x}^i \beta_i||^2 + \lambda \sum_g ||||\boldsymbol{\beta}^g||||_0, \tag{3.30}$$

and

$$\begin{aligned} \min_{\boldsymbol{\alpha}} J(\boldsymbol{\alpha}) = ||\mathbf{y} - \sum_g \sum_{i \in g} \mathbf{x}^i \beta_i||^2 \\ \text{s.t.} \quad \sum_g ||||\boldsymbol{\beta}^g||||_0 \leq \xi \end{aligned} \tag{3.31}$$

3.6.1.3 ℓ_2/ℓ_0-norm as Penalty

For (3.30) the Lipschitz auxiliary function is

$$Z(\boldsymbol{\beta}, \tilde{\boldsymbol{\beta}}) = \frac{p}{2}||\boldsymbol{\beta} - \mathbf{a}||^2 + \lambda \sum_g ||||\boldsymbol{\beta}^g||||_0 + C, \tag{3.32}$$

where $\mathbf{a}$ and C are defined as the same as (3.23) and (3.24). Then one needs to solve

$$\begin{aligned} \min_{\boldsymbol{\beta}} J_Z(\boldsymbol{\beta}) =& \frac{p}{2} \sum_g ||\boldsymbol{\beta}^g - \mathbf{a}^g||^2 + \lambda \sum_g ||||\boldsymbol{\beta}^g||||_0 \\ =& \sum_g \left(\frac{p}{2}||\boldsymbol{\beta}^g - \mathbf{a}^g||^2 + \lambda||||\boldsymbol{\beta}^g||||_0 \right). \end{aligned} \tag{3.33}$$

For this problem, we have the following theorem,

Theorem 3.6.3 *The optimal solution of (3.33) is given by,*

$$\boldsymbol{\beta}_g = \begin{cases} \mathbf{0} & \textit{if} \quad \lambda \geq p||\mathbf{a}_g||/2 \\ \mathbf{a}_g & \textit{if} \quad \lambda < p||\mathbf{a}_g||/2 \end{cases}, \ G \in \mathcal{G}. \tag{3.34}$$

Proof Since (3.34) can be decoupled *w.r.t.* G, without loss of generation, we solve the following problem,

$$\min_{\boldsymbol{\beta}} J(\boldsymbol{\beta}) = \frac{p}{2}||\boldsymbol{\beta} - \mathbf{a}||^2 + \lambda||||\boldsymbol{\beta}||||_0. \tag{3.35}$$

Obviously, $\forall \boldsymbol{\beta} \neq 0, \lambda L_0(||\boldsymbol{\beta}||) = \lambda$. In this case, the minimum of the first term $\frac{p}{2}||\boldsymbol{\beta} - \mathbf{a}||^2$ is zeros, when $\boldsymbol{\beta} = \mathbf{a}$. Then $J(\boldsymbol{\beta}) = \lambda$. And if $\boldsymbol{\beta} = 0, \lambda L_0(||\boldsymbol{\beta}||) = 0$, and $J(\boldsymbol{\beta}) = p||\mathbf{a}||/2$. Thus when $\lambda \geq p||\mathbf{a}||/2$, $\boldsymbol{\alpha} = 0$ gives the lowest objective value, while $\lambda < p||\mathbf{a}||/2, \boldsymbol{\beta} = \mathbf{a}$ gives the lowest objective value. Thus, by considering the decoupling property of (3.33), (3.34) gives the optimal solution.

Notice that the solution of (3.34) is not continuous on the boundary $\lambda = p||a_g||_2$. However, on this boundary, whichever of the two solutions gives the same objective value.

3.6.1.4 ℓ_2/ℓ_0-norm as Constraint

For (3.31), the Lipschitz auxiliary function is

$$Z(\boldsymbol{\beta}, \tilde{\boldsymbol{\beta}}) = \frac{p}{2}||\boldsymbol{\beta} - \mathbf{a}||^2 + C, \tag{3.36}$$

where **a** and C are defined as the same as (3.23) and (3.24). In order to employ Algorithm 2, by ignoring the constant term C and the positive coefficient $p/2$, we need to solve the following constrained problem,

$$\begin{aligned} &\min_{\boldsymbol{\beta}} ||\boldsymbol{\beta} - \mathbf{a}||^2 \\ \text{s.t.} \qquad &\sum_g ||||\boldsymbol{\beta}^g||||_0 \leq \xi \end{aligned} \tag{3.37}$$

Unlike, other problems in previous discussion, this problem cannot be decoupled with respective to g. However, we provide the following,

Theorem 3.6.4 *The solution of problem (3.37) is given by,*

$$\boldsymbol{\beta}_i = \begin{cases} \mathbf{a}_i & \textit{if} \quad i \in g_{\pi(k)}, k \leq \xi \\ \mathbf{0} & \textit{if} \quad i \in g_{\pi(k)}, k > \xi \end{cases}, \tag{3.38}$$

where π is the sorting index such that $||\mathbf{a}^{g_{\pi(1)}}|| \geq ||\mathbf{a}^{g_{\pi(2)}}||, \cdots, \geq ||\mathbf{a}^{g_{\pi(K)}}||$, and $||\mathbf{a}^{g_k}|| = \sqrt{\sum_{i\in g_k} \mathbf{a}_i^2}$.

Proof We rewrite the objective and constraint of (3.37) as,

$$J(\boldsymbol{\beta}) = ||\boldsymbol{\beta} - \mathbf{a}||^2 = \sum_g ||\boldsymbol{\beta}^g - \mathbf{a}^g||^2 \triangleq \sum_g \Delta^g, \tag{3.39}$$

and

$$\sum_g ||\boldsymbol{\beta}^g||_0 \triangleq \sum_g \tau^g \leq \xi. \tag{3.40}$$

For any $\boldsymbol{\beta}^g \neq 0$, $||\boldsymbol{\beta}^g||_0 = 1$. In such case, the optimal $\boldsymbol{\beta}_G$ which gives the lowest objective value is $\boldsymbol{\beta}^g = \mathbf{a}^g$. And for any $\boldsymbol{\beta}^g = 0$, $\Delta^g = ||\boldsymbol{\beta}^g - \mathbf{a}^g||^2$. Thus, in order to give the lowest objective value, we have to select the first $\lfloor \xi \rfloor$ (the largest integer not larger than ξ) largest Δ_G and set the corresponding $\boldsymbol{\beta}^g = \mathbf{a}^g$, which gives the solution of (3.38).

3.6.2 Multi-Task Learning

3.6.2.1 Multi-Task Learning With ℓ_2/ℓ_1-norm Regularization

Let us consider (3.5a). Due to the piecewise linearity of the set of solutions as a function of the regularization parameter λ [26]. For multi-task learning, however, the path is only piecewise differentiable, and following such a path is not as efficient as for the Lasso. Recently, researchers have been putting effort on new algorithms to solve (3.5a), *e.g.* [53].

As an example of the Lipschitz auxiliary function, we here develop a new algorithm to solve (3.5a).

$$Z(\mathbf{B}, \tilde{\mathbf{B}}) = \frac{p}{2}||\mathbf{B} - \mathbf{A}||_F^2 + \lambda \sum_i \sqrt{\sum_j B_{ij}^2} + C, \tag{3.41}$$

where

$$\mathbf{A} = \tilde{\mathbf{B}} - \frac{\mathbf{X}\mathbf{X}^T\tilde{\mathbf{B}} - \mathbf{X}\mathbf{Y}}{p}, \tag{3.42}$$

and

$$C = \frac{1}{2}||\mathbf{Y} - \mathbf{X}^T\tilde{\mathbf{B}}||_F^2 - \frac{||\mathbf{X}\mathbf{X}^T\tilde{\mathbf{B}} - \mathbf{X}\mathbf{Y}||_F^2}{2p}. \tag{3.43}$$

Notice that $\mathbf{A}$ and C are constants *w.r.t.* $\mathbf{B}$. In order to employ the general framework of *GLAF* Algorithm, we only need to solve the following sub-problem,

$$\min_{\mathbf{B}} \frac{p}{2}||\mathbf{B} - \mathbf{A}||_F^2 + \lambda \sum_i \sqrt{\sum_j \mathbf{B}_{ij}^2}, \tag{3.44}$$

where the constant C is ignored. Notice that both of the ℓ_2-norm and ℓ_2/ℓ_1-norm can be row-wise decoupled:

$$\begin{aligned}\min_{\mathbf{B}} J_Z(\mathbf{B}) &= \frac{p}{2}\sum_i \sum_j (\mathbf{B}_{ij} - \mathbf{A}_{ij})^2 + \lambda \sum_i \sum_j \sqrt{\mathbf{B}_{ij}^2} \\ &= \sum_i \left(\frac{p}{2}\|\mathbf{b}^i - \mathbf{a}^i\|^2 + \lambda\|\mathbf{b}^i\|\right).\end{aligned}$$

where $\mathbf{B} = \left(\mathbf{b}^1, \mathbf{b}^2, \cdots \mathbf{b}^p\right)^T$, $\mathbf{A} = \left(\mathbf{a}^1, \mathbf{a}^2, \cdots \mathbf{a}^p\right)^T$, and $\mathbf{x}^i$ and $\mathbf{a}^i$ are the i-th row of $\mathbf{B}$ and $\mathbf{A}$, respectively. By directly using Lemma 3.6.2, we have

Theorem 3.6.5 *The optimal solution of (3.44) is given by,*

$$\mathbf{b}^i = \begin{cases} \mathbf{0} & if \quad \lambda \geq p\|\mathbf{a}^i\| \\ \frac{p\|\mathbf{a}^i\| - \lambda}{p\|\mathbf{a}^i\|}\mathbf{a}^i & if \quad \lambda < p\|\mathbf{a}^i\| \end{cases} \tag{3.45}$$

This gives an effective algorithm for ℓ_2/ℓ_1-norm regularization problems.

3.6.3 Multi-Task Learning With ℓ_2/ℓ_0-norm Regularization

In the ℓ_2/ℓ_0-norm multi-task learning, we are interested the regression problem in the following form,

$$\min_X \frac{1}{2}\|\mathbf{Y} - \mathbf{X}^T\mathbf{B}\|_F^2 + \lambda \sum_i \|\|\mathbf{b}^i\|\|_0, \tag{3.46}$$

The corresponding Lipschitz auxiliary function is

$$Z(\mathbf{B}, \tilde{\mathbf{B}}) = \frac{p}{2}\|\mathbf{B} - \mathbf{A}\|_F^2 + \lambda \sum_i \|\|\mathbf{b}^i\|\|_0 + C, \tag{3.47}$$

where A and C are defined as in (3.42) and (3.43). Then *GLAF* Algorithm requires a solution to the following problem,

$$\begin{aligned}\min_{\mathbf{B}} J_Z(X) &= \frac{p}{2}||\mathbf{B}-\mathbf{A}||_F^2 + \lambda \sum_i ||||\mathbf{b}^i||||_0 \\ &= \sum_i \left(\frac{p}{2}||\mathbf{b}^i - \mathbf{a}^i||^2 + \lambda||\mathbf{b}^i||_0\right). \end{aligned} \tag{3.48}$$

By applying Theorem 3.6.3 we have

Theorem 3.6.6 *The optimal solution of (3.48) is given by,*

$$\mathbf{b}^i = \begin{cases} \mathbf{0} & \textit{if} \quad \lambda \geq p||\mathbf{a}^i||^2/2 \\ \mathbf{a}^i & \textit{if} \quad \lambda < p||\mathbf{a}^i||^2/2, \end{cases} \tag{3.49}$$

3.6.4 ℓ_2/ℓ_0-norm as Constraint

One can also impose the ℓ_2/ℓ_0-norm as constraint,

$$\begin{aligned} &\min_X \frac{1}{2}||\mathbf{Y} - \mathbf{X}^T\mathbf{B}||_F^2 \\ \text{s.t.} \ &\sum_i ||||\mathbf{b}^i||||_0 \leq \xi, \end{aligned} \tag{3.50}$$

The corresponding Lipschitz auxiliary function is equivalent to

$$\min_{\mathbf{B}} J_Z(\mathbf{B}) = \frac{1}{2}||\mathbf{B}-\mathbf{A}||_F^2 \quad s.t. \ \sum_i ||||\mathbf{b}^i||||_0 \leq \xi, \tag{3.51}$$

where **A** is defined as same as (3.42). Following the similar techniques in Theorem 3.6.4 we have the following,

Theorem 3.6.7 *The following gives the global optimal solution of (3.51).*

$$\mathbf{x}^{\pi(j)} = \begin{cases} \mathbf{a}^{\pi(j)} & j \leq \xi \\ \mathbf{0} & j > \xi, \end{cases} \tag{3.52}$$

where $X = \left(\mathbf{x}^1, \mathbf{x}^2, \cdots \mathbf{x}^p\right)^T$, $A = \left(\mathbf{a}^1, \mathbf{a}^2, \cdots \mathbf{a}^p\right)^T$ *and* π *is the sorting index such that* $||\mathbf{a}^{\pi(1)}|| \geq ||\mathbf{a}^{\pi(2)}||, \cdots, || \geq \mathbf{a}^{\pi(p)}||$.

3.7 Optimization Algorithm Analysis

In this section, we provide more theoretical properties of our algorithms.

3.7.1 Convergent Rate of *GLAF* Algorithm

Theorem 3.7.1 *Let* $\mathbf{X}^0$ *be the initialization of Algorithm* GLAF *and* $\mathbf{X}^1, \cdots, \mathbf{X}^T$ *be the updating results of first* T *iterations. Assume that there exist a point set* $\mathcal{D}$ *and* $T_0 < T$ *such that* $X_t \in D, t = T_0, T_0 + 1, \cdots T$, *and* $\Phi(X)$ *is convex on* $\mathcal{D}$, *then the following bound holds,*

$$J(\mathbf{X}^T) - J(\mathbf{X}^*) \leq \frac{p_T||\mathbf{X}^{T_0} - \mathbf{X}^*||_F^2}{2(T - T_0)},$$

where $\mathbf{X}^*$ *is the local optimal of Eq. (3.9) in* $\mathcal{D}$, p_T *is the* p *value in* T*-th iteration.*
The proof will be given in the Appendix B.

Theorem 3.7.1 suggests that when the solution is close to the local minimum, the convergent rate is $O(1/t)$ where t is the iteration number. The requirement of the convexity of $\Phi(\mathbf{X})$ on $\mathcal{D}$ is easy to satisfy, since that in a small region around local minimum, the norm functions can be precisely approximated by quadratic functions.

3.7.2 Computational Complexity Analysis

For multi-task learning, the computation cost comes from two parts: The computation of matrix **A** and the optimization of Lipschitz auxiliary function.

3.7.2.1 The computation of **A**

A is defined in (3.42). Notice that $\mathbf{X}$ is a $p \times n$ matrix and $\mathbf{B}$ is a $p \times K$ matrix, where p, n, K are the number of dimension, data points, and tasks, respectively. If p is large, one can first compute $\tilde{Y} = \mathbf{X}^T\tilde{\mathbf{B}}$ then compute $\mathbf{X}\mathbf{X}^T\mathbf{B} = \mathbf{X}\tilde{Y}$, both of which cost $O(npK)$. If n is large and we do not want to compute $\mathbf{X}\mathbf{X}^T\mathbf{B}$ in each iteration of the main loop in *GLAF* Algorithm, we can compute $\mathbf{X}\mathbf{X}^T$ before the main loop. Then in each iteration, the cost of computing $\mathbf{X}\mathbf{X}^T\mathbf{B}$ is p^2K, which is $O(1)$ with respect to n.

3.7.2.2 The optimization of Lipschitz auxiliary function

For ℓ_2/ℓ_1-norm, the solution of Lipschitz auxiliary function is given by Theorem 3.6.5. It is easy to check that the computational cost is $O(pK)$. For ℓ_2/ℓ_0-norm penalty (see (3.46)) form, the computation cost of Lipschitz auxiliary function is $O(pK)$ (see Theorem 3.6.6). And for ℓ_2/ℓ_0-norm constraint (see (3.50)) form, the cost of Lipschitz auxiliary function is $O(pK + \log(p))$ (see Theorem 3.6.7).

3.8 Accelerated Lipschitz Auxiliary Function Optimization

Nesterov shows that gradient method is capable to reach the convergent rate of $O(1/t^2)$ [75, 73]. More recently, many optimization techniques demonstrate that for some non-smooth function, similar convergent rate can also be derived. In this section, we develop an accelerated version of GLAF (AGLAF) by following the techniques in [67] or [53].

Again, the *AGLAF* algorithm requires no optimization parameters (the choice of γ does not change the convergency of the algorithm). For this algorithm, we have the following property,

Theorem 3.8.1 *Let* $\mathbf{X}^0$ *be the initialization of Algorithm* AGLAF *and* $\mathbf{X}^1, \cdots, \mathbf{X}^T$ *be the updating result of first* T *iterations. Assume that there exists a set* $\mathcal{D}$ *and* $T_0 < T$ *such that* $\mathbf{X}_t \in D, t = T_0, T_0 + 1, \cdots T$, *and* $\Phi(\mathbf{X})$ *is convex on* $\mathcal{D}$*, then the following bound holds,*

$$J(\mathbf{X}^T) - J(\mathbf{X}^*) \leq \frac{2p_T \|\mathbf{X}^{T_0} - \mathbf{X}^*\|_F^2}{(T - T_0 + 1)^2},$$

where $\mathbf{X}^*$ *is the local optimal of Eq. (3.9) in* $\mathcal{D}$*,* p_T *is the* p *value in* T*-th iteration.*

The proof is similar to that in Appendix of [53]. We omit the proof here. The convergent property of *GLAF* and *AGLAF* will be studies in the experimental section.

3.9 Experimental Results

In this section, we validate the efficiency of the presented algorithms. We first test our algorithm in an SNPs (Single Nucleotide Polymorphisms) data set in the 21st chromosome of *H. sapiens*[2], After that four image data sets (MSRC [3], *AT&T* face database[4], *barcelona* dataset[5], and the *TrecVideo 2006* [76]) and one music data [77] are used to compare the efficiency of the ℓ_2/ℓ_0-norm with ℓ_2/ℓ_1-norm group Lasso and multi-task learning.

3.9.1 Group Lasso

Data mining techniques are widely used in Bioinformatics, such as [78, 79]. Here we use the SNPs for the application of Group Lasso.

[2] http://hapmap.ncbi.nlm.nih.gov/downloads/phasing/2005-03_phaseI/full/genotypes_chr21_CEU.phased.gz

[3] http://research.microsoft.com/en-us/projects/objectclassrecognition/default.htm

[4] http://www.cl.cam.ac.uk/research/dtg/attarchive/ facedatabase.html

[5] http://mlg.ucd.ie/content/view/61

Table 3.1. Objective value, feature recovery recall and precision comparison on SNPs data under different number of selected SNP blocks.

#Block	Objective		Recall		Precision	
	ℓ_2/ℓ_1	ℓ_2/ℓ_0	ℓ_2/ℓ_1	ℓ_2/ℓ_0	ℓ_2/ℓ_1	ℓ_2/ℓ_0
10	41679.10	6434.77	29.85	55.22	86.96	100.00
15	14816.15	2597.58	49.25	65.67	81.67	93.62
30	1560.54	1513.90	70.15	74.63	75.81	80.51
60	338.58	238.2	85.07	88.06	53.27	58.82

For this experiment, we solve the problem of (3.21) and (3.30). To generate the data matrix X, we select 200 SNPs in the 21st chromosome of *H. sapiens* for 120 patients, *i.e.* X is a 120×200 matrix.

In order to generate the grouping of the SNPs, first detect the blocks using Linkage Disequilibrium (LD) of SNPs, see Figure 3.1 A. We first calculate the LD values of neighbor SNPs: $v_i = \mathbf{LD}(i, i+1)$, which is plotted in Figure 3.1 B. Then we cut v using 0.2 to split the 200 SNPs into 79 blocks, see Figure 3.1 C. We use the block structure as the groups in group Lasso. To get the response $\mathbf{y}$, we randomly select 10, 15, 30, and 60 blocks and we let

$$\mathbf{y} = \sum_{k=1}^{N} \sum_{i \in g_k} \mathbf{x}^i \beta_i + \epsilon,$$

where $\epsilon \sim \mathcal{N}(0, 0.1)$ is drawn from normal distribution (with $\mu = 0$ and $\sigma = 0.1$) and $N = 10, 15, 30, 60$. With the $\mathbf{X}, \mathbf{y}$, and the selected groups, we trained the model of group Lasso using ℓ_2/ℓ_1-norm and ℓ_2/ℓ_0-norm.

We evaluate the objective values ($||\mathbf{y} - \mathbf{X}^T \boldsymbol{\beta}||^2$), the precision and recall of the feature recovered under different number of selected blocks. The results are shown in Table 3.1. One can observe that the ℓ_2/ℓ_0-norm group Lasso achieve much lower objective. As a byproduct, it also generate higher recall and precision under the same number of selected blocks.

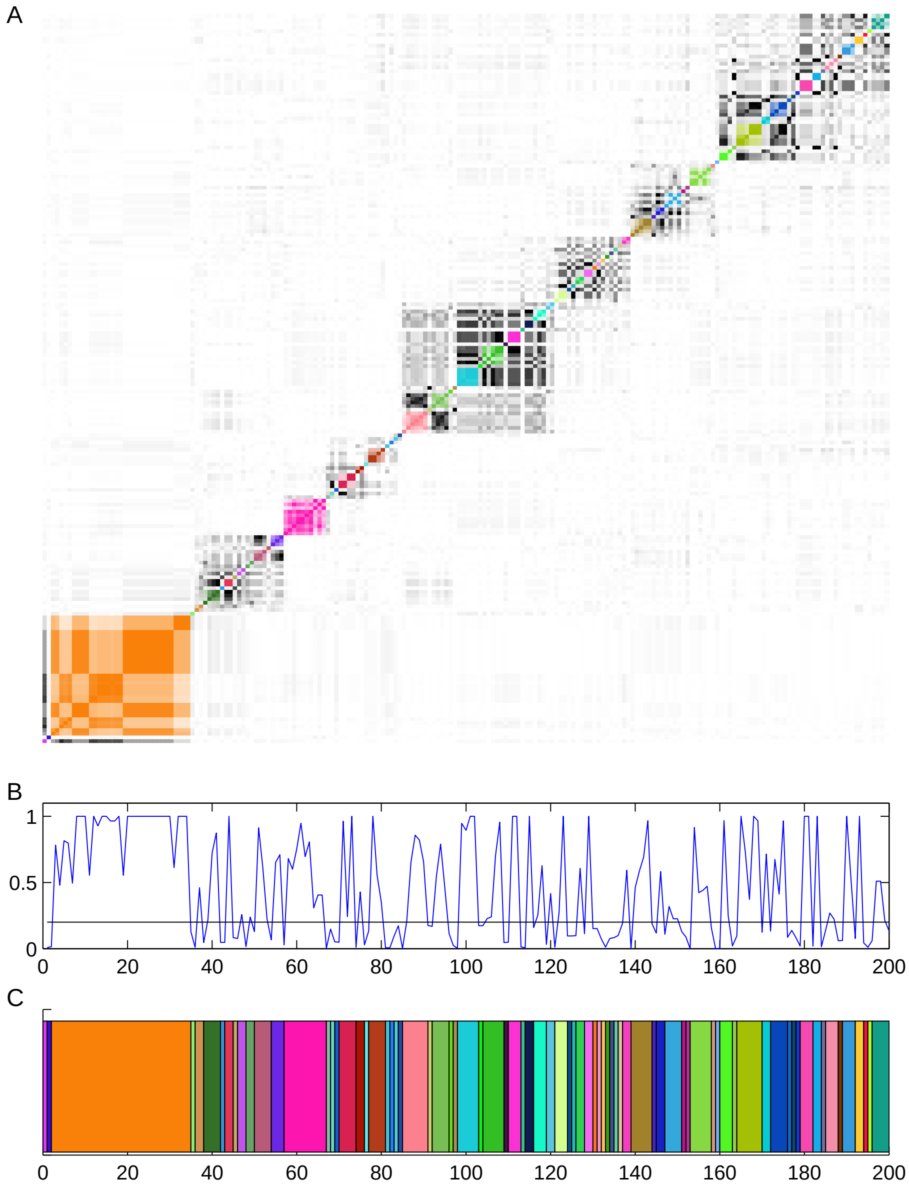


Figure 3.1. [5] SNP dataset used in our experiment. A: the Linkage disequilibrium (LD) values of pairwise SNPs. B: The LD values of neighbor SNPs. C: The block structure split using threshold of 0.2 in B. .

From Table 3.1 we see that ℓ_2/ℓ_0 norms are consistently better than ℓ_2/ℓ_1 norms in all the measurements we are interested.

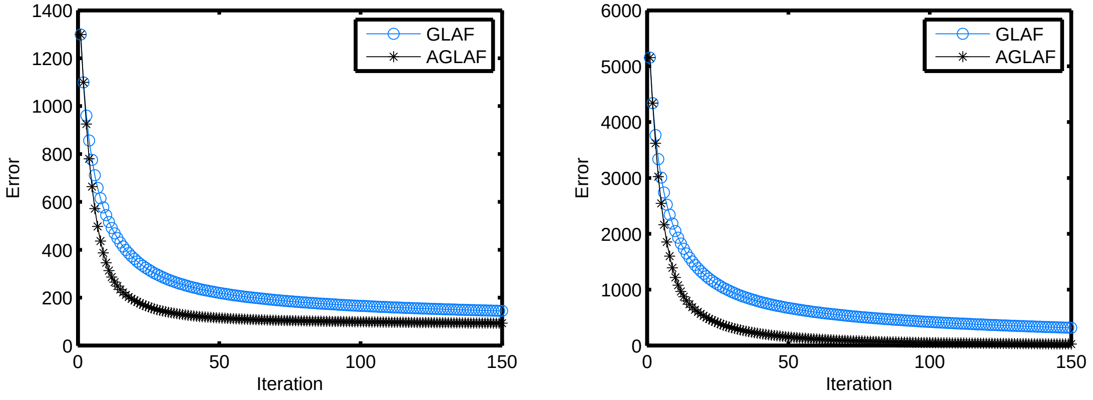
3.9.2 Multi-Task Learning on SNP Data

In this experiment, we solve the following problem,

$$\min_{X} ||\mathbf{Y} - \mathbf{X}^T B||_F^2 + \lambda\Phi(\mathbf{B}), \tag{3.53}$$

where either $\Phi(\mathbf{B}) = ||\mathbf{B}||_{\ell_2/\ell_1}$ or $\Phi(\mathbf{B}) = ||\mathbf{B}||_{\ell_2/\ell_0}$. We employ both *GLAF* and *AGLAF* algorithms to solve these problems. The data is generated as following. We select 100 single nucleotide polymorphisms (SNPs) from the 21st human Chromosome across 120 patients [80], which forms a 100×120 matrix (denoted by $\mathbf{X}_t$). Then we randomly generate a 100×20 matrix (denoted by $\mathbf{B}_t$) and let $\mathbf{Y}_t = \mathbf{X}^T\mathbf{B}_t + \sigma z$, where $\sigma = 0.05$ and $z \sim N(0, 1)$ is a Gaussian noise. Here we are simulating a multi-task learning problem with 20 tasks in which only the selected 100 SNPs are related to the target. We also randomly select other T_n SNPs from the same chromosome to form a $T_n \times 120$ matrix (denoted by $\mathbf{X}_n$) and let $\mathbf{X} = [\mathbf{X}_t^T, \mathbf{X}_n^T]^T, \mathbf{Y} = \mathbf{Y}_t$. Here $\mathbf{X}_t$ is relevant to the tasks and $\mathbf{X}_n$ is the noise. In this experiment, we try to recover the correlated SNPs from $\mathbf{X}$ and $\mathbf{Y}$. We set $\lambda = 1$ and $T_n = 300$ in this experiment. We apply both *GLAF* and *AGLAF* for 200 iterations and measure the error at each iteration, which are plotted in Figure 5.3. The error is computed as $Error = ||\mathbf{Y} - \mathbf{X}^T\mathbf{B}||_F^2/||\mathbf{Y}||_F^2$. One can observe that *AGLAF* always converges faster than *GLAF* in both cases, even for non-convex ℓ_2/ℓ_0-norm shown in Figure 3.2(b).

We also compare the ℓ_2/ℓ_1 norm and ℓ_2/ℓ_0 norm under different $T_n = 100, 300, 900$. For each norm and every λ we have different $||\mathbf{Y} - \mathbf{X}^T\mathbf{B}||_F^2/||\mathbf{Y}||_F^2$ values and different numbers of selected SNPs, which are plotted in Figure 3.3. We can see from the figure using the same number of SNPs, ℓ_2/ℓ_0 norm method gives much lower error.



(a) ℓ_2/ℓ_1-norm

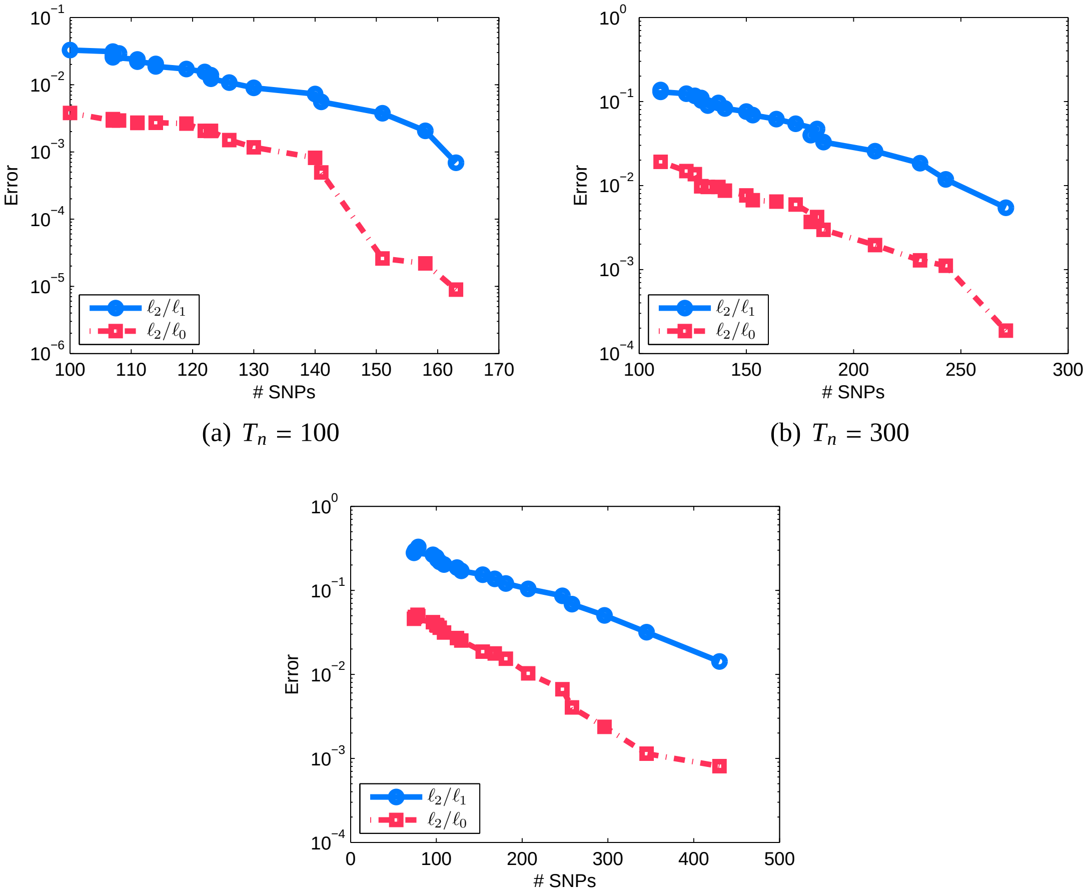
(b) ℓ_2/ℓ_0-norm

Figure 3.2. [5] The convergence of ℓ_2/ℓ_1-norm (left), and ℓ_2/ℓ_0-norm (right) for *GLAF* and *AGLAF* methods..

3.9.3 Multi-Task Learning on Image Data

Here we try to solve (3.53) as multi-task feature learn. $\mathbf{X}_{p\times n}$ and $\mathbf{Y}_{n\times K}$ are obtained as following. For *AT&T* and *Barcelona*, we use pixels as features. For *TrecVideo* and*MSRC* we evenly divide each image into $8 \times 8 = 64$ blocks and compute the first and second moments (mean and variance) of each color band and total get $64 \times 2 \times 3 = 384$ moment features $\mathbf{x}^i$. Let $\mathbf{X} = [\mathbf{x}^1, \mathbf{x}^2, \cdots, \mathbf{d}^p]^T, p = 384$. And $\mathbf{Y}_{ik} = 1$ if the i-th image belongs to the k-th group, $\mathbf{Y}_{ik} = 0$ otherwise, $k = 1, \cdots, K$ where K is the number of groups. *AT&T* : $n = 400, p = 10304, K = 40$; *Barcelona*: $n = 139, p = 10000, K = 4$; *TrecVideo*: $n = 384, p = 3718, K = 39$; *MSRC*: $n = 591, p = 380, K = 23$, *Music*: $n = 593, p = 78, K = 6$. We compare the error of $||\mathbf{Y} - \mathbf{X}^T\mathbf{B}||_F^2/||\mathbf{Y}||^2$ under different selected number of pixels using ℓ_2/ℓ_1-norm and ℓ_2/ℓ_0-norm learning, which are plotted in Figure 3.4. For all the datasets, ℓ_2/ℓ_0-norm obtains much lower objective than ℓ_2/ℓ_1-norm.

We also solve the model in (3.50) with $\xi = 1000, 2000, \cdots, 6000$. We select three columns of X and plot them as images in left panel of Figure 3.5 under different choice of ξ. Some discriminant areas are highlighted in rectangles. For example, the third person has long hair on the top left of her head, the corresponding area has negative values. For



(a) $T_n = 100$

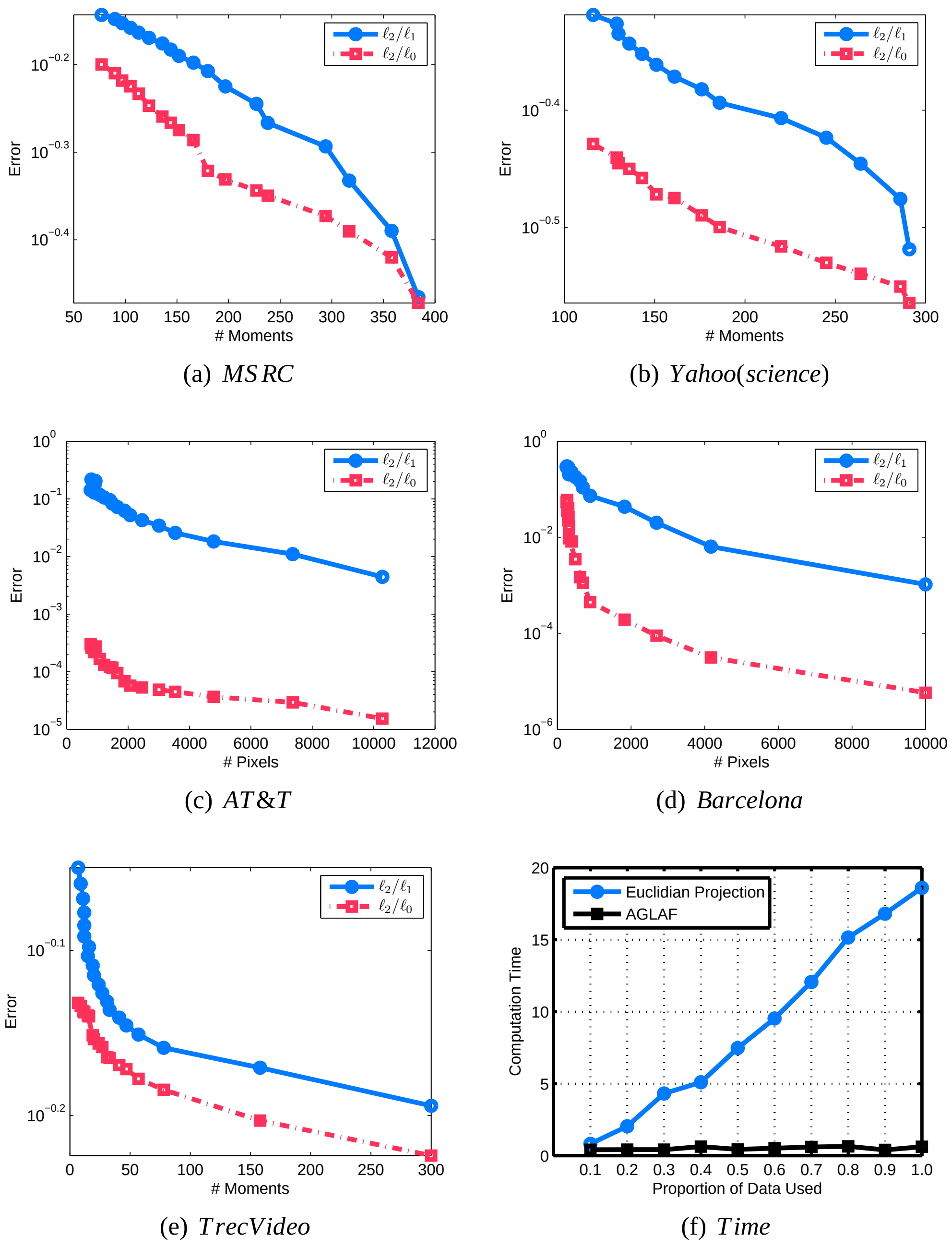
(b) $T_n = 300$

(c) $T_n = 900$

Figure 3.3. [5] Error comparisons under different number of selected SNPs for ℓ_2/ℓ_1 norm and ℓ_2/ℓ_0 norm in three synthetic data sets. In each data set, we add 100 (left), 300 (middle), and 900 (right) irrelevant SNPs. .

all other persons, this area is not dark, and thus has large negative values in the decision function $\mathbf{b}_k^T d$, where $\mathbf{b}_k$ is decision weight vector for 3rd person. A heat map of the covariance of the $\mathbf{b}$ are also plotted in the right panel. From Figure 3.5, one can observe that the sparsity of the multi-task learning problem obtained by our algorithm is quite consistent with human interpretation.

We also compare the running time for ℓ_2/ℓ_1 norm (which is implemented using Euclidian Projection method [53]) and our method in Figure 3.4(f). The computational time



(a) *MSRC*

(b) *Yahoo*(*science*)

(c) *AT&T*

(d) *Barcelona*

(e) *TrecVideo*

(f) *Time*

Figure 3.4. [5] Error comparison under different number of selected pixels/moments for *MSRC* (a), Yahoo (b), *AT&T* (c), Barcelona (d), and TrecVideo (e) data set. (f) is the computational time comparison of Projected Gradient method and our method on TrecVideo.
.

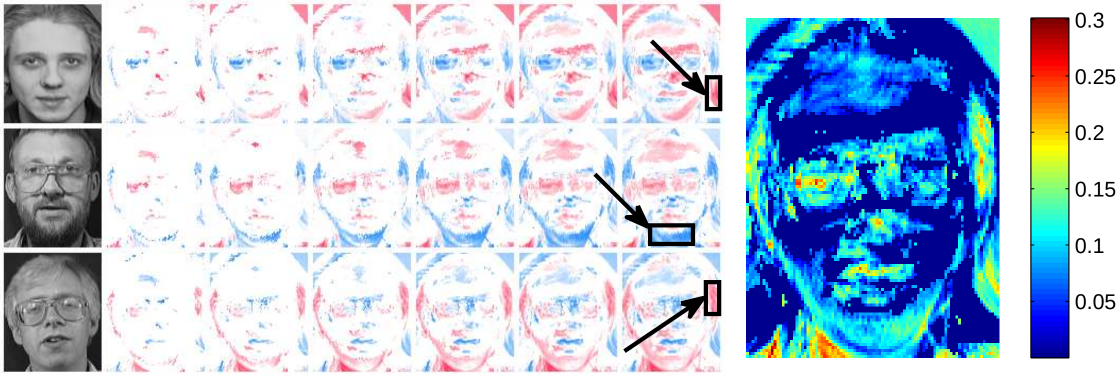


Figure 3.5. [5] Visualization of columns of X of solution in multi-task learning with by ℓ_2/ℓ_0-norm constraint. The left most column in the left panel are example images for each person. From the second column to the last column are the results for $\xi = 1000, 2000, \cdots, 6000$. Red color represents positive values, dark blue represents negative values, and white color for 0. The discriminative areas of the corresponding people are highlighted by black arrows. Right panel are the variance of $\mathbf{x}_i$ over different tasks. Higher variance indicates higher discriminative capability..

is calculated using 10%, 20%, $\cdots$, 100% of the data. For Euclidian Projection method, we use the software (version 3.0) downloaded at web site[6] with default settings. For our method, the computation time includes the gradient computation and the updating time of Eq. (5.6). One can see that the computation time for our algorithm remains approximately constant with respective to the number of data points while the Euclidian Projection method grows linearly. These results indicate that our method is much more efficient for large-scale data.

3.9.4 Experiments on TCGA Data

We continue to make use of the TCGA data described in Section 2.6.5. We use the subtype of ovarian cancer as the multi-tasks and analysis the bio-marker patterns in discriminating the cancer subtypes. The results indicate that both *has-mir-200c* and *PI3K*

[6] http://www.public.asu.edu/ jye02/Software/SLEP/download.htm

have high coefficients to discriminate subtype 1 and subtypes 2/3. And this is consistent with results in Section 2.6.5.

CHAPTER 4

ORTHOGONAL REGRESSION FOR NON-REDUNDANCY FEATURE SELECTION

4.1 Motivations

Recent trends in healthcare and medicine enhance traditional knowledge driven approaches with data extracted information, considered together with knowledge for making treatment and other decisions. As more and more comprehensive EHR data become available, a diverse set of clinical features can be constructed and potentially leveraged for clinical decision support applications. From both theoretical and application perspectives, feature selection is a key component with a lot of challenges.

From statistics and machine learning research, feature selection provides many benefits: 1) speed up the subsequent learning process, 2) improve the model generalizability and alleviate the effect of the curse of dimensionality [81] and overfitting [82]. A large number of feature selection methods have been proposed in the literature [83, 84, 85, 86, 87] and there are many recent reviews and workshops devoted to this topic, *e.g.*, NIPS Conference [88]. Despite the vast literature on feature selection, the problem is by no algorithms solved. Many practical feature selection are developed in the context of concrete applications, such as Bioinformatics applications[89, 90]. A survey on various feature selection methods and applications are presented in Section 4.2.

Our motivating healthcare application and its associated new challenges for feature selection are presented next.

Motivating example: EHR data provide a longitudinal view of patients. This typically includes diagnosis info such as ICD9 codes, medication info such as drug names, lab results and symptoms. EMR data have been growing rapidly in quantity over the past few years, and are increasingly considered to be a valuable asset by leading medical institutions. Predictive modeling using EHRs for targeted high cost diseases has become highly valuable in modern healthcare. One high cost disease is Heart Failure (HF). The clinical and societal implications of HF are truly staggering. One in 5 US citizens over age 40 is expected to develop HF in their lifetime and HF is the leading cause of hospitalization among Medicare beneficiaries. With the aging population, HF will continue to be a leading cause of healthcare use. The hope is that through mining the longitudinal EHR data, predictive features can be identified from a large number of input features that will aid us predict HF with high accuracy. Furthermore, the selected features should be parsimonious (i.e., non-redundant). Often there is a known set of features (risk factors) that leads to HF. Any additional features should not only have great predictive value to HF but also complement to the known risk factors in order to minimize redundancy.

Motivated by this clinical application, we propose Scalable Orthogonal Regression (SOR) [1] to address the aforementioned requirements. In particular, SOR has the following properties:

- *Scalable:* SOR achieves nearly linear scale-up with respect to the number of input features and the number of samples;
- *Optimal:* SOR is formulated as a sparse learning problem that can be solved efficiently using alternative convex optimization with theoretical convergence and global optimality guarantee;
- *Non-redundant:* SOR is designed specifically to select less redundant features without sacrificing the quality, where redundancy is measured by an orthogonality measure added as a penalty term in the objective function;
- *Extensible:* SOR can enhance an existing set of preselected features by adding additional features that complement the existing set but still with strong predictive power.

[1] Most of the major results in this chapter have been published in paper [91].

In order to evaluate our algorithm, we compare other state-of-the-art feature selection algorithms in 9 real data sets from various domains, including gene expression, general UCI benchmark data, and multimedia data. Extensive experimental results confirmed that SOR significantly outperforms several state of the art feature selection methods with respect to various quality metrics. In particular, SOR achieves orders of magnitude improvement of speed compared to several other methods. Besides overall competitive AUC measure, SOR can also achieve less redundancy and better stability in terms of selected features.

As a case study, we apply SOR to a clinical application on predictive modeling of HF. The study is done on over 20 million real EHR records on 30K patients over 7 years from a large healthcare provider network. The data contain diagnosis, medication, lab results and HF diagnostic symptoms. The goal is to predict the onset of HF x months before the actual diagnosis. In our cross validation evaluation, we achieve increased AUC measure in comparison to knowledge driven baseline which is provided by clinical experts.

The rest of the chapter is organized as the follows. A brief survey on various feature selection methods and applications are presented in Section 4.2. We then introduce our method and the related optimization algorithms in Section 4.3. Theoretical analysis for our method is given in Section 4.4. We demonstrate the quality and scalability of our algorithm in Section 4.5. Finally we highlight a case study on EHR data in the experimental section.

4.2 Related Work

In feature selection, our purpose is to select a subset of K informative features where K is the number of required features. There are two major sub-problems in feature selection. One is the measurement of *how informative a given subset of features is*, and the other one is how to obtain the subset of features. Given a measurement of the quality of features, the feature selection problem is essentially a combinatorial optimization problem, and is usually solved by an approximation or greedy search. In general, there are two types of feature selection methods in the literature: (1) filter methods [83] where the selection is independent of classifiers and (2) wrapper methods [84] where the selection is tightly coupled with a specific classifier.

The filter methods evaluate features one by one, then select the top K features according to their scores. This type of scheme can be interpreted as a greedy approach by iteratively selecting one feature from the remaining unselected feature set. Within this category, one can implement it using two approaches. Univariate filtering, *e.g.* Information Gain, or multivariate filtering, *e.g.* Minimum Redundancy-Maximum Relevance (mRMR) [90].

Feature selection using wrapper methods provides an alternative way to obtain multivariate subset selection by incorporating the classifiers, *e.g.* directly approximating the area under the ROC curve [92] or optimization of the LASSO (Least Absolute Shrinkage and Selection Operator) model [93, 94].

The learning of non-redundant features has also been discussed in literature. For example, mRMR explicitly prefers low redundant features [90], and non-redundant codebook feature learning method was also proposed [95].

4.3 Sparse Orthogonal Regression

This section presents the *Sparse Orthogonal Regression* (SOR) algorithm in detail. First we will introduce some notation and symbols that will be used throughout the chapter.

4.3.1 Notations

We use $\mathbf{X}$ to denote the data matrix containing n observations on the p covariates: $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_p] \in \mathbb{R}^{n\times p}$. Without the loss of generality, we assume all covariate vectors are normalized, i.e., $||\mathbf{x}_i||_2 = 1$ $(i = 1, \cdots, p)$. As we only care about the supervised setting in this chapter, we are further given the corresponding response vector $\mathbf{y} \in \mathbb{R}^n$, then the feature selection problem is a *linear regression* under *square loss*, which takes the following form.

$$\min_{\alpha} J_r(\boldsymbol{\alpha}), \quad J_r(\boldsymbol{\alpha}) = \frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{\alpha}||^2 = \frac{1}{2}||\mathbf{y} - \sum\nolimits_j \alpha_j \mathbf{x}_j||^2, \tag{4.1}$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_p]^T \in \mathbb{R}^p$ is the regression coefficient vector. The absolute value of $|\alpha_j|$ can be regarded as the importance of covariate j, $j = 1, 2, \cdots, p$. If $\alpha_i = 0$, then that means covariate i is not selected.

4.3.2 Othogonality of Features

As *nonredundancy* is one of the major claims of the method we proposed in this chapter, we first give the definition of the *redundancy* between two covatiates.

Definition 1 (Redundancy). *Given two covariates $\mathbf{x}_i$ and $\mathbf{x}_j$, as well as their corresponding regression coefficients α_i and α_j (which are fixed) as in Eq.(4.1), we define the* redundancy *between them as follows,*

$$R_{ij} = \left(\alpha_i \alpha_j \mathbf{x}_i^T \mathbf{x}_j\right)^2. \tag{4.2}$$

Obviously, if $\mathbf{x}_i$ and $\mathbf{x}_j$ are orthogonal to each other, then $\mathbf{x}_i^T \mathbf{x}_j = 0$ and $R_{ij} = 0$, indicating that they are totally non-redundant. If $\mathbf{x}_i$ and $\mathbf{x}_j$ are identical, then $\mathbf{x}_i^T \mathbf{x}_j$ is maximized. In this case, $\mathbf{x}_i$ and $\mathbf{x}_j$ are redundant.

Based on definition 1, in order to obtain a set of non-redundant covariates, we can minimize the following objective

$$J_o(\boldsymbol{\alpha}) = \frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{\alpha}||^2 + \frac{\beta}{4} \sum_{ij} \left(\alpha_i \mathbf{x}_i^T \mathbf{x}_j \alpha_j\right)^2, \tag{4.3}$$

where the term $\sum_{ij} R_{ij} = \sum_{ij} \left(\alpha_i \mathbf{x}_i^T \mathbf{x}_j \alpha_j\right)^2$ is the summation of the redundancies over all pairwise features, and β is a tradeoff parameter which controls the importance of the redundancy.

In feature selection, we also want the number of selected features to be as small as possible, thus we further impose the sparsity penalty term of $||\boldsymbol{\alpha}||_1$ on the objective function. Then our goal becomes to minimize the following objective.

$$J(\boldsymbol{\alpha}) = \frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{\alpha}||^2 + \lambda||\boldsymbol{\alpha}||_1 + \frac{\beta}{4}\sum_{ij}\left(\alpha_i \mathbf{x}_i^T \mathbf{x}_j \alpha_j\right)^2, \tag{4.4}$$

where $||\boldsymbol{\alpha}||_1$ is the ℓ_1 norm of $\boldsymbol{\alpha} : ||\boldsymbol{\alpha}||_1 = \sum_j |\alpha_j|$. We will show later that $J(\boldsymbol{\alpha})$ is convex and develop an efficient algorithm to minimize $J(\boldsymbol{\alpha})$ with respect to $\boldsymbol{\alpha}$.

Here λ is a model parameter which controls the sparsity. One can easily show that if $\lambda_i \geq \max_i |(\mathbf{X}^T\mathbf{y})_i|$, $\boldsymbol{\alpha} = 0$ gives the optimal solution of Eq. (4.4). Thus the parameter λ has a natural range of $0 \sim \lambda_{max} = \max_i |(\mathbf{X}^T\mathbf{y})_i|$. In the rest of the chapter, without loss of generalization, we use a normalized λ (ranging from $0 \sim 1$, where $\lambda = 1$ indicate we use λ_{max}). Once the optimal solution of $\boldsymbol{\alpha}^*$ is obtained, we use the absolute values of $|\alpha_i^*|$ as the importance of features.

Our method performs particularly well in cases where the problem includes identifying a set of relevant predictors from a really large collection of variables that are not necessarily independent. We will provide detailed evidence in the experimental section.

4.3.3 Preliminaries

In this section we will present some preliminaries on how to minimize Eq. (4.4). For notational convenience, we will use

$$f(\boldsymbol{\alpha}) = J_o(\boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\alpha}\|^2 + \frac{\beta}{4}\sum_{ij}\left(\alpha_i \mathbf{x}_i^T \mathbf{x}_j \alpha_j\right)^2, \tag{4.5}$$

through the rest of this chapter. Before diving into the details, first we need to prove that $f(\alpha)$ is *locally Lipschitz continuous*, which is defined as follows.

Definition 2 (Lipschitz continuous) [96]. *A function $f : \mathbb{R}^d \longrightarrow \mathbb{R}^m$ is Lipschitz continuous if for $\forall \mathbf{a}, \mathbf{b} \in \mathbb{R}^d$, we can find a constant L satisfying the following inequality*

$$\|\mathbf{a} - \mathbf{b}\| \leqslant L\, \| f(\mathbf{a}) - f(\mathbf{b})\| \tag{4.6}$$

The function f is called locally Lipschitz continuous, *if for each $\mathbf{c} \in \mathbb{R}^m$, there exists an $L > 0$ such that f is Lipschitz continuous on the open ball of center $\mathbf{c}$ and radius L.*

$$B_L(\mathbf{c}) = \{\mathbf{x} \in \mathbb{R}^m : \|\mathbf{x} - \mathbf{c}\| < L\}. \tag{4.7}$$

In our case, as $f(\alpha)$ is continuously smooth, the gradient is locally Lipschitz continuous [97]. Then we have the following inequality [96].

$$f(\alpha) \leq f(\tilde{\alpha}) + (\alpha - \tilde{\alpha})^T \nabla f(\tilde{\alpha}) + \frac{L}{2}\|\alpha - \tilde{\alpha}\|^2, \tag{4.8}$$

which immediately leads to

$$\begin{aligned} & f(\boldsymbol{\alpha}) + \lambda\|\boldsymbol{\alpha}\|_1 \\ \leq\ & f(\tilde{\alpha}) + (\alpha - \tilde{\alpha})^T \nabla f(\tilde{\alpha}) + \frac{L}{2}\|\alpha - \tilde{\alpha}\|^2 + \lambda\|\boldsymbol{\alpha}\|_1. \end{aligned} \tag{4.9}$$

In this section, we will employ Eq. (4.10) and derive an efficient iterative algorithm which is guaranteed to converge to the global solution of minimizing Eq. (4.4). Denote the right hand side of Eq. (4.10) by $Z(\boldsymbol{\alpha}, \tilde{\boldsymbol{\alpha}})$, *i.e.*

$$Z(\alpha, \tilde{\alpha}) = f(\tilde{\alpha}) + (\alpha - \tilde{\alpha})^T \nabla f(\tilde{\alpha}) + \frac{L}{2} \|\alpha - \tilde{\alpha}\|^2 + \lambda \|\alpha\|_1, \qquad (4.10)$$

where ∇f is the gradient of f. Bringing $J(\boldsymbol{\alpha})$ in Eq.(4.4) into Eq.(4.10), we can easily find that

$$J(\boldsymbol{\alpha}) = Z(\boldsymbol{\alpha}, \boldsymbol{\alpha}) \leq Z(\boldsymbol{\alpha}, \tilde{\boldsymbol{\alpha}}). \qquad (4.11)$$

Then let $\tilde{\boldsymbol{\alpha}} = \boldsymbol{\alpha}^t$ and

$$\boldsymbol{\alpha}^{t+1} = \arg\min_{\alpha} Z(\boldsymbol{\alpha}, \boldsymbol{\alpha}^t), \qquad (4.12)$$

thus we have

$$J(\alpha^{t+1}) = Z(\alpha^{t+1}, \alpha^{t+1}) \leq Z(\alpha^{t+1}, \alpha^t) \leq Z(\alpha^t, \alpha^t) = J(\alpha^t) \qquad (4.13)$$

This suggests that we can iteratively update $\boldsymbol{\alpha}$ by solving problem (5.6) (i.e., minimizing $Z(\boldsymbol{\alpha}, \tilde{\boldsymbol{\alpha}})$ with $\tilde{\boldsymbol{\alpha}} = \boldsymbol{\alpha}^t$) to decrease the objective function monotonically.

4.3.4 Algorithm Details

Based on the contents in last subsection, in order to minimize Eq.(4.4), we need to solve the following sub-problem iteratively

$$\min_{\alpha} Z(\boldsymbol{\alpha}, \boldsymbol{\alpha}^t). \qquad (4.14)$$

As $f(\boldsymbol{\alpha}^t)$ is constant with respect to $\boldsymbol{\alpha}$, we can minimize the following objective instead with respect to $\boldsymbol{\alpha}$

$$J_m(\alpha) = (\alpha - \alpha^t)^T \nabla f(\alpha^t) + \frac{L}{2} \|\alpha - \alpha^t\|^2 + \lambda \|\alpha\|_1, \qquad (4.15)$$

where the gradient of $f(\alpha)$ is

$$[\nabla f(\alpha)]_i = \left[\mathbf{X}^T\mathbf{X}\alpha\right]_i + \beta \sum\nolimits_j \left(\alpha_i\alpha_j\mathbf{x}_i^T\mathbf{x}_j\right)\mathbf{x}_i^T\mathbf{x}_j\alpha_j, \tag{4.16}$$

which can be written in its matrix form as

$$\nabla f(\alpha) = (\mathbf{G} + \beta\mathbf{A} \odot \mathbf{G} \odot \mathbf{G})\,\alpha, \tag{4.17}$$

where $\mathbf{A} = \alpha\alpha^T$, $\mathbf{G} = \mathbf{X}^T\mathbf{X}$, and $\odot$ is the matrix Hadamard (elementwise) product.

Next we will show that the minimization of Eq. (4.15) has closed form solution. First, as $||\nabla f(\alpha^t)||$ is a constant with respect to α, then minimize $J_m(\alpha)$ in Eq. (4.15) is equivalent to minimize

$$\begin{aligned} & J_m(\alpha) + \frac{1}{2L^2}||\nabla f(\alpha^t)||^2 \\ = \; & \left(\alpha - \alpha^t\right)^T \nabla f(\alpha^t) + \frac{L}{2}||\alpha - \alpha^t||^2 + \frac{1}{2L^2}||\nabla f(\alpha^t)||^2 + \lambda||\alpha||_1 \\ = \; & \frac{L}{2}\left\|\alpha - \left(\alpha^t - \frac{1}{L}\nabla f(\alpha^t)\right)\right\|^2 + \lambda||\alpha||_1. \end{aligned}$$

Furthermore, we can easily prove the following Lemma.

Lemma 1. *The global minimum solution of minimizing the following objective over* $\mathbf{u}$

$$J(\mathbf{u}) = \frac{1}{2}||\mathbf{u} - \mathbf{a}||^2 + \mu||\mathbf{u}||_1, \tag{4.18}$$

where where $\mathbf{u} = [u_1, u_2, \cdots, u_p]^T$ *and* $\mathbf{a} = [a_1, a_2, \cdots, a_p]^T$ *are* $p \times 1$ *vectors, is given by*

$$u_i = \begin{cases} 0 & \text{if} \quad \mu \geq |a_i| \\ \frac{|a_i| - \mu}{|a_i|} a_i & \text{if} \quad \mu < |a_i| \end{cases}, \quad i = 1, 2, \cdots, p,$$

or equivalently,

$$u_i = (|a_i| - \mu)_+ \mathbf{sign}(a_i), \tag{4.19}$$

where $(x)_+ = x$ *if* $x > 0$, $(x)_+ = 0$ *if* $x <= 0$ *and* $\mathbf{sign}(\cdot)$ *is the sign function (*$\mathbf{sign}(0)$ *is defined as 0 here).*

By applying the above lemma, and letting $\mu = \lambda/L, \mathbf{u} = \boldsymbol{\alpha}, \mathbf{a} = \boldsymbol{\alpha}^t - \frac{1}{L}\nabla f(\boldsymbol{\alpha}^t)$, one can easily obtain the following close form optimal solution for minimizing Eq. (4.15),

$$\alpha_i = \left(\left|\left[\boldsymbol{\alpha}^t - \frac{1}{L}\nabla f(\boldsymbol{\alpha}^t)\right]_i\right| - \frac{\lambda}{L}\right)_+ \mathbf{sign}\left(\left[\boldsymbol{\alpha}^t - \frac{1}{L}\nabla f(\boldsymbol{\alpha}^t)\right]_i\right), \tag{4.20}$$

where $i = 1, 2, \cdots, p$.

The following Algorithm summarizes the whole procedure of our *Scalable Orthogonal Regression* (SOR) algorithm. In the algorithm γ is a optimization parameter to increase L when the Lipschitz condition is not satisifed and is set to be 1.2 in all experiments. Next section presents some analysis of the algorithm and its extensions.

Algorithm 3 SOR (Scalable Orthogonal Regression)

```
Require: $\lambda, L_0, \boldsymbol{\alpha}_0, \gamma$
while Not converged do
    Compute $\nabla f(\boldsymbol{\alpha})$ using Eq. (4.17)
    $\mathbf{a} \leftarrow \boldsymbol{\alpha} - \nabla f(\boldsymbol{\alpha})/L$
    Solve $\tilde{\boldsymbol{\alpha}} \leftarrow \arg\min_\alpha \|\boldsymbol{\alpha} - \mathbf{a}\|^2 + \lambda\|\alpha\|_1$ (Eq. (4.20))
    if $J(\tilde{\alpha}) < J(\alpha)$ then
        $\boldsymbol{\alpha} \leftarrow \tilde{\boldsymbol{\alpha}}$
    else
        $L \leftarrow \gamma L$
    end if
end while
return $\boldsymbol{\alpha}$
```

4.4 Analysis and Extension

In this section, we will provide some analysis and extensions of the SOR algorithm. First we show that the objective Eq. (4.4) is convex with respect to $\boldsymbol{\alpha}$.

4.4.1 Convexity

We have the following theorem.

Theorem 4.4.1 ***(Convexity).*** Eq. (4.4) is convex w.r.t. $\boldsymbol{\alpha}$.

Proof: See Appendix C.

Based on the convexity, we can prove the following theorem, which serves as the foundation of the follow up analysis on convergence rate.

Theorem 2 (Lipschitz Continuity). *f in Eq. (4.5) is locally Lipschitz continuous. Furthmore, there exists a global L such that Eq. (4.5) is Lipschitz continuous at $\boldsymbol{\alpha}_t$ with Lipschitz continuity constant L, where $\boldsymbol{\alpha}_t$ is the solution of Algorithm 3 at the t-th iteration.*

Proof: $f(\boldsymbol{\alpha})$ is continuously smooth, thus it is locally Lipschitz [97]. On the other hand, $f(\boldsymbol{\alpha})$ is convex and lower bounded, then the set $\mathcal{S} = \{\boldsymbol{\alpha} : f(\boldsymbol{\alpha}) \leq f(\boldsymbol{\alpha}^0)\}$ is close convex set. Obviously, $\boldsymbol{\alpha}_t \in \mathcal{S}$. As $f(\boldsymbol{\alpha})$ is locally Lipschitz with constant L_α at $\boldsymbol{\alpha}$, $L = \max_{\alpha \in \mathcal{S}} L_\alpha$ is obviously the global Lipschitz constant for the solutions of Algorithm 3.

4.4.2 Convergence

As discussed in section 4.3.3, SOR can monotonically decrease the value of $J(\boldsymbol{\alpha})$, and it is obvious that $J(\boldsymbol{\alpha})$ is lower bounded by zero, thus SOR will converge. Based on Theorem 1 and 2, we can prove the following theorem analyzing the convergence rate of Algorithm 1.

Theorem 4.4.2 ***(Convergence Rate of SOR).*** Algorithm 1 converges to global solution of Problem in Eq. (4.4). Furthermore,

$$J(\boldsymbol{\alpha}_T) - J(\boldsymbol{\alpha}^*) \leq \frac{L_T \|\boldsymbol{\alpha}_0 - \boldsymbol{\alpha}^*\|^2}{2T},$$

T is number of iterations in Algorithm 3, L_T is the value of L in the last iteration ,$\boldsymbol{\alpha}^*$ is the global optimal of Eq. (4.4), and $\boldsymbol{\alpha}_T$ is the output of Algorithm 3.

Proof: See the Appendix D.

Theorem 3 also guarantees that Algorithm 3 converges to the global solution, since $J(\boldsymbol{\alpha}_T) - J(\boldsymbol{\alpha}^*) \to 0$ as $T \to \infty$ (notice that $L_T \leq L$ because of the locally Lipschitz continuity of $f(\boldsymbol{\alpha})$ guaranteed by Theorem 2.

4.4.3 Accelerated Algorithm

As it is obvious that the $J_m(\boldsymbol{\alpha})$ in Eq. (4.15) is convex, we can also derive an accelerated algorithm shown in Algorithm 4.4.3, with much higher convergence rate. For the *accelerated SOR* (aSOR), we have the following theorem.

Theorem 4 (Convergence Rate of aSOR). *Algorithm 1 converges to global solution of Problem in Eq. (4.4). Furthermore,*

$$J(\boldsymbol{\alpha}_T) - J(\boldsymbol{\alpha}^*) \leq \frac{L_T \|\boldsymbol{\alpha}_0 - \boldsymbol{\alpha}^*\|^2}{2T^2},$$

T is number of iterations in Algorithm 4.4.3, L_T is the value of L in the last iteration ,$\boldsymbol{\alpha}^*$ is the global optimal of Eq. (4.4), and $\boldsymbol{\alpha}_T$ is the output of Algorithm 3.

The theorem can be proved using similar tricks as in [98], and we omit the details here due to limited space.

By comparing the convergence rate of SOR and aSOR, one should notice that the gap to the optimal solution in aSOR decreases as $\frac{1}{T^2}$, which is much faster than in SOR with $\frac{1}{T}$, where T is the number of iterations. We will demonstrate the convergence speed comparison of these two algorithms in the experimental section.

```
**Require:** $\lambda, p_0, \boldsymbol{\alpha}_0, \gamma$
 1: $p \leftarrow p_0, \boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha}_0, \eta \leftarrow \boldsymbol{\alpha}_0, \tilde{\boldsymbol{\alpha}} \leftarrow \boldsymbol{\alpha}, \zeta \leftarrow 1,$
 2: **while** Not converged **do**
 3:     $\mathbf{a} \leftarrow \boldsymbol{\eta} - \nabla f(\boldsymbol{\eta})/p,$
 4:     Solve $\tilde{\boldsymbol{\alpha}} \leftarrow \arg\min_{\alpha} ||\boldsymbol{\alpha} - \mathbf{a}||^2 + \lambda||\alpha||_1$ (Eq. (4.20))
 5:     **if** $J(\boldsymbol{\alpha}) < J(\boldsymbol{\eta})$ **then**
 6:         $\boldsymbol{\eta} \leftarrow \boldsymbol{\alpha} + 2(\zeta - 1)(\boldsymbol{\alpha} - \tilde{\boldsymbol{\alpha}})/(1 + \sqrt{1 + 4\zeta^2})$
 7:         $\tilde{\boldsymbol{\alpha}} \leftarrow \boldsymbol{\alpha}$
 8:         $\zeta \leftarrow (1 + \sqrt{1 + 4\zeta^2})/2$
 9:     **else**
10:         $p \leftarrow \gamma p$
11:     **end if**
12: **end while**
13: **return** $\alpha$
```

4.4.4 Computational Complexity

We will analyze the computational complexity of SOR in this section. Specifically, solving $\boldsymbol{\alpha}$ at Step 5 in Algorithm 3 needs $O(p)$ time, where p is the dimension of $\boldsymbol{\alpha}$. The computational bottleneck of the Algorithm 1 is the evaluation of the gradient of $f(\boldsymbol{\alpha})$ in Eq. (4.17), which needs $O(np^2)$ time at the first glance. However, we can develop a more efficient way to obtain the gradient in $O(np)$ time. Specifically, we can first compute $\mathbf{B} = \mathbf{X} \odot (\alpha \mathbf{e}^T)$, where $\mathbf{e} = [1, 1, \cdots 1]^T$ with proper size. Then $\mathbf{B}_{\ell j} = \alpha_j \mathbf{x}_j^{\ell}$ where $\mathbf{x}_j^{\ell}$ is the ℓ-th

Table 4.1. Complexity comparison of SOR, Information Gain (IG), LARS, and mRMR. For sparse features, α is the average proportion of nonzeros.

	Dense			
	SOR	IG	LARS	mRMR
Time	np	np	np^2	np^3
Storage	np	np	$np + p^2$	
	Sparse			
	SOR	IG	LARS	mRMR
Time	αnp	αnp	np^2	αnp^3
Storage	αnp	αnp	$\alpha np + p^2$	αnp

element of $\mathbf{x}_j$ or $\mathbf{b}_j = \alpha_j \mathbf{x}_j$, where $\mathbf{b}_j$ is the j-th column of $\mathbf{B}$. Obviously, the computation of $\mathbf{B}$ only needs $O(np)$ time. Then

$$\sum\nolimits_j \left(\alpha_i \alpha_j \mathbf{x}_i^T \mathbf{x}_j\right) \mathbf{x}_i^T \mathbf{x}_j \alpha_j = \alpha_i (\mathbf{x}_i^T \sum\nolimits_j \mathbf{b}_j)^2,$$

the summation of $\mathbf{v} = \sum_j \mathbf{b}_j$ takes $O(np)$ time, which does not depends on the index i. Notice that computing $\mathbf{x}_i^T \mathbf{v}$ only requires $O(n)$ time. One the other hand $\mathbf{X}^T \mathbf{X} \mathbf{y} = \mathbf{X}^T (\mathbf{X} \mathbf{y})$ also requires $O(np)$, thus the whole complexity of computing the gradient is $O(np)$.

We also compare the computational and storage complexity of SOR with some other state-of-the-art approaches (Information Gain, LARS, and mRMR), which are summarized in Table 4.1.

4.4.5 SOR with Preselected Features

In some real world scenarios, we may already have a set of features preselected with prior knowledge. For example, physicians in hospitals have years of experience on some specific diseases, they have their own knowledge on which features (factors) are more important. In this case, we may want to select a set of features (with data driven approaches) complementary to those preselected features.

Fortunately our SOR algorithm can easily adapted to incorporate this prior knowledge. Assume the preselected feature set is $\mathcal{P}$ and the remaining feature set is $\mathcal{Q}$, then we can partition the whole data matrix as $\mathbf{X} = [\mathbf{X}_{\mathcal{P}}, \mathbf{X}_{\mathcal{Q}}]$, where $\mathbf{X}_{\mathcal{P}}$, $\mathbf{X}_{\mathcal{Q}}$ only contains the observations on the features in $\mathcal{P}$ and $\mathcal{Q}$ and our goal is to select features from $\mathcal{Q}$. For the feature set $\mathcal{P}$, we first compute their regression coefficients with simple least squares:

$$\boldsymbol{\alpha}_{\mathcal{P}} = \arg\min_{\alpha} \|\mathbf{y} - \mathbf{X}_{\mathcal{P}}\boldsymbol{\alpha}\|^2 = (\mathbf{X}_{\mathcal{P}}^T \mathbf{X}_{\mathcal{P}})^{-1} \mathbf{X}_{\mathcal{P}}^T \mathbf{y}. \tag{4.21}$$

Then we define

$$\begin{aligned} f_p(\boldsymbol{\alpha}) =& \frac{1}{2}\|\mathbf{y} - \mathbf{X}_{\mathcal{Q}}\boldsymbol{\alpha}\|^2 \\ &+ \frac{\beta}{4}\left[\sum_{ij\in\mathcal{Q}} \left(\alpha_i \mathbf{x}_i^T \mathbf{x}_j \alpha_j\right)^2 + \sum_{i\in\mathcal{Q}, j\in\mathcal{P}} \left(\alpha_i \mathbf{x}_i^T \mathbf{x}_j \alpha_j\right)^2\right], \end{aligned}$$

where $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_{\mathcal{P}}^T, \boldsymbol{\alpha}_{\mathcal{Q}}^T]^T$ is the concatenated regression coefficient vector with $\boldsymbol{\alpha}_{\mathcal{P}}$ computed using Eq.(4.21). Note that there are two terms to punish the feature redundancy. One measures the feature redundancy selected from $\mathcal{Q}$, the other measures the redundancy between the feature selected from $\mathcal{Q}$ and the preselected feature set $\mathcal{P}$. Then we can minimize the following objective with respect to $\boldsymbol{\alpha}_{\mathcal{Q}}$.

$$J_p(\boldsymbol{\alpha}) = f_p(\boldsymbol{\alpha}) + \lambda\|\boldsymbol{\alpha}\|_1. \tag{4.22}$$

Comparing Eq. (4.4) and Eq. (4.22), one can immediately see that Algorithm still applies for the minimization of Eq. (4.22). The only step we need to change is the computation of

Table 4.2. Data Description

Data	# Pos	#Neg	#Sample	#Features	Type
heart	150	120	270	13	UCI
vehicle	416	430	846	18	UCI
coil	720	720	1440	1024	Image
jaffe	108	105	213	4096	Image
SRBCT	40	43	83	2308	Gene Expression
MLL	24	48	72	12582	Gene Expression

gradient. Notice that in this optimization, α_j is a constant for $j \in \mathcal{P}$. The corresponding gradient is

$$\nabla f_p(\boldsymbol{\alpha}) = (\mathbf{G} + \beta\mathbf{A} \odot \mathbf{G}_Q \odot \mathbf{G}_Q)\,\boldsymbol{\alpha} + \beta(\mathbf{X}_Q^T\mathbf{X}_{\mathcal{P}}\boldsymbol{\alpha}_{\mathcal{P}}) \odot \boldsymbol{\alpha}.$$

4.5 Experimental Results

In this section, will first demonstrate the convergence of *SOR* and *aSOR* and the scalability of the algorithm, then evaluate the quality (measured by AUC and stability) and orthogonality of the features selected by our algorithm.

4.5.1 Datasets

We evaluate our algorithm on various kinds of data. The first kind is the general datasets from UCI data mining and machine learning repository [99], which include heart and vehicle data sets. The second kind of data are image data, including Columbia object image library (coil) [100] and the Japanese Female Facial Expression (jaffe) Database[2]. The third type is gene expression data including MLL [101], and SRBCT [102]. We summarize the data description in Table 4.2.

[2] Available at http://www.kasrl.org/jaffe.html

4.5.2 Convergence

We now present the experiment on the convergence speed in Figure 4.2. For our algorithms 3 and 4.4.3, we set $\lambda = 0.1$ and $\beta = 0.1$. Figure 4.2 shows the objective function vs. number of iterations. It confirms that aSOR converges much faster than SOR on all data sets[3]. Next we will present the evaluation results compared to other feature selection methods.

4.5.3 Baselines

We compare with several feature selection methods with very different design:

- InfoGain: Information gain is a greedy approach that uses mutual information to select features.
- LARS gives the entire solution path of LASSO. For this method, we rank the features according to their order of turning from zero to nonzero in the solution path [93].
- mRMR: mRMR is another widely used feature selection method which aims at obtaining a set of non-redundant features by greedy search [90].

We have witnessed many other feature selection methods which are designed for various purposes as we discussed in Section 2. The purpose in our experiments here is to compare with the close related and representative feature selection methods in each category. Since we focus more in feature selection methods designed for general purpose, some other methods designed for specific classifiers (such as SVM-RFE [103, 104]) are not considered here.

4.5.4 Scalability

To test scalability, we generate different datasets by subsampling from a large dataset by varying the number of samples and features. The data dependent parameters include the

[3] We only present the results on 3 datasets, but the same trend persist on others

number of features p and the number of samples n. Figure 4.1 shows CPU time vs p or n. For our method, we use the following stop criteria. If $(J^t - J^{t+1})/J^t < 10^{-5}$ then we stop the algorithm, where J^t and J^{t+1} are the objective function values at the t-th and $t+1$-th iterations, respectively. For the other method, we use the default settings. We observe aSOR is orders of magnitudes more efficient than LARS and mMRMR. Among them, only aSOR and InfoGain can apply to large datasets with over 10K features and samples. In particular, despite its sophisticated optimization mechanism, aSOR achieves similar computational performance to InfoGain, which is a very simple and greedy method.

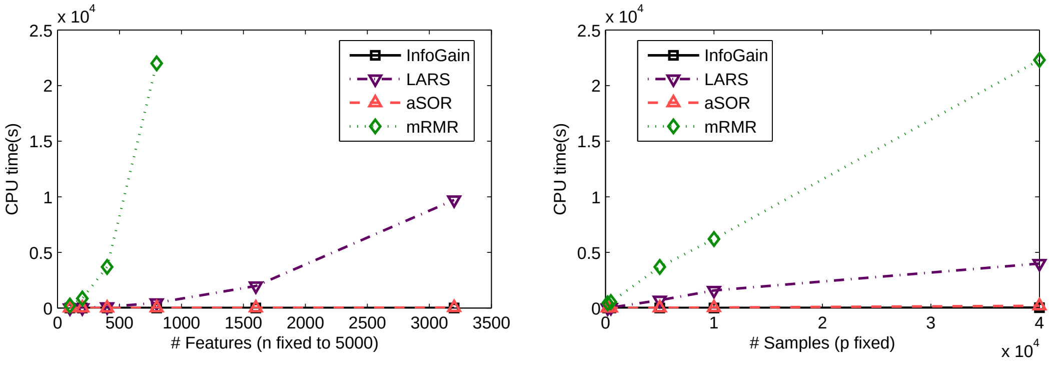


Figure 4.1. [91] CPU time comparison of Information Gain (InfoGain), LARS, aSOR, and mRMR. Left: fix he number of samples to 5000, and vary the number of features. Right: fix the number of features to 400, and vary the number of samples. .

4.5.5 Classification Accuracy

In all the comparison evaluation, we conduct a standard 80-to-20 split of the data at random at T times (in our case, T=20).

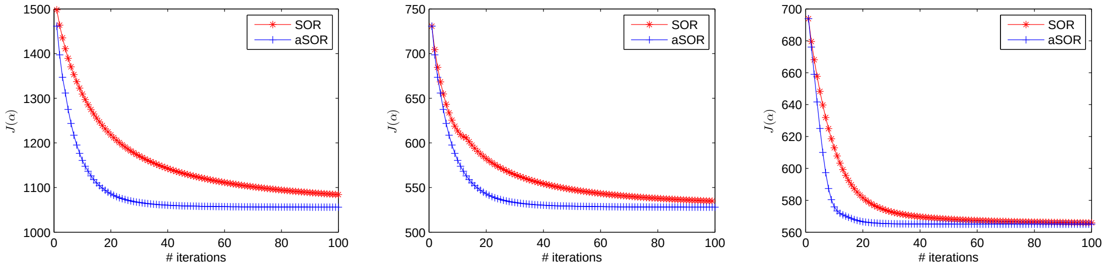


Figure 4.2. [91] Convergent rate comparison between algorithm SOR and aSOR on three data sets, *vehicle* ($p = 18, n = 846$), *coil*, ($p = 1024, n = 1440$) and *MLL*, ($p = 12582, n = 72$), where p is the number of dimensions and n is the number of samples. .

Table 4.3. [91] AUC and feature stability comparison with SOR, LARS, mRMR, and Information Gain. The best results on each data are highlighted in bold.

	Our		LARS		mRMR		Information Gain	
	AUC	Stable	AUC	Stable	AUC	Stable	AUC	Stable
MLL	**0.990± 0.024**	0.579	0.977±0.044	0.450	0.965±0.054	0.246	0.966±0.047	**0.589**
PROS	**0.967± 0.041**	**0.842**	0.956±0.044	0.794	0.944±0.056	0.422	0.959±0.046	0.755
SRBC	**0.990± 0.025**	**0.774**	0.978±0.039	0.699	0.960±0.059	0.352	0.946±0.066	0.486
coil	**0.931± 0.051**	0.671	0.911±0.053	0.509	0.915±0.041	0.645	0.890±0.046	**0.689**
hear	**0.846± 0.058**	0.935	0.775±0.085	**0.938**	0.827±0.057	0.737	0.785±0.084	0.858
isol	**0.829± 0.043**	0.853	0.798±0.053	0.716	0.803±0.059	0.436	0.711±0.077	**0.884**
jaff	**0.981± 0.024**	**0.512**	0.954±0.057	0.346	0.976±0.027	0.350	0.945±0.052	0.319
vehi	**0.891± 0.047**	**0.991**	0.846±0.055	0.918	0.776±0.082	0.964	0.773±0.045	0.893
yale	**0.778± 0.103**	**0.288**	0.709±0.105	0.250	0.730±0.082	0.147	0.706±0.097	0.154

Classification accuracy is captured in terms of Area Under Curve (AUC) measure. To compute AUC, we use a SVM classifier with Gaussian kernel:

$$\mathbf{K}_{ij} = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2(a\bar{r}^2)}}, \tag{4.23}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are data samples and $\bar{r}$ is the average of pairwise distances among all the data samples and a is chosen from $[2^{-3}, 2^{-2}, 2^{-1}, 1, 2^1, 2^2, 2^3]$. The SVM trade off parameter C is chosen from $[0.01, 0.1, 1, 10, 100]$. For all data and feature selection methods, we report the best results among all the combinations of a and C. We directly use the LIBSVM [105] software in our experiments. For SOR, we further choose λ from $[0.001, 0.01, 0.1, 0.5]$ and β from $[0.001, 0.01, 0.1, 0.5]$.

We compare the average of AUC in Figure 4.3 while varying the number of features selected. We observe the AUC of SOR is clearly above most of the other methods. More specifically, among all 119 comparisons, SOR outperforms the best of the other methods in 88, tie in 17. Our method is only worse than the best of the other methods in 4 cases.

To compare the variability of the AUC, we present the average and standard deviation of the AUC when 5 features are selected in Table 5.1. For all the 6 data sets, SOR outperforms the other methods in terms of AUC.

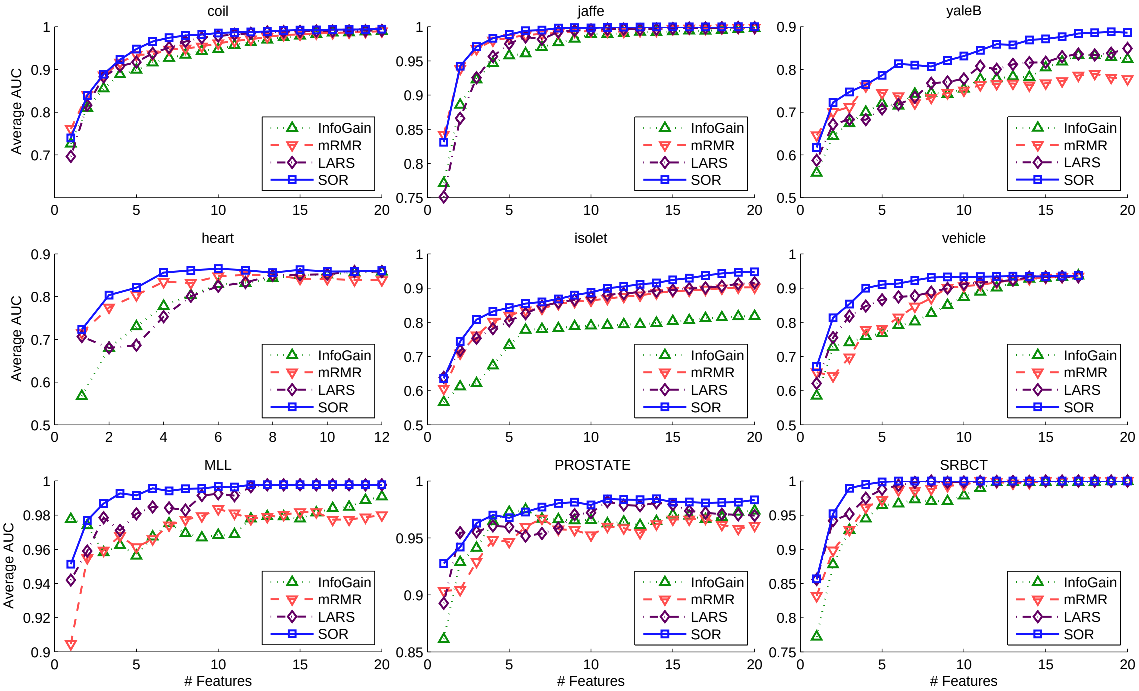


Figure 4.3. [91] AUC comparison on 6 data sets (*heart*, *vehicle*, *jaffe*, *coil*, *MLL* and *SRBCT*). .

4.5.6 Stability

We are interested in two types of stability measures: 1) Selection stability measures the overlap of selected features when we run it on different subsets of the data, and 2) Parameter stability measures how much the performance varies as we change the parameters of the algorithms.

Selection stability is defined as

$$\textbf{Stability} = \frac{1}{T(T-1)} \sum_{i=1}^{T} \sum_{j=1, j \neq i}^{T} \frac{|\mathcal{S}_i \cap \mathcal{S}_j|}{|\mathcal{S}_i \cup \mathcal{S}_j|}, \tag{4.24}$$

where T is the number of runs. Table 5.1 shows the selection stability in column "stable", in which SOR performs the best in 5 out of 6 datasets.

In terms of parameter stability, SOR requires only two parameters λ and β. We show that our method is stable to those parameters in Table 4.4, where the maximum, minimum, average, and the range of the AUC are reported. One can observe that though the parameters change dramatically in wide ranges, the AUC measure only changes about 1% – 5% for most of the data except for the heart, PROSTATE, and yaleB data sets. In our experiments, we looked into the value of λ which gives the best AUC, and we found that the typical value has a relative narrow range (around 0.1) after the normalization, indicating that λ is not a sensitive parameter.

4.5.7 Redundancy

Next we compare the redundancy of the features selected by different methods. Redundancy is measured by orthogonality between sets of selected features S:

$$\textbf{Redundancy} = \frac{1}{T(T-1)} \sum_{i,j \in \mathcal{S}, i \neq j} \frac{\mathbf{x}_i^T \mathbf{x}_j}{||\mathbf{x}_i||\, ||\mathbf{x}_j||}. \tag{4.25}$$

Table 4.4. [91] Stability to parameters of SOR. Reported are the AUCs of SOR while λ and β vary from $[0.001, 0.01, 0.1, 0.5]$

Data	min	max	average	range
MLL	0.9480	1.0000	0.9900	0.0520
PROSTATE	0.8571	0.9939	0.9514	0.1369
SRBCT	0.9789	1.0000	0.9969	0.0211
coil	0.9441	0.9836	0.9641	0.0395
heart	0.7572	0.9532	0.8514	0.1959
isolet	0.8542	0.9073	0.8734	0.0530
jaffe	0.9849	0.9993	0.9939	0.0144
vehicle	0.8912	0.9353	0.9149	0.0441
yaleB	0.6656	0.9088	0.8003	0.2432

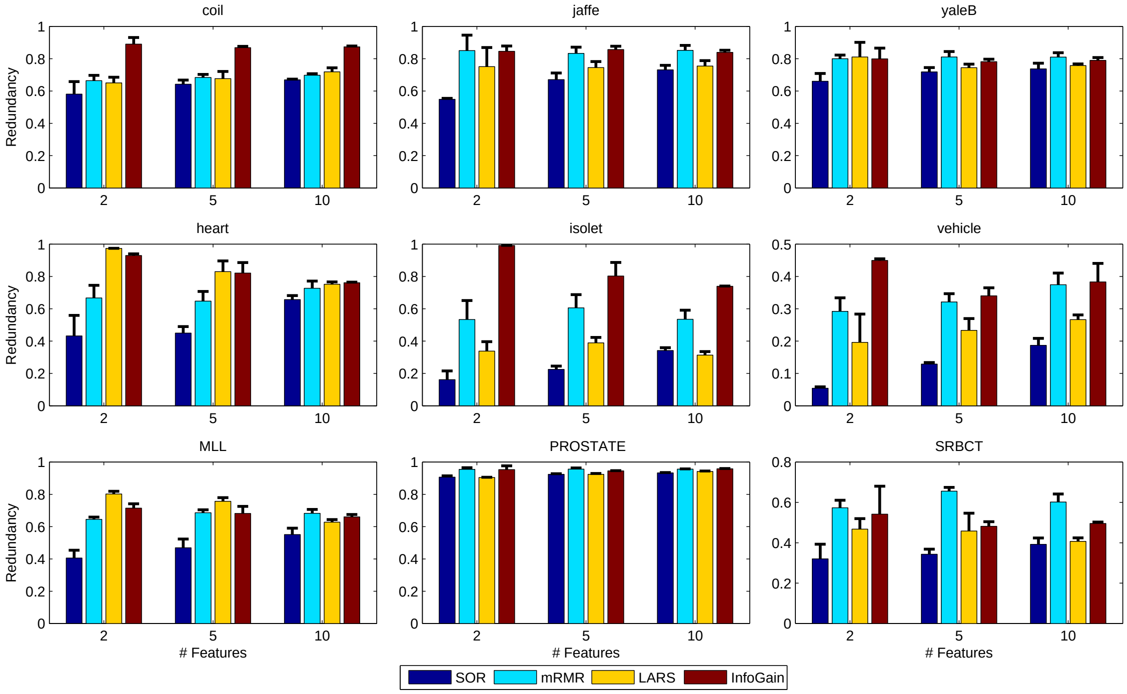


Figure 4.4. [91] Redundancy comparison of features selected by SOR, mRMR, Lars, and Information Gain (InfoGain) 6 data sets (heart, vehicle, jaffe, coil, MLL and SRBCT) .

It measures the average cosine similarity between all pairs of features. As shown in Figure 4.4, SOR clearly has the lowest redundancy in selected features across all settings. In particular, the only scalable method InfoGain performs badly with respect to this measure.

4.5.8 Applications to TCGA Data

We perform the experiments on the data described in Section 2.6.5 and apply the SOR algorithm. Both *PI3K* and *has-mir-200c* are selected but the correlation of *PI3K* and *has-mir-200c* is low. This result is consistent with multi-task learning and also indicates that low redundancy dose not destroy the results. Further more, it enhances the interpretability with higher accuracy.

CHAPTER 5

GRAPH EVOLUTION VIA SOCIAL DIFFUSION PROCESS

5.1 Introduction of Graph Clustering

Data clustering, assignment, and dimensional reduction have been the focuses for exploring unknown data [106, 107]. Among them, graph-based data analysis techniques have recently been investigated extensively in traditional machine learning problems. One reason for the popularity of graph-based approaches is the broad availability of graph data. For example, social objects (users, blog items, photos) are generated with relational links, and for objects represented in Euclidean space, one can easily obtain a graph by using similarity measurements (*e.g.* Gaussian kernels). Graph-based approaches fall into two categories. The first one is *spectral graph partitioning* methods which address the group detection problem by identifying an approximately minimal set of edges to remove from the graph to achieve a given number of groups [108, 109, 110, 111]. Impressive results have been shown in these methods which have been applied in many practical applications. These approaches relax NP-hard combinatorial problems into continuous optimization problems which can be solved by eigenvector decompositions.

Another approach category is *stochastic modeling*. In stochastic models, the observed data are assumed to be drawn from some distribution and generative assumptions [112, 113, 114, 115, 116]. These approaches often lead to a maximum likelihood problems that can be solved by Expectation Maximization (EM) or approximately Variational EM algorithms [117].

Among these models, the Chinese Restaurant Processes (CRPs) consider a sequence of customers coming to a restaurant according to the convention of Chinese people: one tends to stay in a place where there are more people. Each customer seeks some previously occupied table and the probability is proportional to the number of customers already sitting there. The new customer also sits in a new table with probability proportional to some parameter. CRP and its variations have been theoretically and empirically studied in many previous researches [115, 116, 118, 119]

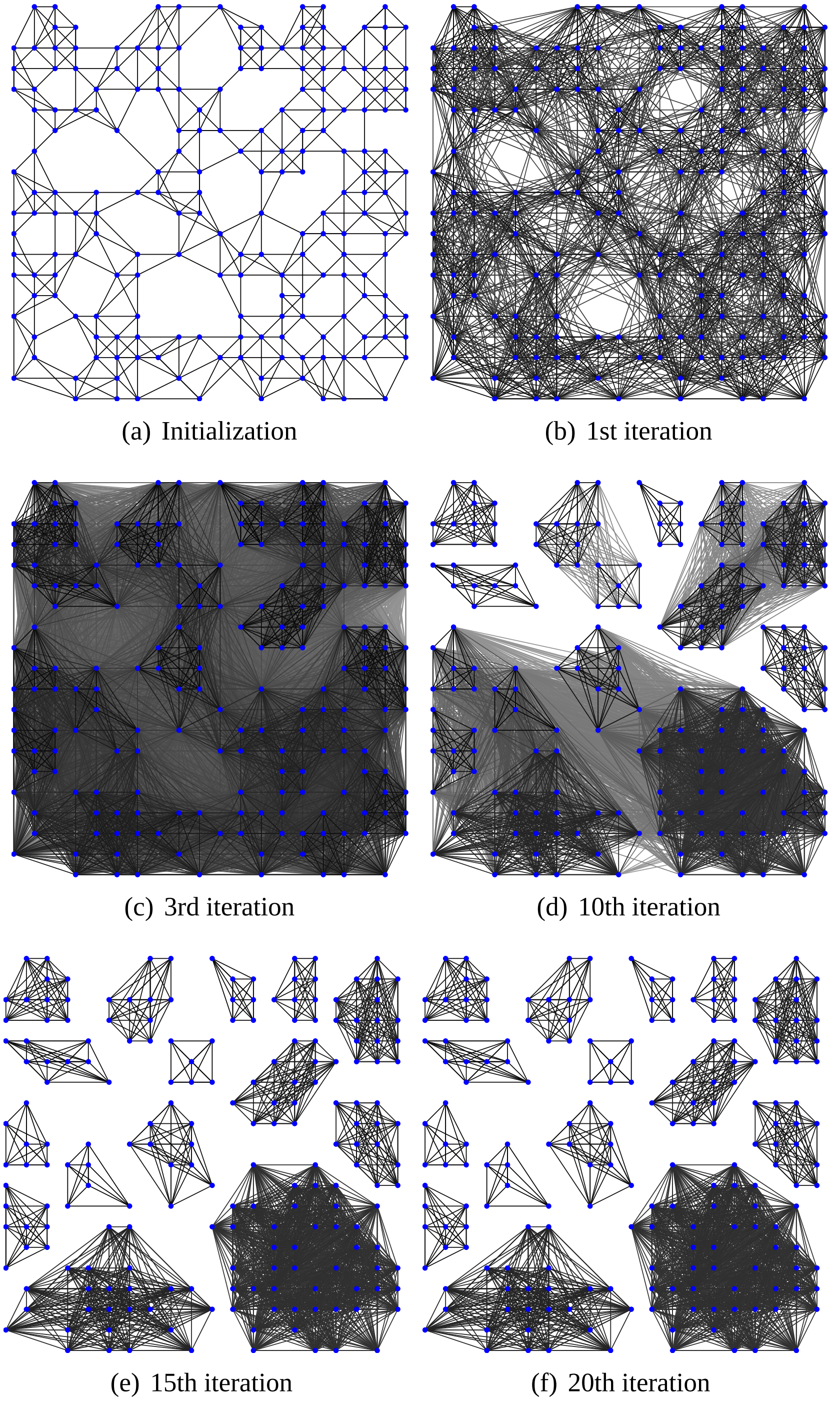
In a CRP mixture, customers are data points, and customers sitting at the same table belong to the same cluster. Since the number of occupied tables is random, the resulting posterior distribution of seating assignments provides a distribution of clusterings where the number of clusters is determined by the data.

In this chapter[1], we propose a novel stochastic process which further considers the social events among social members as a metaphor of the intrinsic stochastic process for broad range of data. We call this process as Social Diffusion Process. The basic assumption in this model is that two social members tend to communicate if they are familiar with each other or have many common friends, and that the more times they communicate, the more they are familiar with each other.

Based on our model, we derive an iterative evolution algorithm to model the social structures of the members. The major characteristic of our algorithm which differs from most of previous research is that we do not need to impose latent variables which leads to maximum likelihood estimation. Instead, our evolutionary algorithm iteratively generates a new relational graph among social members in which the social structures become more and more clear, please see Figure 5.1 for a toy example. In this example, our algorithm starts from a random binary network and ends with clearly separated subgraphs.

The similar algorithm which is closest to our intuition is Markov Clustering (MCL) [121] from the point of view of graph evolution. However, MCL is not suitable for the purpose in this chapter. We perform the MCL evolution on the same graph in Figure 5.1

[1] Most of the results in this chapter have been published in paper [120].


(a) Initialization

(b) 1st iteration

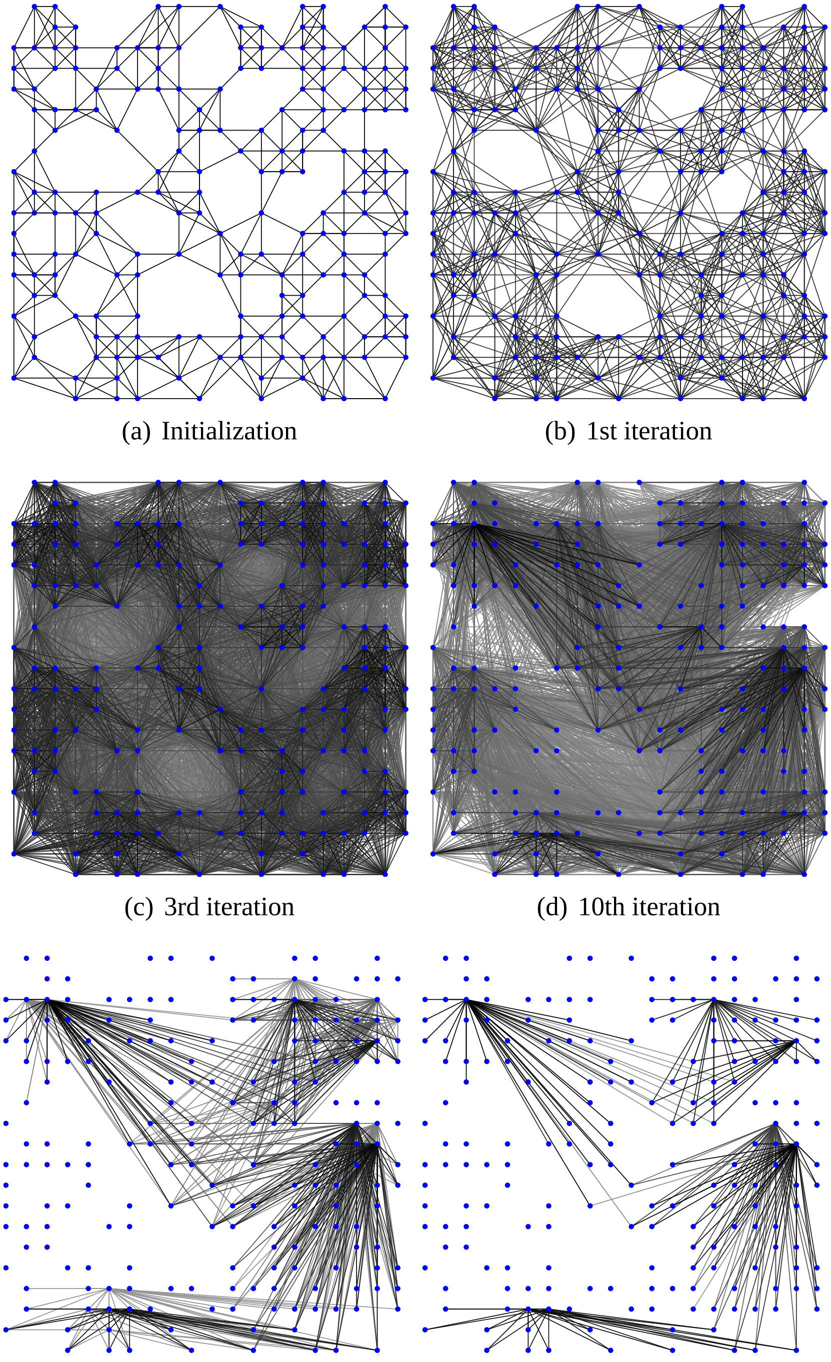
(c) 3rd iteration

(d) 10th iteration

(e) 15th iteration

(f) 20th iteration

Figure 5.1. [120] Graph evolution results on the grid toy data based on Social Diffusion Process. Each point (blue dot) represents a social member and the edge between two social members represents the familiarness between them. (a): the original graph. (b)– (f): the condensation results of the 1st, 3rd, 10th, 15th, and 20th iterations of our evolution algorithm. The darkness of the edge represents the familiarness between the social members (the darker the higher)..



(a) Initialization

(b) 1st iteration

(c) 3rd iteration

(d) 10th iteration

(e) 15th iteration

(f) 20th iteration

Figure 5.2. [120] Graph evolution results based on Markov Clustering. .

(a) and the results for MCL are demonstrated in Figure 5.2. One can observe that the result in Figure 5.1 is much more reasonable than that in Figure 5.2.

The results of the evolution algorithm can be viewed as a special case of the *Matthew effect*, in which “The rich get richer”. This is a general phenomenon in nature and societies [122, 123, 124]. One interesting observation in our algorithm is that the evolution of a graph by the SDP enhance the *qualities* of the graph in a wide range of applications. This phenomenon suggests that the SDP assumptions are natural in general. Due to the broad availability of graph-based data, our new model and algorithm have potential applications in various areas.

5.2 Social Diffusion Process for Friendship Broadening

In this section we introduce the Social Diffusion Process based on the notations of graph.

5.2.1 Preliminaries

Let $G = \{V, W\}$ denote an undirected weighted graph, where $V = \{v_1, v_2, \cdots, v_n\}$ is the set of nodes, $W \in \mathbb{R}^{n\times n}$ is a $n\times n$ matrix, and W_{ij} denotes the weight of the edge between nodes v_i and v_j. $W_{ij} = 0$, if there is no edge between v_i and v_j.

5.2.2 Social Events and Broadening of Friendship

We consider the following scenario: A and B are friends. Suppose A brings a friend A_f and meets with B. Now A_f and B become known to each other. If B also brings a friend B_f to the meeting, i.e., the four (A, A_f, B, B_f) meet. Then A_f become known to both B also B_f, i.e., the friendship circle for A_f is broadened. This happens to A, B, B_f as well.

In graph terminology, the initial friendship between A and B is represented by an edge connecting A and B. The broadened friendship between A_f and B (assuming they are not connected at initial stage) has a connection strength somewhere between 0 and 1. In other words, if two persons C and D don't know each other, the existence of a mutual friend connects C and D. Further more, even if A and B are friends (i.e., an edge exists between A and B), their friendship is further enhanced due to the existence of mutual friends. Our main goal is to formally define this friendship broadening process and compute the *friendship enhancement probability*. We expect this enhanced friendship provide a more clear social community structure as shown in Figure 1.

Formally, we define the following events among social members: (1) $Date(v_i, v_j)$: v_i and v_j initial a dating. (2) $Bring(v_i, v_k)$: v_i brings v_k after the event $Date(v_i, v_j)$ for some j. (3) $Meet(v_p, v_q)$: v_p and v_q meet in the same table.

We further impose the following rules: (1) If $Date(v_i, v_j)$ happens, $Meet(v_i, v_j)$ happens, or (2) If $Date(v_i, v_j)$ and $Bring(v_i, v_k)$ happen, $Meet(v_k, v_j)$ happens. (3)If $Date(v_i, v_j)$, $Bring(v_i, v_k)$, and $Bring(v_j, v_l)$ happen, $Meet(v_j, v_l)$ happens.

Here we assume $Date(v_i, v_j)$ is equivalent to $Date(v_j, v_i)$ and $Meet(v_k, v_l)$ is equivalent to $Meet(v_l, v_k)$.

We use the following to denote the rules above

$$\text{Rule 1:} \quad Date(v_i, v_j) \Rightarrow Meet(v_i, v_j) \tag{5.1}$$

$$\text{Rule 2:} \quad \left.\begin{array}{l} Date(v_i, v_j) \\ Bring(v_i, v_k) \end{array}\right\} \Rightarrow Meet(v_j, v_k) \tag{5.2}$$

$$\text{Rule 3:} \quad \left.\begin{array}{l} Date(v_i, v_j) \\ Bring(v_i, v_k) \\ Bring(v_j, v_l) \end{array}\right\} \Rightarrow Meet(v_k, v_l) \tag{5.3}$$

5.2.3 Social Diffusion Process

Now we are ready to introduce the Social Diffusion Process. The process starts with a graph $G = \{V, W\}$ where $V = \{v_1, v_2, \cdots, v_n\}$ denotes a set of social members and W denotes the familiarness between social members, *i.e.* W_{ij} represents the familiarness between v_i and v_j, $i, j = 1, 2, \cdots, n$. We assume that $W_{ij} = W_{ji}$. The SDP happens as following,

(1) Choose a threshold $t \sim U(0, \mu)$ where $\mu = \max_{ij} W_{ij}$ and U denotes the uniform distribution.

(2) $Date(v_i, v_j)$ happens with a constant probability δ if $W_{ij} \geq t$.

(3) $Bring(v_i, v_k)$ and $Bring(v_j, v_l)$ happen with probability $p(i, k, t)$, $p(j, l, t)$, respectively, where

$$p(i,k,t) = \begin{cases} \frac{1}{|\mathcal{N}_{i,t}|} & \text{if } v_k \in \mathcal{N}_{i,t} \\ 0 & \text{otherwise} \end{cases},$$

$$p(j,l,t) = \begin{cases} \frac{1}{|\mathcal{N}_{j,t}|} & \text{if } v_k \in \mathcal{N}_{j,t} \\ 0 & \text{otherwise} \end{cases},$$

$\mathcal{N}_{i,t} = \{q : W_{iq} \geq t\}, \mathcal{N}_{j,t} = \{q : W_{jq} \geq t\}$, and $|\cdot|$ denotes the cardinality of the set.

(4) Apply rules (1)–(3). For any p, q, if $Meet(v_p, v_q)$, $W_{pq} \leftarrow W_{pq} + \alpha\mu$.

The threshold t can be interpreted as the importance of the dating event. Two friends do not date if they are not familiar with each other enough (thresholded by t)[2]. When a social member brings some friend, he/she only considers those friends who are familiar

[2] The reason why we use a thresholding of W_{ij} instead of directly using W_{ij} for event $Date(v_i, v_j)$ is following. Assume we want to date with some one on the wedding of Royal wedding for William and Kate, who are we going to date? Probably one of our most important friends. In the same event, if we want to bring guest to meet our friend in the date, who are we going to bring? Probably another one of our most important friends. In reality, social events happen according to their importance, denoted as threshold t in the chapter. We believe this model is much accurate than directly using W_{ij} as the probability of $Date(v_i, v_j)$.

enough with (thresholded by t). The set $\mathcal{N}_{i,t}$ is the friends the social member v_i can bring with this threshold t. Eq. (5.4) indicates that social member v_i chooses friends in $\mathcal{N}_{i,t}$ with uniform distribution. Notice that there are two parameters in this model δ and α. In section 3, we will introduce an algorithm based on the SDP, in which the two parameters can be eliminated by natural normalization.

5.3 Graph Evolution Based on Social Diffusion Process

5.3.1 The Evolution Algorithm

We first denote A^t as the following

$$(A^t)_{ij} = \begin{cases} 1 & \text{if } W_{ij} \geq t \\ 0 & \text{otherwise} \end{cases} \tag{5.4}$$

where t is a positive threshold. Consider two social members v_i and v_j. The events in which they meet each other can be divided into three cases:

Case (1). $Date(v_i, v_j)$. In this case the probability that they meet is

$$P(Meet(v_i, v_j)) = \delta(A^t)_{ij}.$$

Case (2). $Date(v_i, v_k)$ and $Bring(v_k, v_j)$. By definition $|\mathcal{N}_{k,t}| = \sum_j A^t_{jk} = d^t_k$, where d^t_k is the degree k in A^t. In this case,

$$\begin{aligned} & P(Meet(v_i, v_j)) \\ = & \sum_k P(Meet(v_i, v_j)|Date(v_i, v_k), Bring(v_k, v_j)) \\ = & \sum_k \delta(A^t)_{ik} \frac{A^t_{jk}}{d_k} = \delta(A^t D^{-1} A^t)_{ij}, \end{aligned}$$

where $D = \mathbf{diag}(d_1, d_2, \cdots, d_n)$.

Case(3). $Date(v_k, v_l)$, $Bring(v_k, v_i)$, and $Bring(v_l, v_j)$. Similar with case (2), we have

$$\begin{aligned} P(Meet(v_i, v_j)) &= \sum_{kl} \delta(A^t)_{kl} \frac{A^t_{ik}}{d_k} \frac{A^t_{jl}}{d_l} \\ &= \delta(A^t D^{-1} A^t D^{-1} A^t)_{ij}. \end{aligned}$$

By summing up the three cases, we have

$$\begin{aligned} & P(Meet(v_i, v_j)) \\ = \ & \delta A^t_{ij} + \delta(A^t D^{-1} A^t)_{ij} + \delta(A^t D^{-1} A^t D^{-1} A^t)_{ij}. \end{aligned}$$

From the definition of updating of W, we have

$$\begin{aligned} & \mathbb{E}(\Delta W_{ij}) \\ =& \alpha\mu\delta \left(A^t_{ij} + (A^t D^{-1} A^t)_{ij} + (A^t D^{-1} A^t D^{-1} A^t)_{ij} \right) \\ \triangleq& \alpha\mu\delta M^t_{ij}. \end{aligned} \tag{5.5}$$

Here $A^t_{ij} + (A^t D^{-1} A^t)_{ij} + (A^t D^{-1} A^t D^{-1} A^t)_{ij}$ is denoted by M^t_{ij}. This suggests that the expectation $\mathbb{E}(\Delta W_{ij})$ is proportional to M^t_{ij}. In our implementation we normalize M^t_{ij} by $M^t_{ij} \leftarrow M^t_{ij} / \sum_{i'j'} M^t_{i'j'}$, which leads to the following algorithm,

Algorithm 4 $\tilde{W}$ = **GraphEvolution**(W)

```
**Input**: Graph $W$
**Output:** Graph $\tilde{W}$
$\mu = \max_{ij} W_{ij}, \tilde{W} = \mathbf{0}$
**for** $i = 1 : T$ **do**
    $t = i\mu/T$
    Calculate $M^t$ using Eq. (5.5)
    Normalize $M^t : M^t_{ij} \leftarrow M^t_{ij}/\sum_{i'j'} M^t_{i'j'}$
    $\tilde{W} \leftarrow \tilde{W} + M^t$
**end for**
**Output:** $\tilde{W}$
```

In this algorithm, we use an evenly distributed threshold t to approximate the uniform distribution from which t should be drawn from. In our experiments, we set $T = 50$. One should notice that no matter what the choice of the normalization is, the algorithm has the following properties.

Property 5.3.1 *The result of* GraphEvolution *is scale invariant,* i.e. $\forall \beta > 0$,

$$GraphEvolution(W) = GraphEvolution(\beta W).$$

This is because the threshold t is always evenly distributed in the interval $[0, \max_{ij} W_{ij}]$ and M^t remains the same. In other words, the choice of the normalization does not change any terms in M^t.

Property 5.3.2 *If W is a set of disconnected full cliques with same size and same weight,* i.e. *there is a partition* $\Pi = \{\pi_1, \pi_2, \cdots, \pi_K\}, \pi_k \cap \pi_l = \Phi, 1 \le k, l \le K, \cup_k \pi_k = \{v_1, v_2, \cdots, v_n\}$ *such that* $\forall i, j \in \pi_k, W_{ij} = c$ *where c is a constant, and* $\forall i \in \pi_k, j \in \pi_l, k \neq l, W_{ij} = 0$*, then*

$$W \propto \textbf{GraphEvolution}(W).$$

This is easy to show since if W is a set of disconnected full cliques with the same weight, A^t is the same for every $t : A^t_{ij} = 1$ if $A_{ij} \neq 0$, $A^t_{ij} = 0$ otherwise. Thus $M^t \propto W$, which leads to $W \propto GraphEvolution(W)$. This property shows a hint of conditions in which the algorithm of $W \leftarrow GraphEvolution(W)$ converges, which will be discussed later.

5.3.2 Application of Graph Evolution

The algorithm *GraphEvolution* can be used in different purposes. The basic idea is that it improves the quality in terms of the natural structure underlying the graph data. In this chapter, we investigate two applications: clustering and semi-supervised learning.

For the purpose of clustering, one can simply iteratively perform the following

$$W \leftarrow GraphEvolution(W). \tag{5.6}$$

As iterations continue, the structures of the graph is clearer and clearer. We show results of the evolution algorithm on a toy grid data, see Figure 5.1.

In this example, we randomly generate 198 points in a 20×20 grid. We obtain an unweighted graph as follows. If node i is one of K-nearest neighbors of node j, or node j is one of the K-nearest neighbors of node i, we set $W_{ij} = 1$, and $W_{ij} = 0$ otherwise. $K = 7$ in this example and the neighborhood is computed using the Euclidean distance of the nodes on the 2-dimensional grid coordinate. The original graph is shown in Figure 5.2(a).

Starting from this graph, we run the *GraphEvolution* algorithm for 20 iterations and the results of the first, third, 10th, 15th, and 20th iterations are shown in Figure 5.1 (b)–(d). In the third iteration (Figure 5.2(c)), the structure of the data is observable. In the 10th iteration (Figure 5.2(d)), the structure is even more clear. Finally, in the 20th iteration, (Figure 5.2(f)), the clusters are completely separated.

After the graph evolution iterations, the cluster structure encoded in the edge weight matrix is usually obvious to human. In practice, the number of clusters discovered by the algorithm is different from expected number of clusters. We use the following partition scheme to reach a desired number of cluster. We run algorithm in Eq. (5.6) until there are two disconnected subgraphs. Then pick up the subgraph which has a large number nodes to run algorithm in Eq. (5.6), and do the same strategy until we reach a specified number of clusters.

For the purpose of semi-supervised learning, we just use $\tilde{W} = GraphEvolution(W)$ as preprocessing, where W is the input of and $\tilde{W}$ is the output. Instead of performing semi-supervised learning on W, we do it on $\tilde{W}$. We show that the qualities of the $\tilde{W}$ are much higher than W.

5.4 Experimental Results

In this section, we first demonstrate the convergence of algorithm and then show experimental evidence of the quality improvement by apply our graph evolution algorithm. In the clustering comparison, we specify the number of clusters. However, in a microRNA pattern discovery application, we run our algorithm until convergence and let the algorithm determine the number of clusters.

5.4.1 Convergence Analysis

We first demonstrate the convergence of our algorithm on a toy data, which is a 9×9 binary graph, shown in the left most panel of the bottom row of Figure 5.3. There are two cliques in this graph: nodes 1–4 and nodes 5–9. We add some noise by setting $W_{13} = W_{58} = W_{79} = 0$ and $W_{45} = 1$. We run algorithm in Eq. (5.6) for 30 iterations. One can observe that our algorithm converges fast and at the convergent graph, all edges within the same clique have the same value. Also as highlighted in Figure 5.3, the noise values of W_{13}, W_{58}, W_{79}, and W_{45} are corrected by our algorithm.

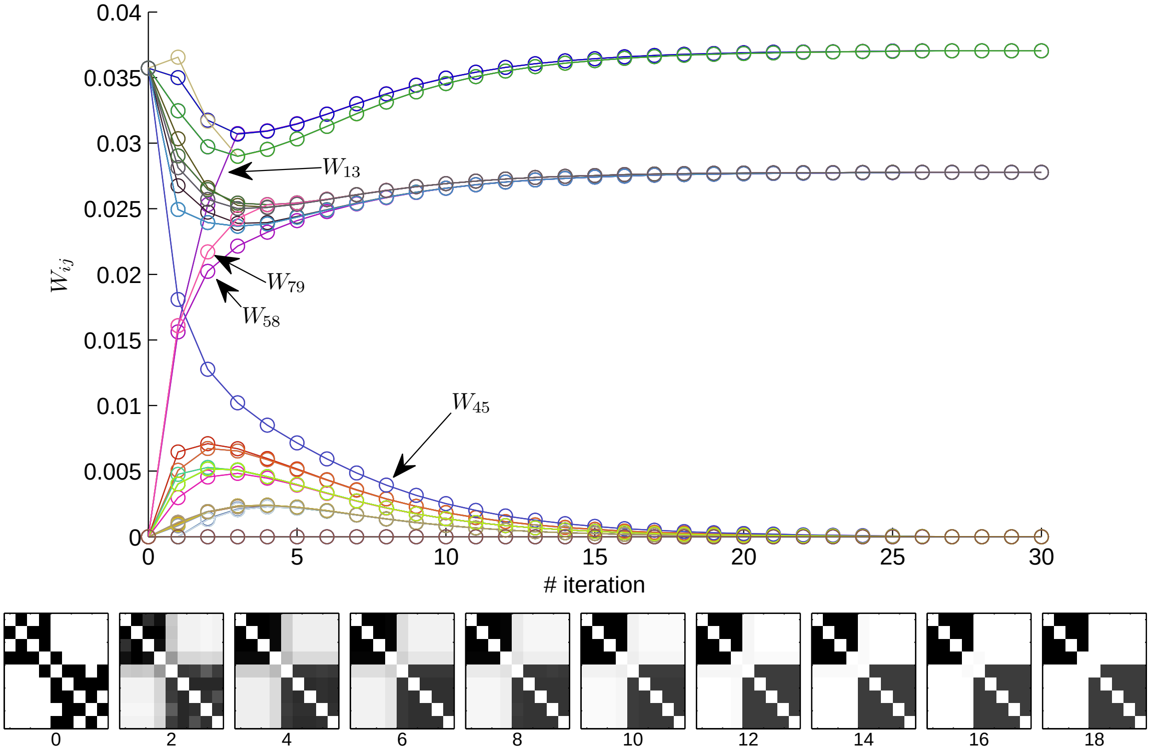


Figure 5.3. [120] Convergence curves and adjacency matrix of our algorithm on a 9×9 toy data. The left most panel of the bottom row is the initial binary graph (black represents 1 and white represents 0) and the rest of the bottom row is the evolution result of 2nd, 4th, $\cdots$, 18th iterations. Initially, nodes 1–4 is a pseudo-clique, as well as nodes 5–9. $W_{13} = W_{58} = W_{79} = 0$ and $W_{45} = 1$. After around 18 iterations, the two cliques become separated and the nodes within the two cliques become full connected. The top panel show the convergence of all the elements in W. Highlighted are the values of W_{13}, W_{58}, W_{79}, and W_{45}, which are corrected by our algorithm. .

Table 5.1. [120] Accuracy, normalized mutual information (NMI), and purity comparison of K-mean (Km), Spectral Clustering (SC), Normalized Cut (Ncut), and Graph Evolution (GE). Both Spectral Clustering and Normalized Cut are achieved by tuning the graph construction parameters and the best results are reported.

	Accuracy				**NMI**				**Purity**			
	Km	SC	Ncut	GE	Km	SC	Ncut	GE	Km	SC	Ncut	GE
UMI	0.458	0.471	0.498	**0.644**	0.641	0.646	0.649	**0.763**	0.494	0.505	0.505	**0.667**
COI	0.570	0.614	0.792	**0.839**	0.734	0.750	0.860	**0.879**	0.623	0.658	0.817	**0.840**
ION	0.707	0.702	0.684	**0.880**	0.123	0.193	0.107	**0.446**	0.707	0.730	0.684	**0.880**
JAF	0.744	0.799	0.965	**0.967**	0.809	0.849	0.959	**0.962**	0.774	0.819	0.965	**0.967**
MNI	0.687	0.713	0.820	**0.833**	0.690	0.698	0.748	**0.769**	0.705	0.733	0.820	**0.833**
ORL	0.582	0.683	0.756	**0.775**	0.786	0.834	0.866	**0.891**	0.624	0.713	0.773	**0.802**
PR1	0.716	0.675	0.562	**0.899**	0.129	0.176	0.102	**0.458**	0.726	0.757	0.708	**0.899**
PR2	0.580	0.566	0.569	**0.706**	0.019	0.017	0.013	**0.136**	0.580	0.566	0.569	**0.706**
SOY	0.908	0.871	**1.000**	**1.000**	0.903	0.859	**1.000**	**1.000**	0.924	0.893	**1.000**	**1.000**
SRB	0.480	0.622	**0.699**	0.639	0.232	0.411	**0.454**	0.421	0.512	0.645	**0.699**	0.639
YEA	0.132	0.327	0.302	**0.395**	0.013	0.129	0.126	**0.231**	0.328	0.430	0.436	**0.540**
ZOO	0.264	0.674	0.629	**0.723**	0.116	0.615	0.570	**0.751**	0.423	0.750	0.737	**0.871**
AML	0.688	0.678	0.659	**0.847**	0.100	0.100	0.073	**0.394**	0.696	0.692	0.666	**0.847**
CAR	0.623	0.729	0.719	**0.799**	0.655	0.743	0.738	**0.779**	0.691	0.789	0.788	**0.822**
WIN	0.961	0.936	0.978	**0.983**	0.863	0.845	0.907	**0.926**	0.961	0.943	0.978	**0.983**
LEU	0.879	0.840	0.958	**0.972**	0.559	0.513	0.735	**0.806**	0.879	0.860	0.958	**0.972**
LUN	0.663	0.672	**0.748**	0.704	0.495	0.485	**0.547**	0.473	0.864	0.860	**0.911**	0.828
DER	0.766	0.848	0.955	**0.964**	0.838	0.818	0.905	**0.931**	0.853	0.876	0.955	**0.964**
ECO	0.552	0.496	0.505	**0.631**	0.467	0.458	0.487	**0.549**	0.739	0.770	0.808	**0.851**
GLA	0.452	0.446	0.453	**0.565**	0.320	0.298	0.333	**0.399**	0.549	0.572	**0.652**	0.650
GLI	0.585	0.548	0.559	**0.700**	0.465	0.410	0.398	**0.505**	0.619	0.569	0.601	**0.700**
IRI	0.802	0.746	0.843	**0.953**	0.640	0.514	0.655	**0.849**	0.815	0.758	0.843	**0.953**
MAL	0.911	0.731	0.902	**0.929**	0.569	0.299	0.544	**0.624**	0.911	0.743	0.902	**0.929**
MLL	0.669	0.637	0.687	**0.861**	0.435	0.376	0.426	**0.681**	0.692	0.651	0.687	**0.861**

5.4.2 Clustering

In this experiment, we extensively compare our algorithm with standard clustering algorithms (K-means, Spectral Clustering, Normalized Cut[3]) in 20 data sets. These data

[3] We also compared with MCL. However the accuracies are much (more than 10%) lower than all the method we compare here. We believe MCL is not suitable for the purpose in this chapter. One can find visual evidence in Figure 5.2.

sets come from a wide range of domains, including gene expressions including gene expressions (PR1,SRB, LEU, LUN, DER, AML, GLI, MAL, MLL), images (ORL, UMI, COI, JAF, MNI) and other standard UCI data sets (ION, PR2, SOY, ECO, GLA, YEA, ZOO, CAR, WIN, IRI) [4]. We use accuracy, normalized mutual information (NMI) and purity as the measurement of the clustering qualities and the results are shown in Table 5.1. Our method achieves the best results in 22 out of 24 data sets. Here notice that for Spectral Clustering and Normalized Cut, we tune the graph construction parameters. More explicitly the graph is constructed as $W_{ij} = \exp\left(-\|x_i - x_j\|^2/(\gamma\overline{r}^2)\right)$ where $\overline{r}$ denotes the average pairwise Euclidean distances among the data points and γ is chosen from $[2^{-2}, 2^{-1}, \cdots, 2^5]$ and the best results are reported.

5.4.3 Semi-supervised Learning

We first run graph evolution algorithm (Eq. (5.6)) for one iteration. After that we use the result weights as input to run Zhu *et al.*'s [125] (marked as HF in the Figure 5.5) and Zhou *et al.*'s [126] (marked as GC) approaches. We compare four methods, HF, GC, HF on resulting graph (HF_GE), GC on resulting graph (GC_GE), on four face image datasets. We tested the methods on AT&T[5], BinAlpha [6], JAFFE[7], and Sheffield [8] data sets. For all the methods and datasets, we randomly select N labeled images for each class, $N = 1, 2, 3, 4, 5$, and use the rest as unlabled images. We try 50 random selections for each dataset, and computer the average of the semi-supervised classification accuracy.

The results are shown in Figure 5.5.

[4] All the mentioned data can be downloaded at parchive.ics.uci.edu/ml/ or csie.ntu.edu.tw/ cjlin/.

[5] http://people.cs.uchicago.edu/~dinoj/vis/ORL.zip

[6] http://www.cs.toronto.edu/~roweis/data.html

[7] http://www.cs.toronto.edu/~roweis/data.html

[8] http://www.shef.ac.uk/eee/vie/face.tar.gz

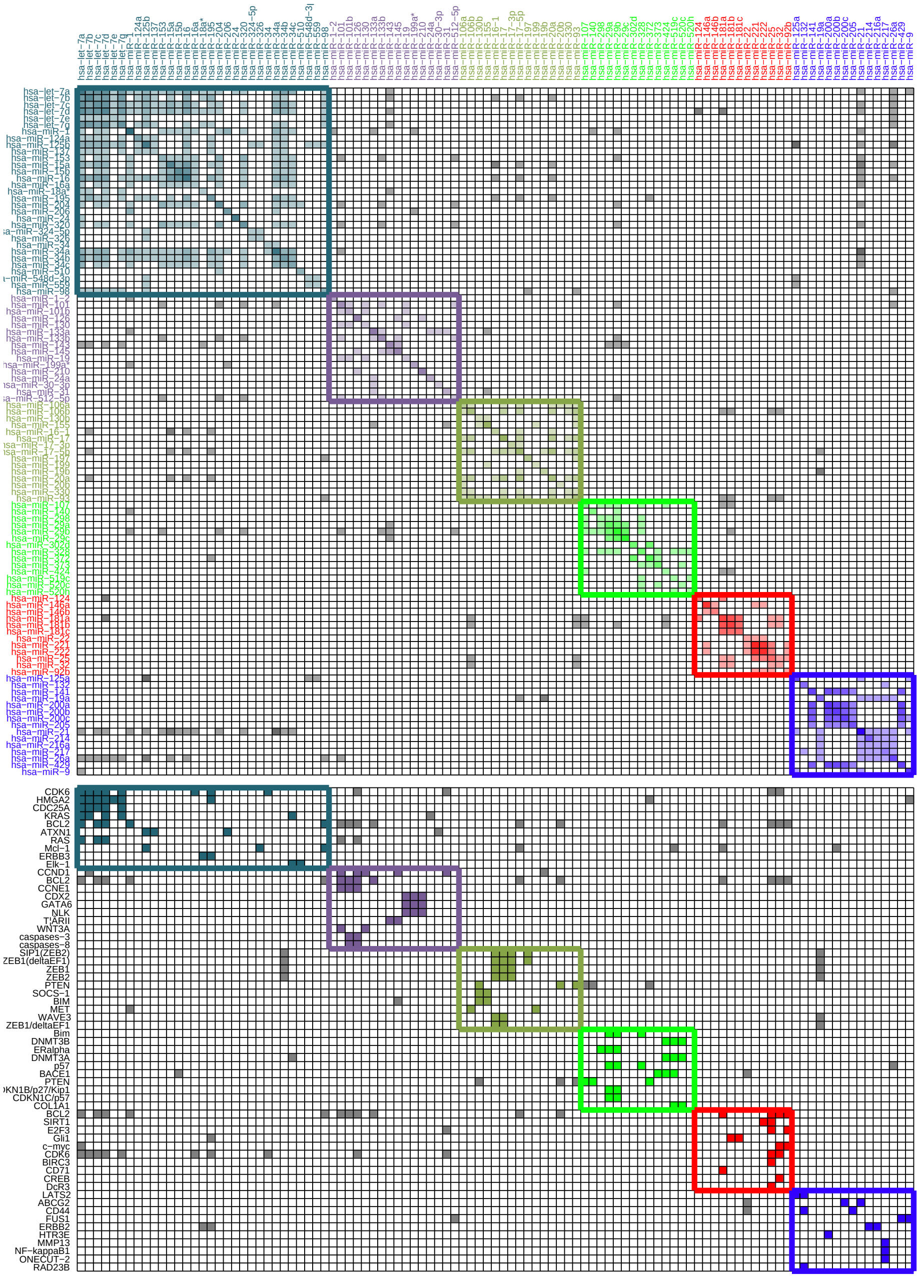


Figure 5.4. [120] 6 miRNA cliques found by Graph Evolution. Top panel is the miRNA graph in which the values denotes the number of common targeting genes of two miRNAs. The bottom panel is the top 10 targeting genes for each clique. The cliques are separated by different colors. The left top part of the top panel is the *let-7* miRNA family and the right bottom part of the top panel is the *hsa-mir-200* family. .

In all these case, we always obtain higher classification accuracy by applying graph condensation. For datasets BinAlpha, JAFFE, and Sheffield, our methods are consistently 5%–10% better than the standard semi-supervised learning methods.

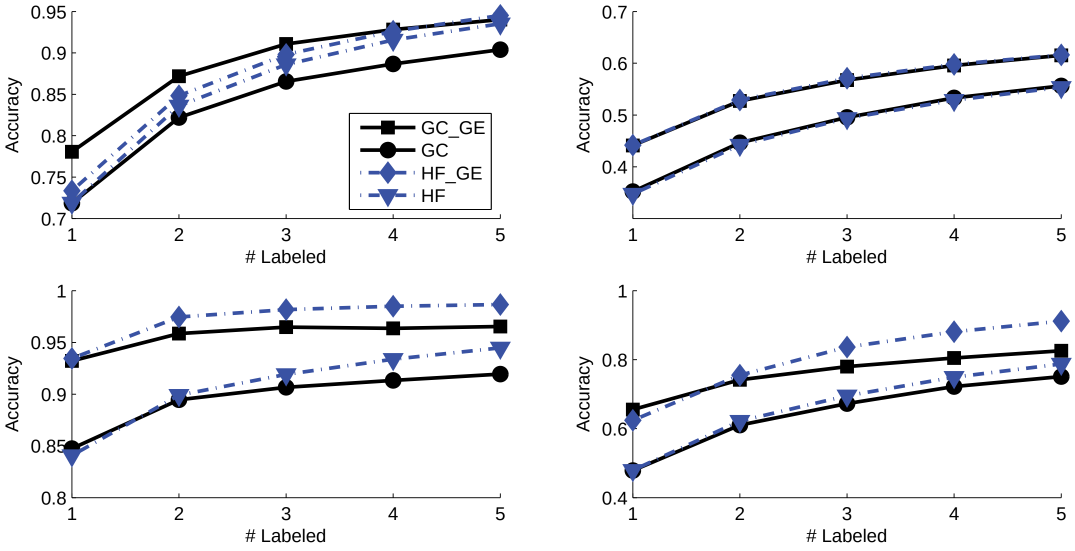


Figure 5.5. [120] Semi-supervised learning on 4 datasets(from left to right): AT&T, BinAlpha, JAFFE, and Sheffield datasets. Classification accuracies are shown for four methods: HF, GC, HF using condensated graph (HF_GE), GC using condensated graph (GC_GE). For each dataset, number of labeled data per class are set to 1, 2, 3, 4, 5. Using the graph evolution consistently improves over original methods. .

5.4.4 Graph Evolution for microRNA Functionality Analysis

In this experiment, we are interested in the interaction network between microRNAs (miRNAs) and genes. MiRNAs play important regulatory roles by targeting messenger RNAs (mRNAs) for degradation or translational repression, and have become one of the focuses of post-transcriptional gene regulation in animals and plants[127, 128, 129] and have been an active research topic in various domains [130, 131, 132, 133]. A database of verified miRNA/target gene relationship can be found in [134]. Here we apply our algorithm to investigate the relationships between the miRNAs and the genes. The main purpose is to discover new interaction patterns in the miRNA regulatory network.

We use the data with version of Nov. 6, 2010. We use the number of targeting genes as the weights of two miRNAs, *i.e.* $W_{ij} = \sum_k B_{ik}B_{jk}$ where $B_{ik} = 1$ indicates miRNA i targets gene k, $B_{ik} = 0$ otherwise. We select the largest disconnected component which has 103 miRNAs and run the *GraphEvolution* algorithm until converges. Finally, we discover 6 separated subgroups of miRNAs, which are shown in Figure 6.6. The following is the outline of our discovery in this experiment. (1) the *let-7* [135, 136] miRNA family is correctly clustered into the same group. (2) The *hsa-mir-200* family are highly connected with each other, which is not reported in literature so far.

CHAPTER 6

REGULATORY ELEMENTS VISUALIZATION

6.1 Background and Motivation

Regulatory elements such as MicroRNAs (miRNAs) are important components in the cell processes. MiRNAs are small non-coding RNAs of 20~22 nucleotides, which were first discovered in [137], and now have been found to be present and highly conserved among species [138]. Like other regulatory RNAs, miRNAs are generally involved in post-transcriptional gene regulation, which control the eukaryotic gene expression by reducing the protein yield from specific target mRNAs. MiRNA genes are synthesized in the nucleus as a double stranded precursor, which is processed by two enzymes, Drosha and Pasha, into a precursor (pre)-miRNA, then exported to the cytoplasm by exportin 5 [139, 140]. Once the pre-miRNA reaches the cytoplasm, it is cleaved by Dicer into a ~22 nt long functional mature miRNA. The mature miRNA can then assemble into a ribonucleoprotein complex known as the RNA-induced silencing complexes to participate in RNA interference [141]. Recent studies indicate that miRNAs may be essential in biological processes, such as cell growth, cell proliferation, tissue differentiation, embryonic development, apoptosis, and cellular signaling networks[142].

MiRNAs have attracted exponentially more research interests in recent years. One of the main reasons is that miRNAs have been discovered to be involved in disease regulations, playing the role of targeting key mRNAs in disease pathways. For example, Cimmino *et al.* [143] showed that both *miR-15a* and *miR-16-1* negatively regulate *BCL2* at a post-transcriptional level, which induces apoptosis in a leukemic cell line model. Similar mechanisms are found in many other cases [144, 129, 145, 128, 146, 147, 148, 149].

Though the over all mechanisms remains unclear, studies have linked miRNAs to several important types of diseases, such as cancers [150, 151, 152], heart diseases [153, 154] which strongly suggests that miRNAs could be useful as diagnostic and prognostic markers [155, 156, 157, 158, 159, 160, 161, 162, 163], and even novel therapy approaches [153, 164, 165, 166, 167].

However, more and more studies indicate that the targeting patterns between miRNAs and mRNAs are complicated. First, one miRNA can target a large number (up to thousands) of mRNAs [168, 169]. Second, on the other hand, multiple miRNAs are found to work synergistically to control individual genes. For example, *lin-4* and *let-7* are cooperative and are the earliest miRNA pair to be experimentally verified [170]. Krek *et al*. (9) also demonstrated that *miR-375*, *miR-124* and *let-7b* jointly regulate *Mtpn*, providing evidence for coordinate miRNA control [169]. Further more, it was demonstrated that the majority of all human genes are modulated by miRNAs [171, 127].

Though these discoveries offer deeper insights of disease regulation and open a wide direction on diagnostic and therapy, they also bring a challenging problem in the analysis of genes, disease, and miRNAs as a whole network. Obviously the independent study of miRNAs, their targets, and the related diseases do no suffice in fully understanding the their functions and in exploring other potential unknown mechanisms. The challenge here is how to incorporate the known evidences to establish a big picture. This chapter provides visualization tool for this purpose using *in-silico* analysis of publicly available data.

To be more specific, we offer a global view of of miRNAs by visualize all miRNAs in a single shot. The basic idea is to incorporate simultaneously all the directing targeting relationship (local relationship) and obtain a global visualization of the miRNAs. The visualization results visually answer questions like the following, (1) does one miRNA function similarly to another miRNA? (2) does one miRNA function differently from another miRNA? (3) do a set of miRNAs function as a group? We establish a novel visualization system based on solely computational targeting predictions, and interestingly, our visualization results are verified by a series of experimental studies from other investigators.

By providing a big picture of the whole interaction network, our visualization tool helps miRNA research in the several ways. (1) Discovery of miRNA complexes. By miRNA complex, we mean a group of miRNAs which function similarly to each other in the whole regulatory network. Similar to protein complexes [172, 173, 174, 175, 176, 177, 178], miRNA complexes play important roles in functional analysis. However, in contrast to proteins, miRNAs do not directly interact with each other; instead, they interacts with mRNAs. Further more, the interaction networks among miRNAs and targeting mRNAs are complicated, a visualization tool is essential at the beginning of the analysis. Surprisingly, we discover several miRNA complexes, two of which have been verified by independent research groups and two of which of which are not reported yet. (2) To verify results of biological experiments. We highlight 234 miRNAs which are verified to function in AML (Acute Myeloid Leukemia), prostate cancer, lung cancer, breast cancer, and ovarian cancer in the global visualization of 711 miRNAs of human beings using a single picture. We discover that miRNAs are often close to each other if they have similar function(s). And based on these observation, our visualization tool (3) provides a reasonable range if miRNAs on which researchers should focus. The visualization tool systematically offers a series of candidates for some specific diseases. Instead of blindly testing the functions of all the miRNAs, the visualization tool helps to narrow down the search range to some candidates. (4) We also predict miRNA regulatory candidate of the five diseases by combining the causally verified miRNAs and unverified miRNAs near them.

Our techniques can also be employed in other networks for other purposes. As an example, we build a miRNA predictor by considering the global information of miRNAs. To be specific, we combine local and global miRNA structures to establish detect novel miRNAs. By validating in human miRNAs, we show that our predictor is accurate, robust, and stable. We also apply our predictor in *D. Melanogaster* and successfully discovery 30 novel miRNAs, 14 of which are conserved in other species.

6.2 Results and Discussions

6.2.1 Notations

In order to present the results in a more convenient way, we first introduce a set of notations, details of which will be given in the “Materials” sections.

Data point. By *data point*, we mean the objects we are interested in, which means miRNAs in the whole chapter if there is no further explanation. Without confusion, we also call a data point an *object*.

Embedding. In order to obtain a visualization of data points, we compute a Euclidian coordinate system for all the data points from some non-Euclidian system, for example, a graph in this chapter. The resulting Euclidian coordinate system is called *embedding space* in which each object is represented as data point in Euclidian coordinates (3-dimensional Euclidian coordinates in our chapter).

Graph. A *graph* in this chapter is a weighted graph, in which each vertex represent a miRNA and the weights of the edges represent the similarity between miRNAs which the corresponding edges connect. In this chapter, we use $\mathbf{W}$ to represent a weight graph, where $\mathbf{W}_{ij}$ represents the weight between object i and j, $i, j = 1, 2, \cdots, n$ and n is the number of objects we consider.

Bipartite graph. Bipartite graph is a special graph, in which there are two disjoint group of vertices. In this chapter, the two groups are miRNAs and mRNAs and the edge between a miRNA and a mRNA represent that the miRNA directly interacts with the mRNA. In the whole chapter, we use a matrix $\mathbf{B} \in \mathbb{R}^{m \times n}$ to represent the bipartite graph, where m

and n are the number of miRNA and mRNA, respectively, and $\mathbf{B}_{ij} = 1$ if miRNA i directly interact with mRNA $j, i = 1, \cdots, m$ and $j = 1, \cdots, n$.

Local information. By *local information* of objects we mean the direct interaction or similarity of objects. For example, consider two miRNAs (say **a** and **b**). The the local information of the might mean **a** and **b** interact with 10 mRNAs in common. We say *local* because the information does no change no matter we consider the other miRNAs or not. This is in contrast to *global information*, which consider all the objects as a whole. Consider a local resistor network, where resistors connect among nodes. The local information for this network are the individual resistors, connecting pairs of nodes, which are independent to each other. And the global information here is the effective electric resistance between nodes, considering all the resistors together. The main purpose of this chapter is to incorporate the individual and local information to derive global information and obtain a bigger picture which demonstrates the *effective* relationships among objects we consider.

Distance profile. In our analysis, use the distance profile to estimate the how close a pair of miRNAs are to each other in the embedding space. Consider a pair of miRNAs m_1 and m_2. We use four numbers $d(x\%, a, b)$ to represent the distance profile. By "distance between m_1 and m_2 is $d(x\%, a, b)$" we mean: (1) the Euclidean distance of m_1 and m_2 in the embedding space is d, (2) there are x% of the miRNA pairs have closer distance than d, (3) m_2 is the a-th nearest neighbor of m_1, and (4), m_1 is the b-th nearest neighbor of m_2 in the embedding space, using the Euclidean distance. Obviously, the smaller d is, the closer the pair of miRNAs are to each other. However, we do not know the scale of the distances, we use the other three numbers of the distance ranking to represent the relative distance. *Sequence alignment profile* is the same except that the neighborhoods are computed using the sequence alignment score.

Naming system. In the whole chapter, the miRNAs represent the *mature miRNAs* and without other notations, they mean miRNAs in *Homo sapiens*. For example, by *miR-*

21a, we mean the mature miRNA *hsa-miR-21a*. For those miRNAs which come from different precursors but have identical mature sequence, we consider them a unique object. For example three separate precursors in different locations, *let-7a-1*, *let-7a-2* and *let-7a-3*, produce the mature *let-7a* sequence.

6.2.2 Embedding Results of Whole *Homo sapiens* miRNAs

We use new computational tools, the Green's functions with the corresponding Dirichlet Boundary Conditions, to incorporate local information and derive a global embedding coordinates of 711 miRNAs of *Homo sapiens*. We compute the embedding from a weighted graph **W** where the weights represents the number of common mRNAs two miRNAs interact with. To be more precise, we use the following to compute **W**:

$$\mathbf{W}_{ij} = \sum_{k=1}^{K} \mathbf{B}_{ik}\mathbf{B}_{jk}, \tag{6.1}$$

where **B** is a bipartite graph by considering the interactions between 711 miRNAs and 21199 mRNAs. Details can be found in the "Materials" section. Notice that interactions between miRNAs and mRNAs are derive from computational tools by only considering the sequences of the miRNAs and mRNAs, which means that we can still make use of the visualization when we have no prior knowledge from experimental results. We will show that the embedding results are consistent with the experimental results done independently by other researchers.

The embedding results are shown in Figure 6.1. Each sphere represents one miRNA. We color the miRNAs according to their functions, *i.e.* whether they are involved in one or some of the 5 diseases (Acute Myeloid Leukemia, prostate cancer, lung cancer, breast cancer, and ovarian cancer). Since one miRNA can be involved in multiple diseases, there

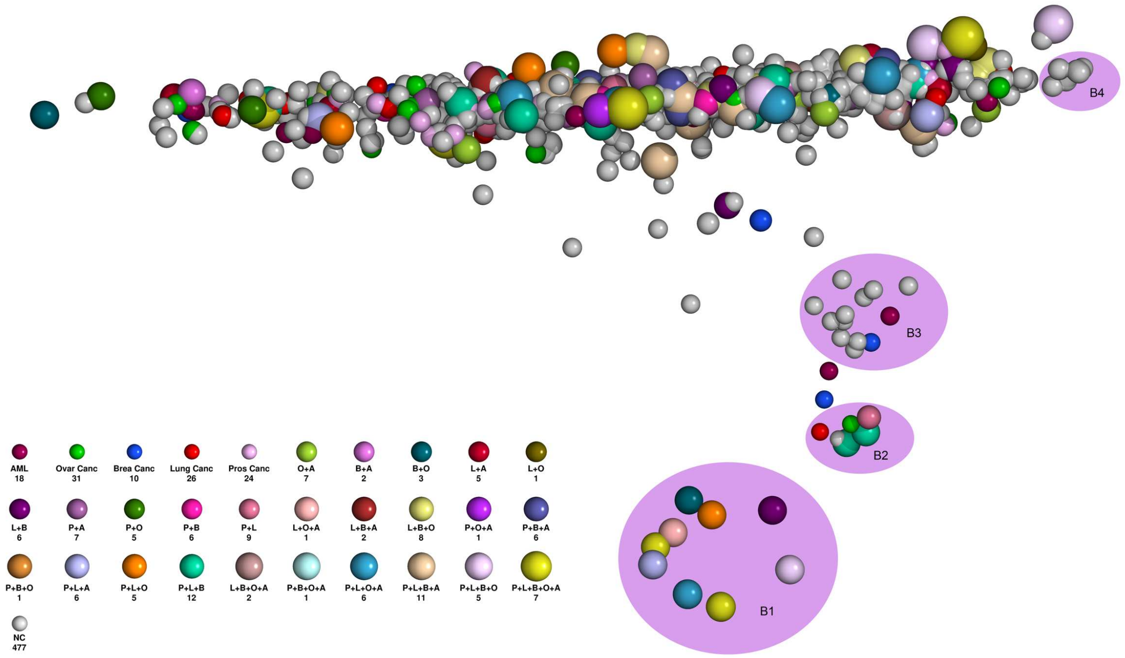


Figure 6.1. 3D embedding visualization results for 711 miRNAs of *H. sapiens*. Each sphere represents one miRNAs. The color of MiRNAs represent the functions in 5 diseases, *i.e.* whether they are involved in the diseases. AML: Acute Myeloid Leukemia; Ovar Canc: ovarian cancer; Brea Canc: breast cancer; Lung Canc: lung cancer; Pros Cans: prostate cancer. For those miRNAs which are involved in multiple diseases, we use "+" to combine the diseases, *e.g. O+A* means the miRNAs are involved in both Ovarian cancer and AML. "NC" represent *Not Classified*, *i.e.* not involved in any of the 5 diseases. The size of the sphere represents the number of diseases the miRNA is involved. The embedding results are derived from sequence of miRNAs and the functions are verified by causal biological experiments or microarray experiments. .

are $2^5 = 32$ possible configurations. The configuration of *Breast Cancer* + *Ovarian Cancer* + *AML* is not found, thus there are total 31 configurations.

We also list the distributions of the miRNAs over 23 chromosomes in Figure 6.2.

Notice that the embedding coordinates are obtained only using the sequence information and the functions of miRNAs (the coloring) are obtained by biological experiments. In the rest of this section, we will introduce the properties of the embedding result and the usage of the visualization in miRNA research.

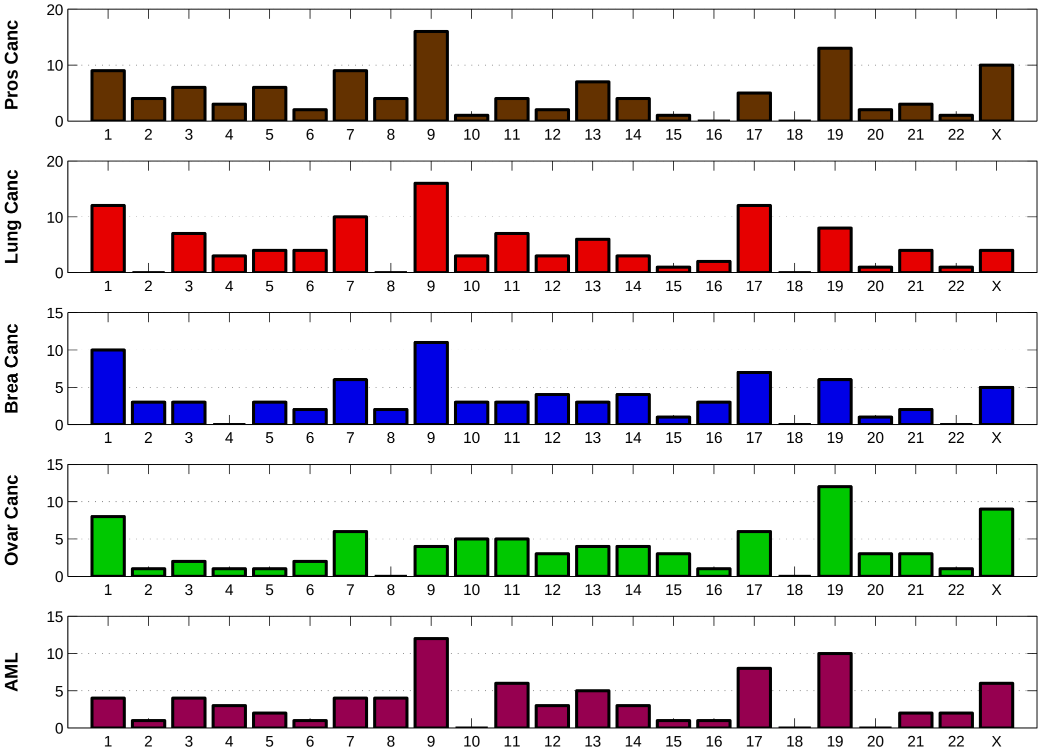


Figure 6.2. Distribution of the miRNAs over 23 chromosomes for prostate cancer, lung cancer, breast cancer, ovarian cancer, and AML (Acute Myeloid Leukemia)..

6.2.3 Over All Observations of The Visualization Result

We first highlight some of the observations. Then introduce other discovery from the visualization in details later.

If a pair of miRNAs have the involved in the same disease, they are often embedded together. Thus the embedding is consistent with the experiment validation, which suggests that it is useful to investigate the visualization to obtain further analysis. Some examples will be shown later in the section.

If two miRNAs are very similar in sequence, they are often embedded in a small distance. But the other way does not hold, i.e. *if two miRNAs are embedded ia a small distance, they are not necessarily similar in sequence.* We can see this observation in Figure 6.3. The reason is that the sequence similarity only capture the local information of miRNAs, and our embedding considers all the possible relationships of miRNAs and the embedding distance reflect the *effective* functional relationship among miRNAs. For example, assume that two miRNAs are both involved in a key pathway in some disease, but the sequences of the two miRNAs are not necessarily similar, however, the *effective* functional similarity should be high, and the embedding distance reflects such functional similarity. We demonstrate the difference between embedding distances and sequence similarity in Figure 2 in which one can observe serval issues. (1) They show that miRNAs which have high number in common target mRNAs are often embedded in small distances. (2) If two miRNAs are very similar in sequence, they are often embedded in a small distance. But the other way does not hold. (3) If two miRNAs are very similar to each other, say the BLAST score is higher than 29, then the number of common target mRNAs must be high and the embedding distance must be small.

These observations also suggest that *the embedding distance reflects the functional similarity more accurately than the sequence similarity.* We will demonstrate this effect using more examples later in the section.

In Figure 6.1, we also see that there are several groups of miRNAs (*B1* – *B4*) in which miRNAs are close to each other and far away from miRNAs outside the group. Two of the four groups have been been well investigated. According to our functional analysis, we believe the other two groups, which have not been reported yet, are equivalently important in studies of miRNAs.

6.2.4 Four Functional Groups of Human miRNAs

We first introduce the four functional groups in human miRNAs, the members of which are listed in Table 6.1.

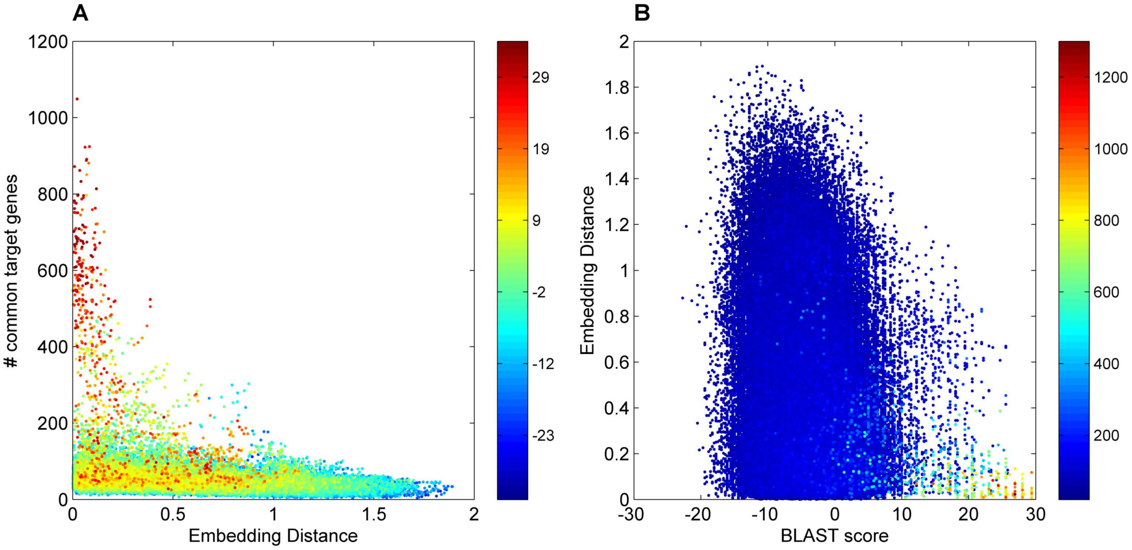


Figure 6.3. A demonstration of the difference between embedding distance and sequence similarity. For each pair of miRNAs, we compute three values: the embedding distance, the sequence similarity, and the number of putative mRNAs they interact in common. The embedding distance are the Euclidean distance of some pair of miRNAs in the embedding space, the sequence similarity is measured by BLAST score of the given pair of mature miRNAs, and the number of common putative target mRNAs is computed using Eq. (6.1). In both plots (*(A)* and *(B)*), each dot represents a pair of miRNAs. In (A), we plot the dots using embedding distance and the number of common putative target mRNAs as x-axis and y-axis, respectively, and color the dots with sequence similarity. In (B), we plot the dots using sequence similarity and embedding distance as x-axis and y-axis, respectively, and color the dots with the number of common putative target mRNAs. (A) shows that miRNAs which have high number in common target mRNAs are often embedded in small distances. (B) indicate that if two miRNAs are very similar in sequence, they are often embedded in a small distance. But if two miRNAs are embedded ia a small distance, they are not necessarily similar in sequence. Both of the plots suggest that if two miRNAs are very similar to each other, say the BLAST score is higher than 29, then the number of common target mRNAs is high and the embedding distance is small. .

6.2.4.1 *B1*: *Let-7/miR-98* family

In our visualization, the *let-7/miR-98* family includes 9 miRNAs: *hsa-let-7a*, *hsa-let-7b*, *hsa-let-7c* , *hsa-let-7d*, *hsa-let-7e*, *hsa-let-7f*, *hsa-let-7g*, *hsa-let-7i*, and *hsa-miR-98*. Members *let-7/miR-98* family are found to express in late mammalian embryonic development [179, 180]. Instead of studying the *let-7/miR-98* family case by case [179, 181, 180],

Table 6.1. Members of the four miRNAs groups found by visualization. The embedding results of the four group are highlighted in Figure 6.1.

Group	miRNA members	Mature sequence	Chromosome
	hsa-let-7a	UGAGGUAGUAGGUUGUAUAGUU	9, 11, 22
	hsa-let-7b	UGAGGUAGUAGGUUGUAUGGUU	22
	hsa-let-7c	UGAGGUAGUAGGUUGUAUGGUU	21
	hsa-let-7d	AGAGGUAGUAGGUUGCAUAGUU	9
	hsa-let-7e	UGAGGUAGGAGGUUGUAUAGUU	19
	hsa-let-7f	UGAGGUAGUAGAUUGUAUAGUU	9, X
	hsa-let-7g	UGAGGUAGUAGUUUGUACAGUU	3
	hsa-let-7i	UGAGGUAGUAGUUUGUGCUGUU	12
	hsa-miR-98	UGAGGUAGUAAGUUGUAUUGUU	X
Group B2	_hsa-miR-106a_	AAAAGUGCUUACAGUGCAGGUAG	X
	hsa-miR-106b	UAAAGUGCUGACAGUGCAGAU	7
	hsa-miR-17	CAAAGUGCUUACAGUGCAGGUAG	13
	hsa-miR-20a	UAAAGUGCUUAUAGUGCAGGUAG	13
	hsa-miR-20b	CAAAGUGCUCAUAGUGCAGGUAG	X
	hsa-miR-93	CAAAGUGCUGUUCGUGCAGGUAG	7
	hsa-miR-302a	UAAGUGCUUCCAUGUUUUGGUGA	4
	hsa-miR-302b	UAAGUGCUUCCAUGUUUUAGUAG	4
	hsa-miR-302c	UAAGUGCUUCCAUGUUUCAGUGG	4
	hsa-miR-302d	UAAGUGCUUCCAUGUUUGAGUGU	4
	hsa-miR-519d	CAAAGUGCCUCCCUUUAGAGUG	19
	hsa-miR-520a-3p	AAAGUGCUUCCCUUUGGACUGU	19
	hsa-miR-520b	AAAGUGCUUCCUUUUAGAGGG	19
	hsa-miR-520c-3p	AAAGUGCUUCCUUUUAGAGGGU	19
	hsa-miR-520d-3p	AAAGUGCUUCUCUUUGGUGGGU	19
	hsa-miR-520e	AAAGUGCUUCCUUUUUGAGGG	19
	hsa-miR-520g	ACAAAGUGCUUCCCUUUAGAGUGU	19
	hsa-miR-520h	ACAAAGUGCUUCCCUUUAGAGU	19
	hsa-miR-526b*	GAAAGUGCUUCCUUUUAGAGGC	19
Group B4	_hsa-miR-374a_	UUAUAAUACAACCUGAUAAGUG	X
	hsa-miR-374b	AUAUAAUACAACCUGCUAAGUG	X
	hsa-miR-548a-5p	CAAAACUGGCAAUUACUUUUGC	6
	hsa-miR-548b-5p	AAAAGUAAUUGUGGUUUUGGCC	6
	hsa-miR-548c-5p	AAAAGUAAUUGCGGUUUUUGCC	12
	hsa-miR-548d-5p	AAAAGUAAUUGUGGUUUUUGCC	8

in this chapter, we utilize the our visualization results and establish more comprehensive understanding of this family.

We first summarize the verified targeting genes of the *let-7/miR-98* family in Figure 6.6 in which we show the cross analysis of the *let-7/miR-98* family and *miR-106/miR-20* family. Here we focus on the *let-7/miR-98* family, and the *miR-106/miR-20* family will be discussed later. In the left panel of the figure, the color represents the number of common targeting genes of pairs of miRNAs. One can see that between the two families, the interaction is weak, while they strongly interact with each other within the same family. The targeting genes of the *let-7/miR-98* family include *HMGA2*, *CDC25A, CDK6, KRAS, BCL2, RAS, BFNF, Cdc34,* and *FUS1*. Notice that HMGA2 (High-mobility group AT-hook 2) itself is a transcriptional regulating factor the 3' UTR of which has seven conserved sites complementary to the members of *let-7/miR-98* family [182].

6.2.4.2 *B2*: *MiR-106/miR-20* family

The *miR-106/miR-20* family in our study includes 6 miRNAs: *hsa-miR-106a, hsa-miR-106b*, *hsa-miR-17*, *hsa-miR-20a*, *hsa-miR-20b*, and *hsa-miR-93*. Their targeting relationships are also shown in Figure 6.6. Their targeting mRNAs include *E2F2*, *p21*, *CDKN1A*, *Mylip*, *PCAF, APP, BMPR2*, *CCL1*, and *FBX031*.

The *MiR-106/miR-20* family comes from three paralog groups *mir-17* miRNA clusters[183], locating in Chromosome 7, 13, and X, which are shown in Figure 6.7. MiRNA cluster is a set of miRNAs which are located very close to each in chromosome (often within one thousand nt), and are often transcribed together as polycistronic primary transcripts and are then processed into multiple individual mature miRNAs. [184, 185]. The genomic organization of these miRNA clusters is often highly conserved, suggesting an important role for coordinated regulation and function.

The polycistron *miR-25, miR-93*, and *miR-106b* are located at Chromosome 7, within intron 13 of the minichromosome maintenance protein 7 (*MCM7*) gene on chromosome 7q22.1, see Figure 6.7 (*A*). Among them, *miR-93* and *miR-106b* are in *Mir-106/mir-20* family we discovered using our visualization tool, which is marked by "*@B2*" in the figure. The second cluster includes *miR-17, miR-18a, miR-19a, miR-10a, miR-19b-1,* and *miR-92a-1*, among which *miR-17* and *miR-20a* are in the *Mir-106/mir-20* family. This cluster is located at intron 3 of open reading frame (ORF) 25 in Chromosome 13 (C13orf25). The third cluster is located at Chromosome X q26.2, which includes *miR-393, miR-92a-2, miR-19b-2, miR-20b, miR-18b*, and *miR-106a,* among which *miR-20b* and *miR-106a* are in the *Mir-106/mir-20* family.

The local embedding region of *Mir-106/miR-20* family in Figure 6.1 is highlighted in Figure 6.7 (*B*). The clear separation of the *Mir-106/mir-20* family from all the other miRNAs suggests that they might play some particular functions which are different from the other miRNAs. The seed sequences, shown Figure 6.7 (*C*), partially supports that their functions in targeting miRNAs, pathways, and disease might be similar. We will discuss more about this miRNA family in the "Discussion" section.

6.2.4.3 *B3*: *miR-302/miR-502* family

The *miR-302/miR-502* family we found in our visualization tool includes 13 miRNAs which are listed in Table 6.1. For this group of miRNAs, several evidences are discovered by previous researches. *miR-302d* is found to be involved in AML [186], and Huang *et al.* found that *miR-520c* is causally involved in breast cancer and that *miR-520c* targets *CD44* and promotes tumor invasion and metastasis [187]. To *et al.* discovered that *miR-520h* targets *ABCG2* [188]. Li *et al.* found that *miR-302d* targets *KLF13, MBNL2,* and *TRPS1.*

Though the function mechanisms are reported less clearly by literacy comparing with the previous two groups (*B1* and *B2*), we can clearly see that the members of *miR-302/miR-*

502 family are close to each other and are far away from other miRNAs in the embedding results.

The cluster of *miR-302a, miR-302b, miR-302c, miR-302d* are located as a cluster in Chromosome 4 and the *miR-520* cluster is located in the Chromosome 19. Interestingly, *miR-302a, miR-302b, miR-302c, miR-302d, miR-519b, miR-519c, miR-520a, miR-520b, miR-520c, miR-520d*, and *miR-520e* have a consensus seed sequence: *AAGUGC*, and the were reported to be simultaneously highly expressed in undifferentiated human embryonic stem cells [189].

6.2.4.4 *B4*: *miR-374/miR-548* family

miR-374/miR-548 family includes 6 mature miRNAs: *miR-374a, miR-374b, miR-548a-5p, miR-548b-5p, miR-548c-5p,* and *miR-548d-5p*. The functional targets of this family are less well explored. Yet, Mees *et al.* showed evidence that *miR-374a/b* potentially target *E1A* binding gene p300, or *EP300* [190]. Piriyapongsa suggested that the family of *mir-548* are derived from *Made1* transposable elements [191].

6.2.5 Functional Analysis With Global Embedding

Here we combine the visualization results and the biological experiments done by other researches together and study interesting cases one by one. We investigate the miRNAs which are close to each other and which are both verified to involved in some same disease. Notice that the visualization results do not rely the biological experiments. Thus the analysis can be done before any biological experiments. For example, if miRNA *a1* is verified to be involved in disease *A*, and we find that miRNA *a2* is close to *a1*, then according to our functional analysis, *a2* is hypothetically related to disease *A*. There have been many miRNAs which are causally verified to be involved in various diseases [1], however,

[1] http://www.mir2disease.org/

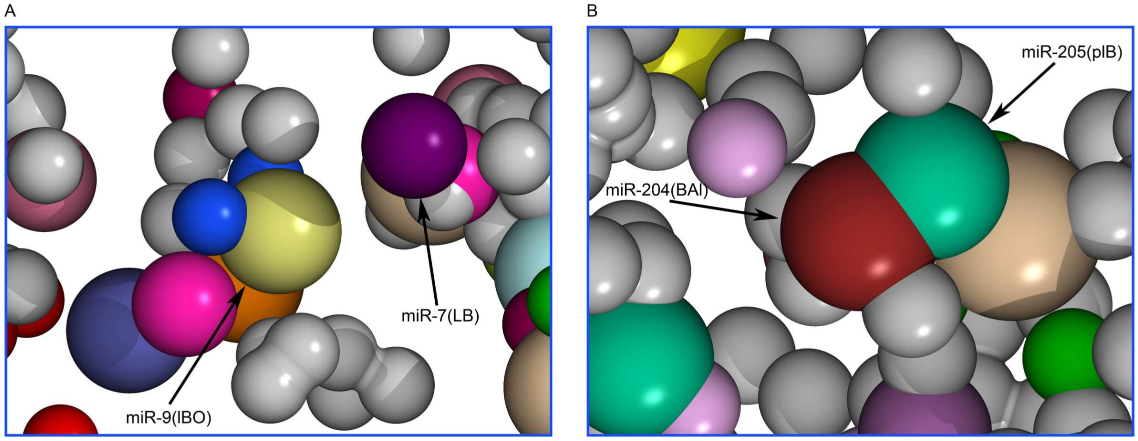


Figure 6.4. Two cases in which the embedding distance of the visualization is close and the sequences are dissimilar in alignment score. *A*: *hsa-miR-7* and *hsa-miR-9*. The symbols in parentheses represent the function of the corresponding miRNA, which uses the same notations as Figure 6.7. The embedding distance profile of this pair is $0.0286(0.2\%, 2, 5)$ which indicates they are close to each other. Their sequence alignment similarity profile is $-3.8824(45.87\%, 360, 336)$ indicating that they are far away from each other in sequence similarity. These tow miRNAs are both causally verified to be involved in breast cancer. *B*: *hsa-miR204* and *hsa-miR-205*. The embedding distance and sequence alignment similarity profile of this pair are $0.0250(0.14\%, 3, 2)$ and $-0.2773(18.13\%, 80, 98)$, respectively. .

there are also many other cases which are not yet explored. The usage of our tool is to narrow the list of candidate miRNAs in our biological studies.

In the section, we (1) first test the cases in which both miRNAs are causally verified and are close to each other in our visualization, and then (2) highlight cases in which miRNA is causally verified and one is verified by large scale micro array studies. The first test is to show that miRNAs which are close to each other often share the same functions. The second task is to offer a set of potential miRNA candidates which we can focus on in biological experiment design.

6.2.5.1 Causal miRNAs

Here we test the consistency of the visualization results and the biological experiments for 5 diseases: AML, prostate cancer, lung cancer, breast cancer, and ovarian cancer.

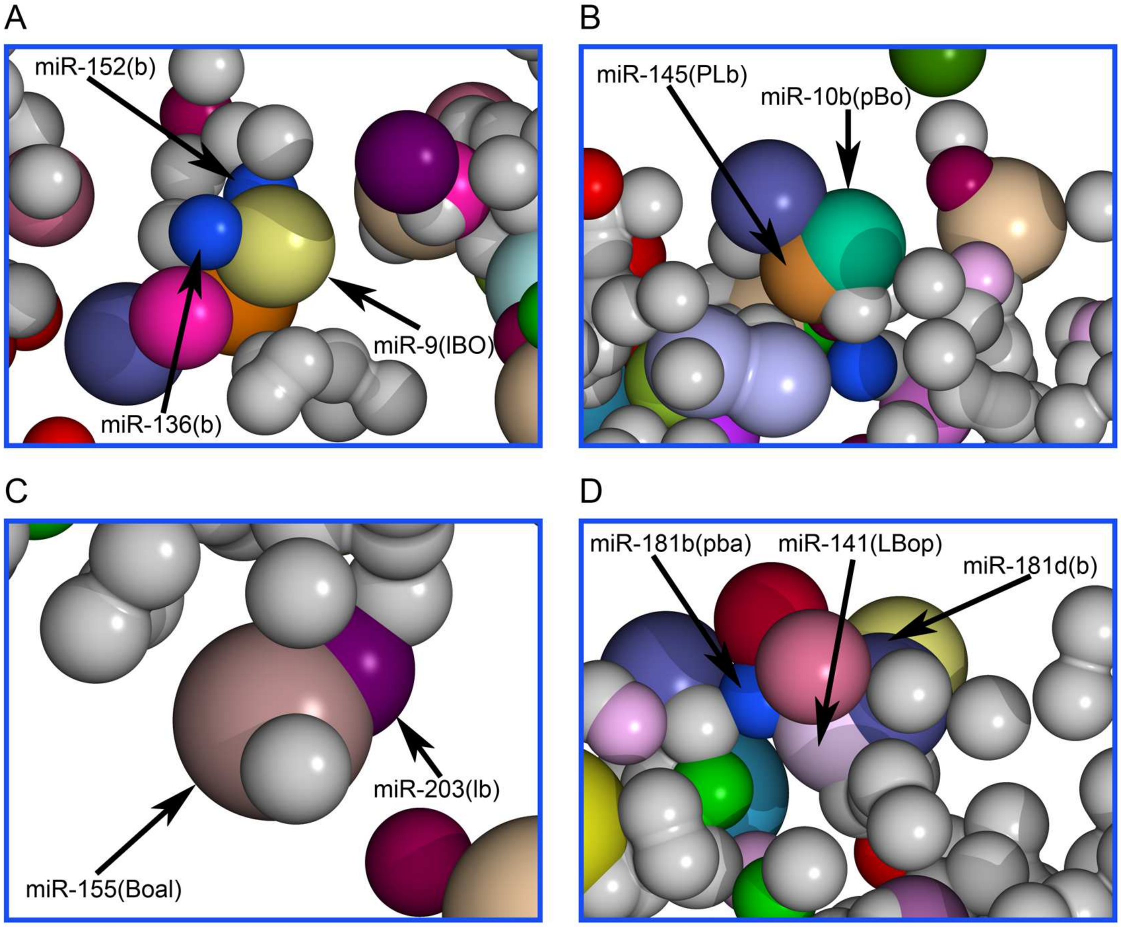


Figure 6.5. MiRNAs which are hypothetically involved in diseases. *A*: *hsa-miR-9* is verified to be involved in breast cancer by causal experiments. The nearby miRNAs *hsa-miR-152b* and *hsa-miR-136* have been identified to be correlated to breast cancer by micro array experiments. *B*: *hsa-miR-145* causally verified for prostate cancer, while *hsa-miR-10b* is verified by micro array experiments. *C*: *hsa-miR-155* causally verified for breast cancer, while *hsa-miR-203* is verified by micro array experiments. *D*: *hsa-miR-141* causally verified for breast cancer, while *hsa-miR-181b* and *hsa-miR-181d* are verified by micro array experiments..

In Table 6.2 we list 20 pairs of miRNAs which are both causally verified by biological experiments. Listed are the name of the pairs of miRNAs, the references, the embedding distance, and the sequence BLAST score.

For the “Embedding distance” column, the numbers of a pair of miRNAs are listed as $d(x\%, a, b)$ where d is the embedding distance in our visualization result, $x\%$ means that the distance this pair of miRNAs ranks bottom $x\%$ among all the possible pairs of miRNAs, a and b are the ranks of the second, first miRNA to the first, second miRNA, respectively. For example, in the first item in Table 6.2, $d = 0.0298$ means that the Euclidean distance of miRNAs *hsa-miR-200a* and *hsa-miR-141* is 0.0298 in our visualization result. But we do not know whether this is a small number, we use three relative numbers to show that they are close to each other. $x = 0.23\%$ means that there are only 0.23% of the pairs of miRNAs are closer than this pair out of $711 \times (711 - 1)/2 = 273705$ pairs of miRNAs. $a = 4$ means that *hsa-miR-200a* has 710 neighbors among which *hsa-miR-141* is the 4-th closest to *hsa-miR-200a*. $b = 4$ means that *hsa-miR-141* has 710 neighbors among which *hsa-miR-200a* is the 4-th closest to *hsa-miR-141*.

For the “Sequence Alignment Score” column, we use the same notations, except that the ranking is sorted using descend order, because the higher Alignment score is, the closer the sequences are. The sequence alignment score is computed by MATLAB function *nwalign*. Details can be found in the “Materials and methods” section.

Among the 20 pairs of miRNAs, 11 pairs were discovered in the same chapter. For example Gibbons *et al.* discovered *hsa-miR-200a* and *hsa-miR-141* were critically involved in lung cancer [192]. The reason they simultaneously tested these two miRNAs might be that they are very similar to each other (0.03%, 1,1). The cases are similar for the other 10 pairs: *hsa-miR-200b/ hsa-miR-429* [192], *hsa-miR-200b/ hsa-miR-200c* [192], *hsa-miR-20a/ hsa-miR-17* [193],*hsa-miR-29b/ hsa-miR-29c* [194] for lung cancer, *hsa-miR-16/ hsa-miR-15a* [195] for AML, *hsa-miR-146a/ hsa-miR-146b-5p* [196], *hsa-miR-200b/ hsa-miR-200c* [197], *hsa-miR-221/ hsa-miR-222* [198] for breast cancer, *hsa-miR-16/ hsa-miR-15a* [195] for prostate cancer, and *hsa-miR-200b/ hsa-miR-429* [199] for ovarian cancer.

These discoveries indicates that we can narrow the list of miRNA candidates using the sequence alignment similarity, *i.e.* if a miRNA is hypothetically involved in some diseases or interact with some genes, we can also test the miRNAs which are very similar to the miRNA we consider.

However, there are also many cases in which we cannot explore by using the sequence alignments an our visualization tool helps in these cases. For the other 9 pairs, the sequences are not similar to each other, but the are close in our visualization. For example the alignment score for *hsa-miR-7* and *hsa-miR-9* is -0.38824, which ranks about 45% according to the alignment similarity, however, they are functionally similar and are close to each other in our visualization. The local embedding results of these pair and another case (*hsa-miR-204*/*hsa-miR-205*) for breast cancer are highlighted in Figure 6.4. These cases suggests that the global embedding reflects the functional relationships among miRNAs more accurate than the local sequence similarity do.

6.2.5.2 Functional Prediction Using Causal miRNAs

In our studies, we also discover that we can often find some miRNAs which might be potentially involved in some diseases, and there is another miRNAs near by which is causally verified by biological experiments. According to our analysis above, these miRNAs might strongly hypothetically be involved in the same diseases, too. For example, in Figure 6.5 *B*, miRNA *hsa-miR-145* is verified to be involved in prostate cancer by causal experiments [200], while large scale micro-array experiments by other independent group support that *hsa-miR-10b* is correlated to the same disease [201]. Notice that the embedding distance profile between *hsa-miR-145* and *hsa-miR-10b* is $0.0223(0.10\%, 2, 3)$ while the sequence alignment similarity profile is $-4.7144(52.17\%, 214, 302)$, indicating that the two miRNAs are close to each other in embedding distance and dissimilar in sequence alignment. We highlight 3 other cases in Figure 6.5 and list more cases in Table 6.3.

One should notice that in Table 6.3, most of the miRNAs are dissimilar to each other according to the sequence alignment profiles. We believe in previous studies, researchers have already employ the sequences to guide how to select miRNA candidates and most of the possible similar miRNAs have been tested. Table 6.3 also suggests that we can discover many more candidates using our visualization tool while direct sequence comparison does not work.

Table 6.2. MiRNA pairs causally verified by biological experiments for AML, breast cancer, Prostate cancer, ovarian cancer. For the "Reference" column, the first number is the year in which the relationship of between the disease and the miRNA is verified, and the second number is the reference number. Details can be found in the main text.

Disease	miRNA	Reference	Embedding Distance	Sequence BLAST Score
L	*hsa-miR-200a*	2009 [192]	0.0298 (0.23%, 4, 4)	25.5131 (0.03%, 1, 1)
	hsa-miR-141	2009 [192]		
	hsa-miR-200b	2009 [192]	0.0520 (1.05%, 3, 2)	15.5297 (0.21%, 4, 3)
	hsa-miR-429	2009 [192]		
	hsa-miR-200b	2009 [192]	0.0666 (1.89%, 6, 13)	25.7904 (0.02%, 1, 1)
	hsa-miR-200c	2009 [192]		
	hsa-miR-20a	2005 [193]	0.0724 (2.29%, 4, 3)	26.8997 (0.01%, 2, 2)
		2007 [202]		
	hsa-miR-17	2005 [193]		
	hsa-miR-29b	2007 [194]	0.0293 (0.22%, 2, 2)	23.2945 (0.04%, 1, 2)
	hsa-miR-29c	2007 [194]		
A	*hsa-miR-16*	2007 [195]	0.0482 (0.86%, 3, 3)	15.5297 (0.21%, 2, 3)
	hsa-miR-15a	2007 [195]		
B	*hsa-miR-200a*	2009 [192]	0.0298 (0.23%, 4, 4)	25.5131 (0.03%, 1, 1)
	hsa-miR-141	2009 [192]		
		2008 [203]	0.0286 (0.20%, 2, 5)	-3.8824 (45.87%, 360, 336)
	hsa-miR-7	2008 [204]		
		2009 [205]		
	hsa-miR-9	2009 [206]		
	hsa-miR-127-3p	2006 [207]	0.0632 (1.68%, 9, 6)	-0.2773 (18.13%, 142, 80)
	hsa-miR-193b	2008 [208]		
	hsa-miR-128a	2008 [203]	0.0720 (2.26%, 11, 29)	-2.2185 (29.98%, 228, 193)
	hsa-miR-510	2008 [209]		
		2006 [210]	0.0765 (2.60%, 16, 42)	25.5131 (0.03%, 1, 1)
	hsa-miR-146a	2008 [196]		
		2008 [211]		
	hsa-miR-146b-5p	2008 [196]		
	hsa-miR-335	2008 [212]	0.0729 (2.33%, 29, 38)	-4.4371 (50.26%, 254, 301)
	hsa-miR-182	2009 [213]		
	hsa-miR-200b	2008 [197]	0.0666 (1.89%, 6, 13)	25.7904 (0.02%, 1, 1)
		2010 [214]		
		2008 [197]		
	hsa-miR-200c	2009 [215]		
		2009 [216]		
	hsa-miR-204	2008 [209]	0.0250 (0.14%, 3, 2)	-0.2773 (18.13%, 80, 98)
	hsa-miR-205	2008 [197]		
	hsa-miR-221	2008 [198]	0.0687 (2.05%, 40, 28)	13.8658 (0.25%, 1, 1)
	hsa-miR-222	2008 [198]		
	hsa-miR-510	2008 [209]	0.0426 (0.63%, 10, 10)	-6.6556 (70.61%, 500, 509)
	hsa-miR-199b-5p	2008 [217]		
P	*hsa-miR-16*	2007 [195]	0.0482 (0.86%, 3, 3)	15.5297 (0.21%, 2, 3)
	hsa-miR-15a	2007 [195]		
	hsa-miR-21	2009 [218]	0.0487 (0.89%, 19, 15)	0.2773 (14.05%, 152, 136)
		2009 [219]		
		2008 [220]		
	hsa-miR-101	2009 [221]		
		2010 [222]		
Ovarian Cancer	*hsa-miR-200b*	2009 [199]	0.0520 (1.05%, 3, 2)	15.5297 (0.21%, 4, 3)
	hsa-miR-429	2009 [199]		
	hsa-miR-200b	2009 [199]	0.0666 (1.89%, 6, 13)	25.7904 (0.02%, 1, 1)
	hsa-miR-200c	2009 [215]		

Table 6.3. 64 miRNAs which are near a causal verified miRNA and which have been tested using micro-array experiments. The “Causal” column is the miRNA which are causally verified. The corresponding reference is also given. The “Candidate” column is the miRNA which are tested using micro-array experiments and corresponding reference(s). Details can be found in the main text.

Disease	Causal	Candidate	Embedding Distance	Sequence BLAST Score
L	*miR-183* [223]	*miR-192* [180]	0.0416 (0.59%, 3, 2)	-0.5546 (19.19%, 178, 108)
		miR-377 [224]	0.0678 (1.99%, 22, 27)	-1.1093 (22.98%, 214, 153)
		miR-126 [180, 225, 226]	0.0460 (0.76%, 6, 23)	-2.7732 (36.15%, 323, 245)
		miR-224 [180]	0.0585 (1.40%, 16, 23)	4.4371 (2.96%, 33, 19)
	miR-197 [227]	*miR-150* [180]	0.0322 (0.28%, 2, 2)	3.8824 (3.73%, 13, 15)
	miR-19a [193]	*miR-101* [225]	0.0504 (0.98%, 18, 18)	3.8824 (3.73%, 21, 26)
	miR-429 [192]	*miR-338-5p* [225]	0.0566 (1.30%, 4, 6)	-1.1093 (22.98%, 145, 229)
A	*miR-126** [228]	*miR-130b* [229]	0.1632 (12.34%, 18, 90)	-3.8824 (45.87%, 268, 343)
	miR-16 [195]	*miR-195* [186]	0.0459 (0.76%, 2, 2)	21.9080 (0.06%, 1, 1)
	miR-204 [230]	*miR-22* [229]	0.0491 (0.91%, 11, 16)	-11.9246 (96.82%, 670, 685)
	miR-210 [231]	*miR-342-3p* [195]	0.0404 (0.54%, 4, 5)	-6.6556 (70.61%, 543, 414)
	miR-23a [231]	*miR-23b* [229]	0.0350 (0.37%, 8, 12)	26.6223 (0.02%, 1, 1)
	miR-320 [232]	*miR-331-5p* [186]	0.0332 (0.32%, 5, 2)	-5.2690 (58.99%, 338, 375)
	miR-34b [233]	*miR-182* [186]	0.0520 (1.06%, 15, 20)	-6.3783 (68.70%, 492, 418)
		miR-451 [229]	0.0516 (1.04%, 13, 14)	3.0505 (5.03%, 33, 53)
B	*miR-10b* [234]	*miR-145* [235]	0.0223 (0.10%, 3, 2)	-4.7144 (52.17%, 302, 214)
		miR-887	0.0041 (0.00%, 2, 2)	-10.8153 (93.96%, 650, 623)
	miR-126 [212, 236]	*miR-148a* [237, 238]	0.0317 (0.27%, 10, 10)	-1.1093 (22.98%, 147, 234)
	miR-141 [197]	*miR-181b* [158, 239]	0.0214 (0.10%, 2, 3)	0.0000 (15.66%, 119, 136)
		miR-181d [239]	0.0248 (0.14%, 3, 4)	0.0000 (15.66%, 120, 118)
	miR-146a [210, 196, 211]	*miR-143* [235]	0.0858 (3.37%, 18, 37)	3.6051 (4.24%, 34, 28)
	miR-146b-5p [196]	*miR-191* [235]	0.0348 (0.36%, 7, 8)	0.8319 (12.12%, 93, 50)
	miR-155 [240]	*miR-203* [235]	0.0334 (0.32%, 2, 2)	-6.6556 (70.61%, 503, 440)
	miR-205 [197, 241]	*miR-22* [242]	0.0608 (1.53%, 12, 22)	-6.1010 (66.21%, 423, 526)
	miR-221 [198]	*miR-148a* [237, 238]	0.0391 (0.49%, 9, 15)	-6.6556 (70.61%, 525, 580)
		miR-143 [235]	0.0563 (1.28%, 27, 11)	-5.2690 (58.99%, 453, 488)
	miR-222 [198]	*miR-202* [235]	0.0479 (0.85%, 16, 6)	2.2185 (7.53%, 66, 43)
		miR-136 [235]	0.0266 (0.16%, 3, 4)	1.3866 (10.07%, 85, 58)
		miR-152 [237, 238]	0.0617 (1.58%, 22, 31)	-2.4958 (32.43%, 315, 314)
	miR-27a [243, 213]	*miR-365* [239]	0.0815 (3.02%, 34, 40)	-7.2102 (74.50%, 546, 444)
	miR-661 [244, 213]	*miR-328* [245]	0.0608 (1.53%, 6, 4)	7.7648 (0.76%, 4, 3)
	miR-7 [203, 204, 205]	*miR-148a* [237, 238]	0.0768 (2.63%, 32, 50)	2.4958 (6.21%, 60, 69)
		miR-152 [237, 238]	0.0674 (1.95%, 22, 37)	-6.1010 (66.21%, 475, 565)
	miR-9 [206]	*miR-136* [235]	0.0119 (0.02%, 2, 2)	3.6051 (4.24%, 25, 20)
		miR-152 [237, 238]	0.0515 (1.03%, 17, 22)	-0.2773 (18.13%, 134, 186)
		miR-202 [235]	0.0507 (0.99%, 15, 11)	-7.2102 (74.50%, 522, 447)
		miR-148a [237, 238]	0.0681 (2.00%, 26, 39)	0.0000 (15.66%, 112, 190)
	miR-96 [213]	*miR-365* [239]	0.0791 (2.81%, 16, 36)	2.4958 (6.21%, 42, 19)

Table 6.4. Cont. of Table 6.3.

Disease	Causal	Candidate	Embedding Distance	Sequence BLAST Score
P	_miR-145_ [246]	_miR-10b_ [201]	0.0223 (0.10%, 2, 3)	-4.7144 (52.17%, 214, 302)
	miR-125b [246]	_miR-937_	0.0940 (4.11%, 17, 10)	1.3866 (10.07%, 58, 70)
		miR-125a-5p [247]	0.0284 (0.19%, 3, 3)	23.5719 (0.04%, 1, 1)
	miR-126* [248]	_miR-26a_ [247]	0.1060 (5.28%, 5, 13)	-0.5546 (19.19%, 102, 126)
	miR-145 [200]	_miR-10b_ [201]	0.0223 (0.10%, 2, 3)	-4.7144 (52.17%, 214, 302)
	miR-146a [249]	_miR-143_ [247]	0.0858 (3.37%, 18, 37)	3.6051 (4.24%, 34, 28)
		miR-373* [247]	0.0969 (4.38%, 26, 66)	-11.0926 (94.99%, 675, 558)
	miR-15a [250]	_miR-31_ [251]	0.0800 (2.89%, 5, 4)	-1.3866 (25.04%, 224, 202)
	miR-16 [250]	_miR-195_ [247, 201]	0.0459 (0.76%, 2, 2)	21.9080 (0.06%, 1, 1)
		miR-31 [247]	0.0686 (2.04%, 4, 3)	-3.0505 (37.82%, 285, 328)
		miR-182* [251]	0.1600 (11.92%, 15, 93)	-10.5380 (93.44%, 667, 654)
	miR-221 [252]	_miR-126_ [201]	0.0670 (1.92%, 39, 35)	-6.6556 (70.61%, 524, 517)
	miR-222 [252, 253]	_miR-202_ [247]	0.0479 (0.85%, 16, 6)	2.2185 (7.53%, 66, 43)
		miR-224 [196]	0.0846 (3.28%, 42, 52)	-2.4958 (32.43%, 318, 279)
	miR-21 [218, 219]	_miR-513-3p_ [247]	0.0199 (0.08%, 3, 5)	0.0000 (15.66%, 170, 101)
		miR-19b [247]	0.0539 (1.15%, 23, 13)	0.0000 (15.66%, 163, 101)
		miR-181a [201]	0.0670 (1.93%, 33, 23)	-0.8319 (20.27%, 218, 161)
	miR-23b [253]	_miR-23a_ [247]	0.0350 (0.37%, 12, 8)	26.6223 (0.02%, 1, 1)
		miR-491-3p [247]	0.0524 (1.08%, 19, 17)	-8.8741 (86.43%, 609, 574)
	miR-330-3p [254]	_miR-96_ [251]	0.0531 (1.11%, 15, 6)	-11.3700 (95.69%, 663, 669)
		miR-27a [247, 201]	0.1012 (4.79%, 47, 53)	-11.0926 (94.99%, 654, 672)
	miR-34a [255]	_miR-503_ [247]	0.0513 (1.02%, 3, 3)	-2.4958 (32.43%, 320, 267)
	miR-521 [256]	_miR-375_ [251]	0.0386 (0.48%, 7, 7)	-1.9412 (28.45%, 204, 175)
		miR-96 [251]	0.0572 (1.33%, 16, 7)	-2.4958 (32.43%, 222, 241)
O	_miR-34b_ [257]	_miR-487b_ [258]	0.0862 (3.41%, 50, 45)	-1.9412 (28.45%, 188, 210)
	miR-34c-3p [257]	_miR-221_ [258]	0.0454 (0.74%, 17, 15)	-7.4875 (76.66%, 444, 582)

6.2.6 Novel MiRNA Detection

By combining local information we obtain a global effective relationships for miRNAs and enhance the understanding. Here we are also interested in applying the similar technique to establish a novel miRNA predictor, *i.e.* to incorporate the local relationship together and to detect novel miRNAs given some known miRNAs. To be more specific, we sample candidates from the whole genome of some species (*H. sapiens* and *D. Melanogaster* in our studies), and pool them together with known miRNAs in the same species, compute the *local similarities* among all the known and candidate miRNAs, and use the global effective similarities to retrieve novel miRNAs from the candidates. The detail retrieval algorithms can be found in the “Materials and Methods” section.

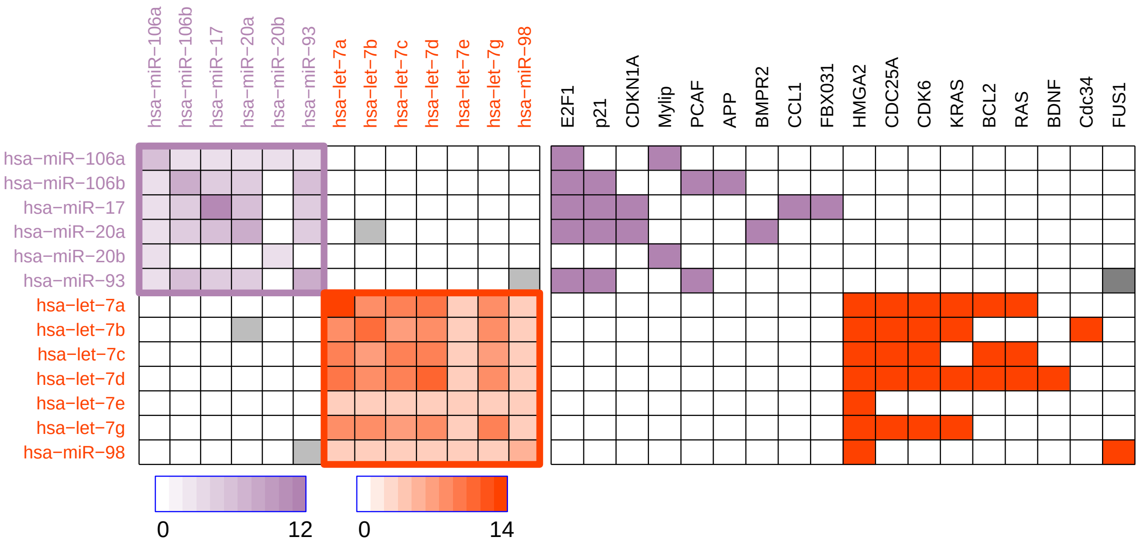


Figure 6.6. Cross analysis of *let-7/miR-98* family and *mir-106/mir-20* family. Left panel: the number of common targeting genes of pair of miRNAs. Right panel: the targeting genes of the members of the two family. White grid means no interaction is found and colored grid means interaction is found. All the targeting relationships are verified by biological experiments. .

We first test our algorithm, which is named, miRNAPred (miRNA gReen fuNction Affinity Prediction), on human miRNAs to see the prediction accuracies then apply it in *D. Melanogaster* to detect novel miRNAs.

6.2.6.1 *H. sapiens* miRNA Prediction Evaluation

We evaluate the performance of our miRNAPred algorithm on *H.sapiens* data based on the known miRNA precursors mixed with the pool of putative candidates to be ranked. The prediction quality is assessed by the recall and precision, which are, respectively, defined as:

$$\text{Recall} = \frac{TP}{TP+FN} \tag{6.2}$$

$$\text{Precision} = \frac{TP}{TP+FP}, \tag{6.3}$$

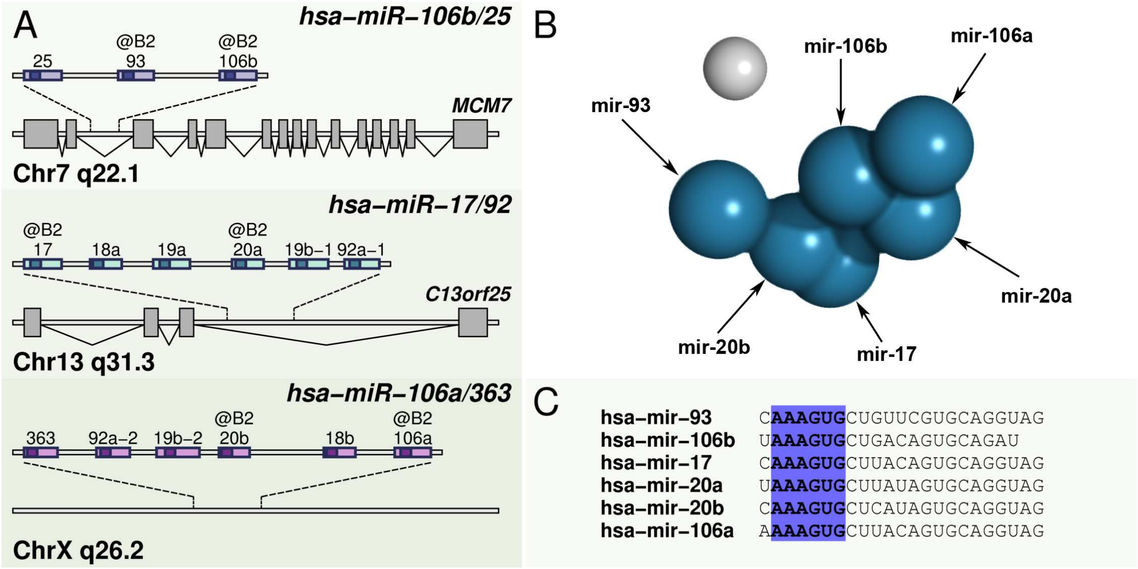

A
hsa–miR–106b/25
@B2
@B2
25
93
106b
MCM7
Chr7 q22.1
hsa–miR–17/92
@B2
@B2
17
18a
19a
20a
19b–1
92a–1
C13orf25
Chr13 q31.3
hsa–miR–106a/363
@B2
@B2
363
92a–2
19b–2
20b
18b
106a
ChrX q26.2
B
mir-106b
mir-106a
mir-93
mir-20a
mir-20b
mir-17
C
hsa–mir–93 CAAAGUGCUGUUCGUGCAGGUAG
hsa–mir–106b UAAAGUGCUGACAGUGCAGAU
hsa–mir–17 CAAAGUGCUUACAGUGCAGGUAG
hsa–mir–20a UAAAGUGCUUAUAGUGCAGGUAG
hsa–mir–20b CAAAGUGCUCAUAGUGCAGGUAG
hsa–mir–106a AAAAGUGCUUACAGUGCAGGUAG

Figure 6.7. Members of miRNAs group *B2* found by our visualization. *A*: The chromosome positions of the members of group *B2*, which are located in three paralog miRNAs clusters, the *miR-25* cluster, *miR-17* cluster, and *mir-106a* cluster. "@B2" means the corresponding miRNA is in group *B2*. *(B)*: The embedding result of the members of group *B2*. The symbols in the parentheses represent the function of the corresponding miRNA. *B*: breast cancer, *L*: Lung cancer, *O* ovarian cancer, *P*: prostate cancer. Upper case of the disease name means the function is verified by causal experiments, lower case means the function is verified by micro array experiments. *C*: The sequences of the members of miRNA group *B2*. The highlighted region is seed sequence which is used to silence the targeting mRNAs. .

where TP, FP and FN are numbers of true positive predictions, false positive predictions, and false negative predictions, respectively.

Our miRNAPred method is compared to the state-of-the-art miRank method [259] that has been proved to outperform previous Support Vector Machine (SVM) based supervised methods [130, 260]. The number of query samples is the most critical parameter for algorithm miRank. We perform the two methods, miRank and miRNAPred, on *H. sapiens* data with 4, 8, 16, and 32 known miRNA precursors that are randomly selected (we do it for 30 times) as query samples. To reiterate, in each of these experiments, the rest known miRNA precursors are combined with the 1000 hairpin sequences extracted

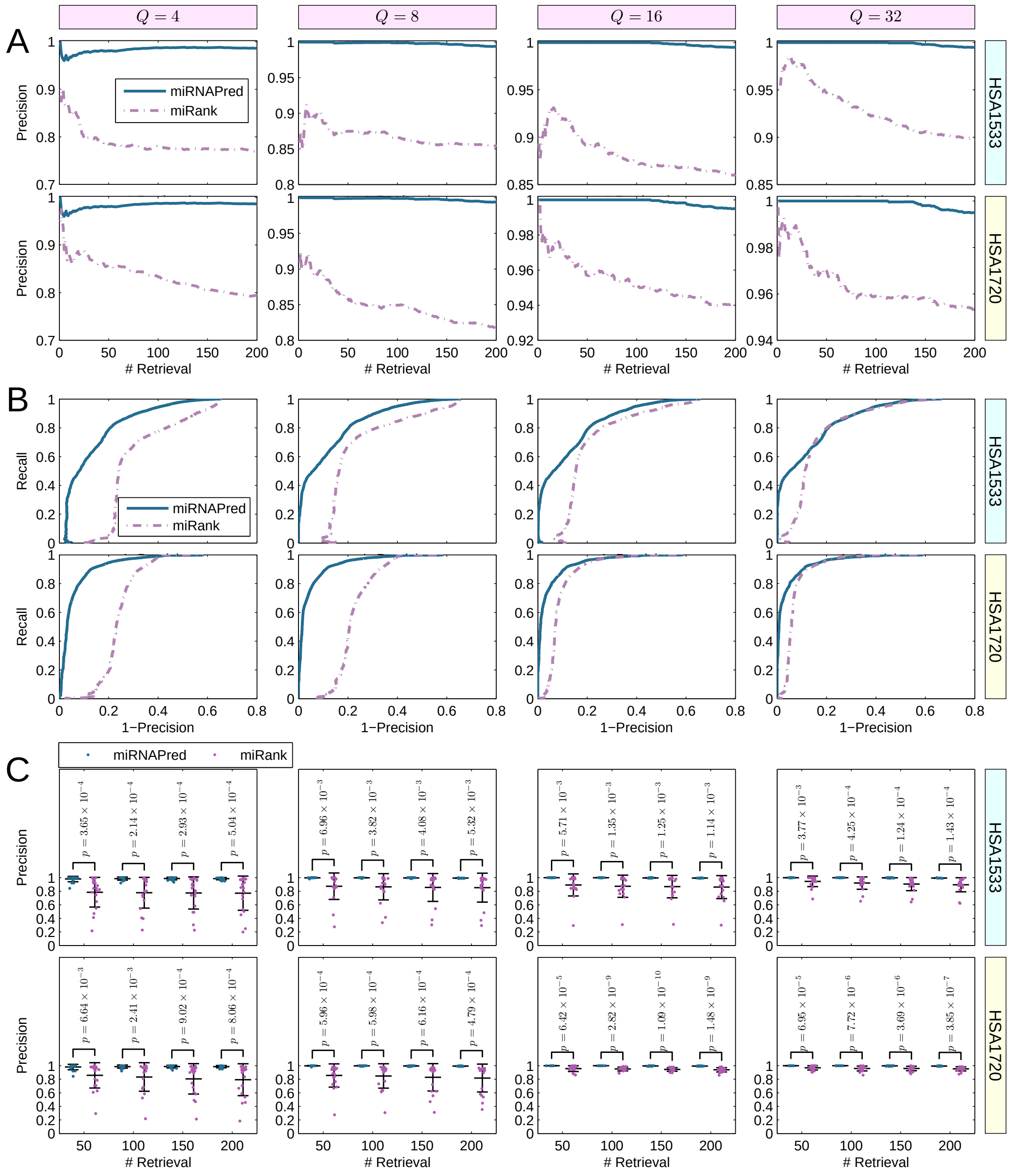


Figure 6.8. Novel miRNA prediction evaluation on *H. sapiens* for miRank and miRNAPred. *A*: Prediction precision with different numbers (4, 8, 16, and 32) of query (known) miRNAs on *HSA1533* and *HSA1720*. *B*: ROC curve with different numbers of query miRNAs on *HSA1533* and *HSA1720*. *C*: The precision distributions of both methods with different numbers of query miRNAs and different numbers (50, 100, 150, and 200) of retrieved miRNAs on *HSA1533* and *HSA1720*. Each dot is for one random choice of the selected query miRNAs. The *p* values indicate the significance of miRNAPred is better than miRank. When the number of known miRNAs is small, miRank is unstable in prediction precision, while miRNAPred remains robust..

from the genome to form the pool of candidates to be ranked. The recall and precision are computed by averaging the measurements over 30 random trials. In each experiment, we choose *n* topmost ranked candidates, and determine the precision and recall of the result by comparing the chosen candidates with the known human miRNAs that are hidden in the candidate pool. We test two algorithms in both HSA1533 (533 miRNAs in miRBase with the version of September 1, 2007, mixed by 1000 putative candidates) and HSA1720 (720 miRNAs in miRBase with the version of Release 16: Sept. 2010, mixed by 1000 putative candidates). By varying the number *n*, we obtain the Receiver Operating Characteristic (ROC) curves ([261]) for both methods and plot results in Figure 6.8 *B*.

With 8 known miRNAs, our first 50 predictions are 100% correct, much higher than miRank (87.45%) in HSA1533. As an extreme example, we use 4 known miRNAs to predict 200 top ranked miRNAs in HSA1720, 98.21% of them are correctly retrieved. In the same scenario, miRank only achieves 79.83%. Detailed comparisons on the two datasets in different settings can be found in Supplementary materials (Table S–1).

We are also interested in the precision with small number of retrieved miRNAs. This is useful in biological experimental design. With different small numbers of retrieval miRNAs, the prediction precisions of two methods are drawn in Figure 6.8 *A*. From the figure, we see that miRNAPred is significantly better than miRank in precisions.

In Figure 6.8 *C*, we plot the distribution of the prediction precision with different numbers of query miRNAs and different numbers of retrieved miRNAs. One can observe that our method miRNAPred is significantly better than miRank with *P* values ranging from 1.14×10^{-3}($Q = 16$, # retrieval=200, *HSA1553*) to 1.09×10^{-10}($Q = 16$, # retrieval=150, *HAS1720*). We also notice that when the number of known miRNAs is small, miRank is unstable in prediction precision, while miRNAPred remains robust.

6.2.6.2 Novel miRNAs detection in *D. melanogaster*

D. melanogaster is a species of Diptera, in the family Drosophilidae. The species is known as the common fruit fly which is one of the most frequently used model organisms in biology, including studies in genetics, physiology, microbial pathogenesis, and life history evolution. In the latest version of miRBase (Sept. 2010), there are 176 miRNAs found.

By applying miRNAPred on the mixed pool of 176 known miRNAs and 1000 putative candidates which are closest to known miRNAs, we retrieve 200 novel miRNAs. Out of the first 30 candidates, associate with the 30 highest ranking scores, 14 of them are conserved in other animal species. By conserved, we mean there are at least 21 nt are conserved in at least one other species.

We list the first 30 miRNAs detected by miRNAPred in Table 6.5. The putative miRNAs are sorted according to ranking scores. The positions (including the name of the chromosome, starting position and ending position) are also listed.

We show the hairpin structures of the first two putative miRNAs (*dme-putative1* and *dme-mir-519*) in Figure 6.9 *A*. The hairpin structures and the entropy are computed by the RNAFold web server [2]. Shown are also the alignment result of *dme-mir-519* and conserved miRNAs in other species.

6.2.7 Discussion

In causal studies of biological mechanisms, we often need to establish some potential hypothesis than design biological experiments to support them. With the fast growing biological techniques, we have exponentially accumulated publicly available experimental data. Putting expensive biological experiments on one hand and the free abundant data on the other hand, we often come up with one simple question: can we guide how to design

[2] http://rna.tbi.univie.ac.at/cgi-bin/RNAfold.cgi

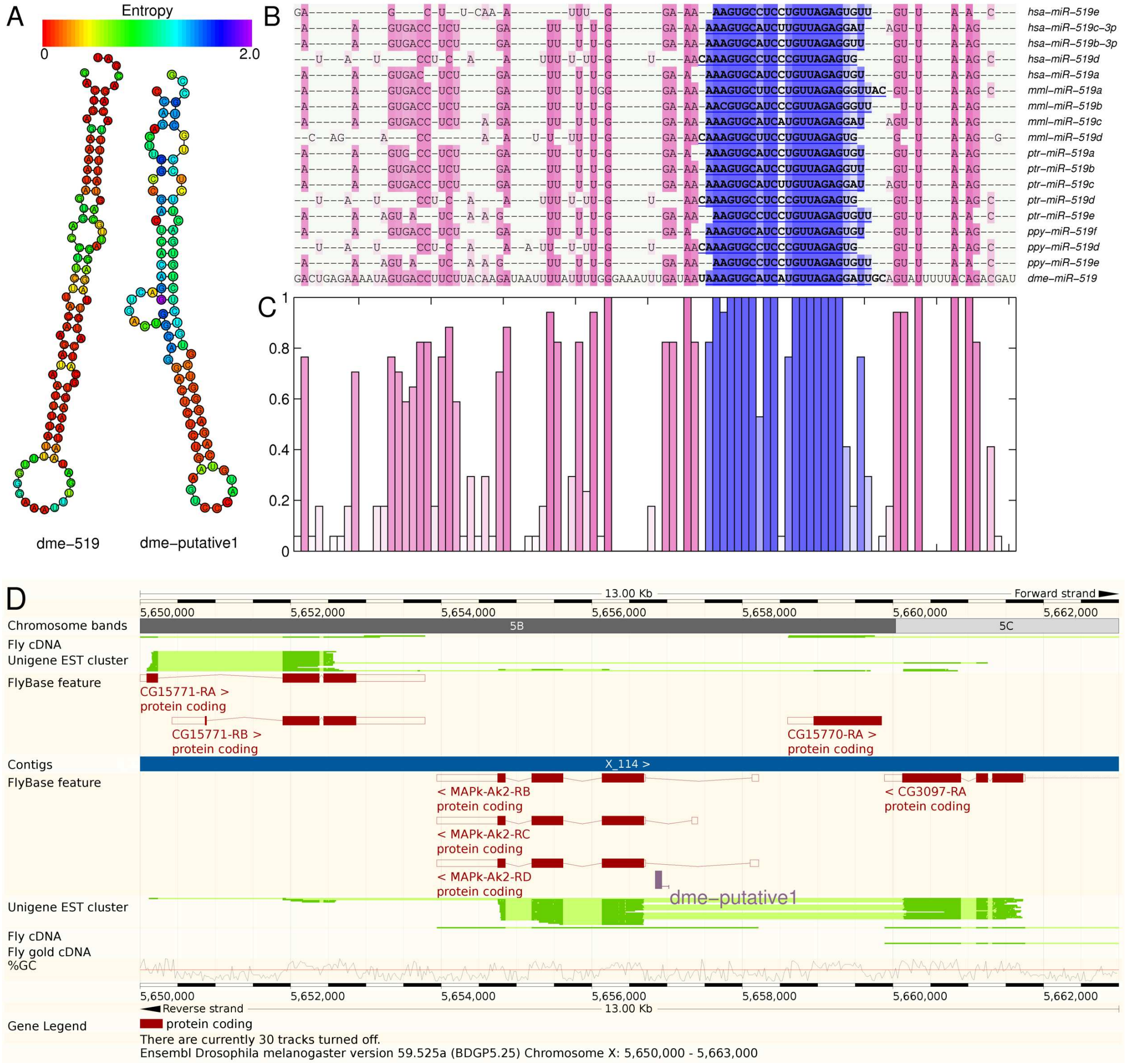


Figure 6.9. First two putative miNRAs (*dme-putative1* and *dme-mir-519*) predicted by miRNAPred. *A*: The hairpin structures of the two predicted miRNAs, computed by RNAFold. *B*: The alignment of *dme-mir-519* and other conserved miRNAs in other species. *C*: the conservation score with corresponding location of *B*. *D*: A view of the chromosome at *dme-putative1*, located at Chromosome X: 5656849–5656938. .

Table 6.5. Top 20 *D. melanogaster* candidate miRNAs discovered by miRNAPred.

ID[a]	Chr	Start	Stop	Strand	Ranking Score	Conserved [b]
dme-putative1	X	5656849	5656938	+	0.4957	No
dme-mir-519	U	1627351	1627450	–	0.4951	Yes
dme-putative2	2R	10410083	10410172	+	0.4949	No
dme-mir-548	3R	503941	504025	–	0.4945	Yes
dme-mir-792	X	14550110	14550189	–	0.4944	Yes
dme-miR1134	X	9961014	9961113	–	0.4944	Yes
dme-mir-669	X	19194916	19195015	+	0.4943	Yes
dme-putative3	2R	17987368	17987457	+	0.4940	No
dme-putative4	X	3964388	3964477	+	0.4938	No
dme-putative5	2R	11048444	11048533	+	0.4937	No
dme-mir-29	3L	11920039	11920138	–	0.4935	Yes
dme-putative6	2R	1409430	1409519	+	0.4930	No
dme-putative7	X	20008745	20008834	+	0.4923	No
dme-putative8	3L	5856504	5856593	+	0.4921	No
dme-putative9	3R	26690572	26690661	+	0.4912	No
dme-putative10	3R	27475642	27475731	+	0.4906	No
dme-putative11	2L	21231648	21231737	+	0.4896	No
dme-putative12	3L	6641688	6641777	+	0.4886	No
dme-putative13	2L	17962072	17962161	+	0.4879	No
dme-mir-1375	2R	19389491	19389570	–	0.4877	Yes

[a] For those putative miRNAs which are conserved in other species, we use the known conserved miRNAs to name the putative miRNAs. [b] 'Conserved' indicates we can find conservation miRNAs in other species. The criterion is that if there are at least 21 nt which are conserved in at least one other species, the candidate is considered as 'conserved'.

our experiments by making use of the available data such that our biological experiments have lower cost or higher chance to succeed? Many previous computational research have provided positive and successful answers. However, when come to the research miRNAs, the contribution from computation techniques to real biological communities is limited, due to the much higher complexity level among miRNAs, genes, pathways, and diseases.

Here we give a bigger picture by briefly reviewing related research and further discuss the potential usage of our techniques presented in this chapter.

6.2.7.1 The Complexity of the MiRNA/gene/disease Network

In history, we first discovery the functions of proteins, then discovered that genes, which express proteins, are much more important. After that, we found that the regulators, which govern the genes are also critical in biological studies. When more factors being considered, we have wider view and deeper understanding on the hidden biological mechanisms. But what is the next step? The answer also becomes more and more difficult.

One of the difficulties is that the complexity of the regulatory network has been raised to a much higher level when regulatory components like miRNAs are being taken into account: tiny regulators have huge impacts. Imagine a whole network of the triangle relationships among genes, diseases, and miRNAs. Since the large number of target genes of single miRNAs, the *density* of the connection becomes much high than the network without miRNAs. We plot the distribution of number of targets of miRNAs in Figure 6.11 to illustrate what the complexity level of miRNAs regulatory networks. If we put them in the triangle network, we can imagine that for each nodes of miRNAs, they have hundreds of connections coming out and there are hundreds of such nodes.

The complexity level of the interaction network of miRNAs have also been supported by the discoveries by Lewis *et al.* [168] and Krek *et al.* [169].

6.2.7.2 Narrowing the List of Candidates

The computational techniques are applied in biological studies, among which candidates selection has been widely accepted.

Cozma *et al.* applied a bioinformatics-based strategy to identify that *c-Myc* and *Cdc25A Apmt* mammary tumor latency modifiers [262]. In their argument they used sequences comparison and pathway analysis, which strongly support that *c-Myc* is the *Apmt1* tumor latency modifier locus. In plat research, Mitchell *et al.* developed a novel computational approach to discover candidate genes for the synthesis and feruloylation of arabinoxylan. They provided strong evidences which strongly support for genes within *the GT43, GT47, GT61,* and *PF02458* families being responsible for the synthesis of arabinoxylans and its side chains. In their research, sequence similarity was the original evidence. Wu developed an analyzing technique to identify candidate genes from DNA microarrays gene expression data. [263]. He used individual test to identify the significance of every considered gene under different conditions and to rank the genes to obtain a list of candidate genes. His method nowadays can be interpreted as *feature selection* in machine learning and bioinformatics communities [264].

The all these computational methods mentioned above, they all considered the objects independently, one case by one case. To use the notations in this chapter, they only made use of *local information* of the available data. Obviously, in the studies of miRNA, especially considering the regulatory network, such techniques do not suffice. From Table 6.2, we observe that there are many cases in which the nearby miRNAs have already been simultaneously discovered in the same paper, suggesting that most of the close (in terms of sequence) by miRNAs have already been explored. We have to find something else to narrow the list of candidates in our studies.

On the other hand we discovered 20 pairs of miRNAs which are both causally verified by biological experiments and are nearby in the visualization results. Out of the 20 pairs, 9 of them are dissimilar in sequences and are discovered by different independent groups. This suggest that these research groups were not able to identify the tight patterns of these miRNAs. So they should be benefited from our visualization.

One should also notice that in Table 6.3, the sequence alignment similarities are often too low to use local pairwise comparison cannot discover any of the item in the table. Another interesting case is *hsa-miR-204* and *hsa-miR-205* in Figure 6.4 *B*. The aignment similarity profile for this pair of miRNA is $-0.2773(18.13\%, 80, 98)$, showing that they are actually very dissimilar in sequences, however, visualization distance profile is $0.0250(0.14\%, 3, 2)$, which is very close. This can also be observed by sight in Figure 6.4 *B*.

These evidences suggest that the visualization has clear superiority when we consider a whole network which involved complicated miRNAs interactions.

6.2.7.3 Discovering Interesting Patterns

In our visualization results, Figure 6.7 demonstrates a good example of the miRNA pattern we discovered. The 6 miRNAs (*hsa-miR-93, hsa-miR-106b, hsa-miR-17, hsa-miR-20a, hsa-miR-20b* and *hsa-miR-106a*) are very close to each other and far away from most of the other miRNAs (see Figure 6.1 (*B2*) for a bigger picture) and have a clear common sequence signature (*AAAGUG*) in their 5'-end. Further more they are actually located at three paralog miRNAs clusters (Figure 6.7 *A*). More evidence suggests that these six miRNAs should be functionally tight to each other, see the interaction network with genes in Figure 6.6.

The *miR-17-92 clusters* are a prototypical example of a polycistronic miRNA gene and have been well explored by other independent research groups [265, 193]. However, notice that our discoveries solely relies on *in silico* studies on publicly available data.

6.2.7.4 Potential Applications of Our Visualization Tool

Our visualization cannot automatically discovered new patterns. Instead, it provides a novel way in which we can analysis the available data. Besides the discoveries above, we believe that there are still many un-explored interesting patterns and other useful knowledge in our visualization results, which requires more careful investigation.

On the other hand, notice that the input of our visualization are the local and direct similarities of the objects we considered and the output is the visualization embedding coordinates of the objects. Thus we can employ our technique in any other network. Nowadays, the gene-gene [266], miRNA-gene [267], miRNA-disease [267, 268], and gene-disease [269] networks have been well established. Can we put all of them together? Since the best advantage of the global embedding is the integration of all kind of local information, our visualization tool shall offer an clear and positive answer.

6.2.7.5 Novel miRNAs Identification

Experimental cloning efforts have successfully identified highly expressed miRNAs from various tissues. In cloning-based approaches, distinct ∼22 nt RNA transcripts are first isolated and then intensively cloned and sequenced. Novel miRNAs identification by such biological experiments highly biased towards abundantly and/or ubiquitously expressed miRNAs; only abundant miRNA genes can be easily detected ([270, 133]). The found miRNAs are collected in the miRBase website[3]([267]). Alternative computational approaches have been developed to complement experimental methods as a powerful aid for finding tissue-specific or lowly expressed miRNAs. A number of computational methods for miRNA prediction were introduced using supervised learning [130, 260] and semi-supervised learning [259]).

6.3 Methods and Materials

We first introduce the whole computational protocol then explain the details of each component in the protocol one by one later. The whole protocol, including the visualization and miRNAPred is illustrated in Figure 6.10. The main part is the Green's Function Affinity, in which objects are modeled by partial differential equations which are solved by Green's function method. The input of the "Green's Function Affinity" is a matrix measuring the pairwise similarities among objects, which are computed by different ways for two different purposes: visualization (*A*) and novel miRNA prediction (*B*).

6.3.1 Laplacian Operator and Green's Function Affinity

The continuous Laplacian operator

[3] http://www.mirbase.org/

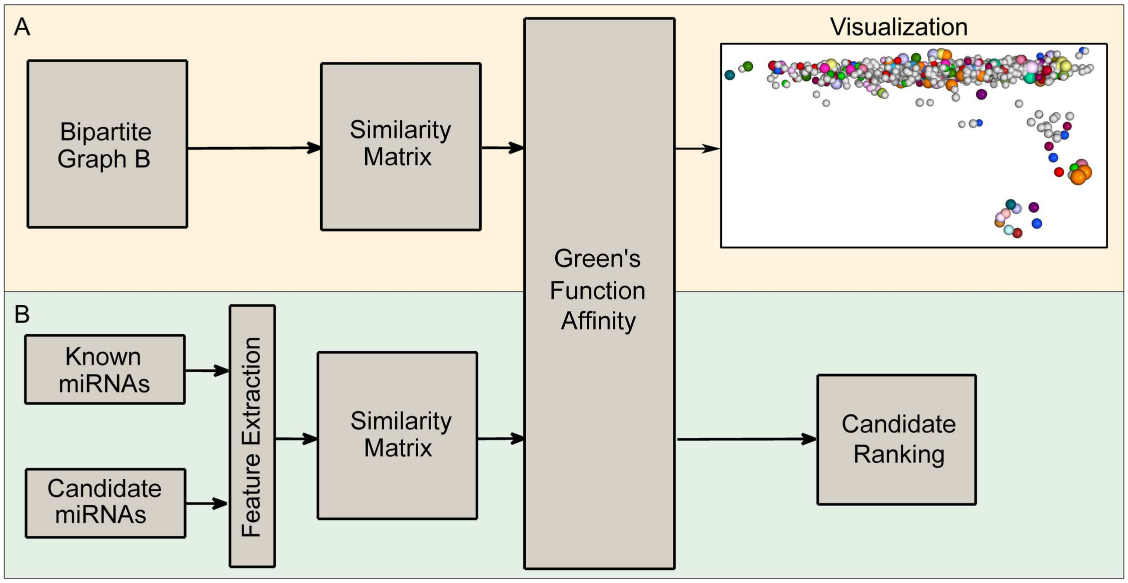


Figure 6.10. Computational protocol used in our miRNA analysis. *A*: Data flow chart for the visualization. The "Bipartite Graph B" block represent the bipartite graph **B** between human miRNAs and human genes. "Similarity matrix" represent the matrix **W** computed using Eq. (6.1). The "Green's Function Affinity" method will be introduced in the main text. *B*: The protocol used in *miRNAPred*. The features of known miRNAs and candidate miRNAs are extracted, then the similarity matrix is computed using Eq. (6.16). All the known miRNAs and candidates are considered to be precursors. Then the candidate miRNAs are ranked by "Green's Function Affinity". In the main text, we will explain how the candidates are obtained and how the feature extraction is done. .

$$\mathcal{L} f(\mathbf{x}) = \nabla^2 f(x_1, x_2, \cdots, x_d) = \left(\frac{\partial^2}{\partial x_1^2} + \cdots + \frac{\partial^2}{\partial x_d^2}\right) f(\mathbf{x}) \tag{6.4}$$

describes the second-order partial differential equation. Here f is a second-order differentiable function in a d-dimension space, $f : R^d \to R$. Given a partial differential equation,

$$\mathcal{L} f(\mathbf{x}) = y(\mathbf{x}). \tag{6.5}$$

The solution can be given by

$$f(\mathbf{x}) = \mathcal{L}^{-1} y(\mathbf{x}) \equiv \int G(\mathbf{x}, \mathbf{x}') s(\mathbf{x}') d\mathbf{x}', \tag{6.6}$$

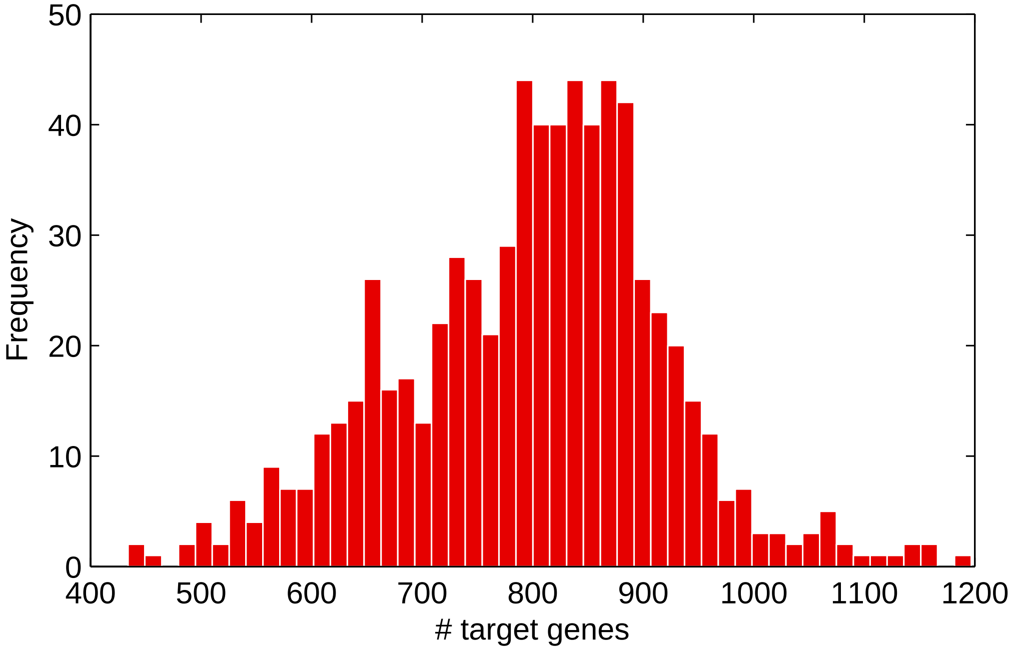


Figure 6.11. Distribution of number of targets of 711 miRNAs. Most of the miRNAs (93.81%) target more than 600 genes. .

where $G(\mathbf{r}, \mathbf{r}')$ is the Green's function, which captures the field response at $\mathbf{x}$ due to a single source at $\mathbf{x}'$ represented by $\delta(\mathbf{r} - \mathbf{r}')$:

$$\mathcal{L}\, G(\mathbf{x}, \mathbf{x}') = \delta(\mathbf{x} - \mathbf{x}'). \tag{6.7}$$

If the differential operator $\mathcal{L}$ admits a set of eigenvectors

$$\mathcal{L}\, \psi_i = \lambda_i \psi_i, \quad i = 1, 2, \cdots$$

then

$$G(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{\infty} \frac{\psi_i(\mathbf{x})\psi_i(\mathbf{x}')}{\lambda_i}.$$

The discrete Laplace operator is an analog of the continuous Laplace operator, and is defined so that it has meaning on a graph or a discrete grid. For the case of a finite-dimensional graph (with a finite number of edges and vertices), the discrete Laplace operator is more commonly called as the Laplacian matrix. Green's function for the Laplace operator represents the propagation of influence from all points. On a graph of pairwise similarities, the Green's function is the inverse of the combinatorial Laplacian. Will see later that the effect of Green's function is to incorporate all the connections together and enhance the local connection through the neighbors, like an affinity propagation. We name the method "Green's function affinity". In this chapter, we employ Green's function affinity to solve two problems in miRNAs: (1) visualization, *i.e.* to embed pairwise network data into Euclidian space, and (2) miRNAs predictions.

6.3.2 Network Embedding for MiRNAs Visualization

We seek a linear subspace to visualize the functional relationships of miRNAs according to the miRNAs:target information. Our aim is to see how one miRNA is close to or far away from another miRNA according to their functions in species. In our model, we define miRNAs as nodes of a graph. By embedding, miRNAs are represented by points in the vector space. The Euclidian distance between each pair of miRNAs in embedding space reflects their biological relationships.

Let $\mathcal{G} = (V, E)$ be a graph with vertices V and edges E, and $f : E \to R$ be a ring-valued function of the vertices. Then, the discrete Laplacian L acting on f is defined by

$$Lf(\mathbf{v}) = \sum_{\mathbf{u}:d(\mathbf{u},\mathbf{v})=1} [f(\mathbf{v}) - f(\mathbf{u})]. \tag{6.8}$$

where $d(\mathbf{u}, \mathbf{v})$ is the distance operator of $\mathbf{u}$ and $\mathbf{v}$ on the graph. $d(\mathbf{u}, \mathbf{v}) = 1$ can be interpreted as that node $\mathbf{u}$ and $\mathbf{v}$ are neighbors.

Let $V = \{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_n\}$ represent n miRNAs, and pairwise connectivity is given by $\mathbf{W}_{ij}$, $1 \leq i, j \leq n$. In our visualization method, $\mathbf{W}_{ij}$ is the number of common targets between two miRNAs i and j. The relationship between random walks on a graph and electric resistors network has been shown in [271]. One of the main results is the relationship between global resistance and the expected commute time is:

$$\tau_{ij} = CR_{ij}, \tag{6.9}$$

where τ_{ij} is the expected number of steps a random walker walks from i to j and comes back to i, R_{ij} is the global resistance between node i and j, and C is a constant. In order to illustrate the model in a more grounded way, we use the electric resistor network instead of random walk models. Let r^L_{ij} denote the *local electric resistor* between node i and j. In other word, the electric resistor network is established by connecting all pair nodes (i, j) via an electric resistor r^L_{ij}. Our goal is to compute the *global* or *effective* electric resistance between node i and j.

The global electric resistor can be derived as the following way. We impose different electric potentials between node i and j by adding an electric current source (assuming i is positive, j is negative, and the current is c_0) on them and all other nodes are free. Let u_k be the electric potential of node k, $k = 1, 2, ..., n$. According to Kirchhoff Law,

$$\mathbf{c}_i = \sum_{j \neq i} \kappa_{ij}(\mathbf{u}_i - \mathbf{u}_j), \tag{6.10}$$

where κ_{ij} is the *local electric conductivity* ($\kappa_{ij} = 1/r^L_{ij}$) and c_i is the *net currency* of node i. In order to visualize the functional relationship among miRNAs in a meaningful way, we use the number of common target genes as the *local conductivity, i.e.* $\kappa_{ij} = \mathbf{W}_{ij}$. Using the Laplacian operator, Equation (6.10) can be rewritten as,

$$L\mathbf{u} = \mathbf{c} \tag{6.11}$$

where $\mathbf{u} = [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_n]$ and L is the Laplacian matrix,

$$L = \mathbf{D} - \kappa,$$

$\mathbf{D} = \mathbf{diag}(\kappa\mathbf{e})$, $\kappa_{ij} = \kappa_{ij}$, and $\mathbf{e}$ is a column vector with all elements 1. Since i and j are applied with a current source c_0 and all other nodes are free, $c_i = c_0,\ c_j = -c_0$, and $c_k = 0,\ k \neq i, j$. Let $\sum_i \lambda_i q_i q_i^T = L$ be the eigenvector decompositions of L. Similarly to continues case, we have

$$\mathbf{u} = \mathbf{G}\mathbf{c} = L^{-1}\mathbf{c} = \left(\sum_{k=2}^{n} \frac{\mathbf{q}_k \mathbf{q}_k^T}{\lambda_k} \right) \mathbf{c}, \tag{6.12}$$

where $\mathbf{G}$ is the discrete Green's function. Notice that eigenvector $\mathbf{q}_1$ associating with the smallest eigenvalue λ_1 is ignored here. Because $\mathbf{q}_1$ is a constant vector and $\mathbf{q}_1^T\mathbf{c} = 0$. Thus

$$\mathbf{u}_i = \sum_{k=2}^{n} \frac{c_0\mathbf{q}_k^i - c_0\mathbf{q}_k^j}{\lambda_k} \mathbf{q}_k^i,$$

and

$$\mathbf{u}_j = \sum_{k=2}^{n} \frac{c_0\mathbf{q}_k^i - c_0\mathbf{q}_k^j}{\lambda_k} \mathbf{q}_k^j,$$

where $\mathbf{q}_k^i$ and $\mathbf{q}_k^j$ are the i-th and j-th component of $\mathbf{q}_k$, respectively. According to Ohm's Law,

$$R_{ij} = \frac{\mathbf{u}_i - \mathbf{u}_j}{c_0} = \sum_{k=2}^{n} \frac{(c_0\mathbf{q}_k^i - c_0\mathbf{q}_k^j)(\mathbf{q}_k^i - \mathbf{q}_k^j)}{c_0\lambda_k} = \sum_{k=2}^{n} \frac{(\mathbf{q}_k^i - \mathbf{q}_k^j)^2}{\lambda_k}.$$

Let

$$\mathbf{p}_i = [\mathbf{q}_2^i/\sqrt{\lambda_2}, \mathbf{q}_3^i/\sqrt{\lambda_3}, ..., \mathbf{q}_n^i/\sqrt{\lambda_n}]^T$$

and

$$\mathbf{p}_j = [\mathbf{q}_2^j/\sqrt{\lambda_2}, \mathbf{q}_3^j/\sqrt{\lambda_3}, ..., \mathbf{q}_n^j/\sqrt{\lambda_n}]^T,$$

we have

$$R_{ij} = \|\mathbf{p}_i - \mathbf{p}_j\|^2,$$

which means the effective resistance is the squared distance between two points in a vector space. More generally, all n miRNAs are embedded into a linear space:

$$[\mathbf{p}_2, \mathbf{p}_3, \cdots, \mathbf{p}_n] = \left[\frac{\mathbf{q}_2}{\sqrt{\lambda_2}}, \frac{\mathbf{q}_3}{\sqrt{\lambda_3}}, \cdots, \frac{\mathbf{q}_n}{\sqrt{\lambda_n}}\right]^T, \tag{6.13}$$

where the squared Euclidean distance represents the effective resistance. Notice that $\lambda_2 \leq \lambda_3 \leq \cdots \leq \lambda_n$, and $q_2, \cdots, q_n$ are orthogonal to each other, thus $[\mathbf{q}_2/\sqrt{\lambda_2}, \mathbf{q}_3/\sqrt{\lambda_3}, \mathbf{q}_4/\sqrt{\lambda_4}]^T$ are the first three *principle components* in the embedding space. In our study, we use these three components to visualize the functional relationship among miRNAs.

6.3.3 Green's function for miRNAs prediction

We use the electric resistor network model to illustrate our miRNAs prediction approach – miRNA gReen's functioN Affinity Prediction (miRNAPred). Let $\mathcal{G} = (\{V, \partial V\}, E)$ denotes a graph, in which each vertex $\mathbf{v} \in V$ represents a putative candidate and $\mathbf{v} \in \partial V$ represents a known miRNA precursor, an edge $\mathbf{e} \in E$ captures the relation between two vertices linked by $\mathbf{e}$, and the weight $\mathbf{w}$ of edge $\mathbf{e}$ quantifies the relation. More explicitly, we write the weights in the following order:

$$\mathbf{W} = \begin{pmatrix} \mathbf{W}_{\partial V \partial V}, \mathbf{W}_{\partial V V} \\ \mathbf{W}_{V \partial V}, \mathbf{W}_{VV} \end{pmatrix}$$

We use ∂V to represent the known miRNAs, because the known nodes can be interpreted as the boundary conditions of the partial differential equations. [272] showed that they are under Dirichlet boundary conditions.

Let the nodes corresponding to known miRNAs be imposed with a positive electric current source c^+ and putative candidates be imposed with negative c^-. We chose c^+ and c^- such that $mc^- = -nc^+$, where n and m are the number of known miRNAs and putative candidates, respectively. Then the electric current vector is

$$\mathbf{c} = (\overbrace{c^+, \cdots, c^+}^{n}, \overbrace{c^-, \cdots, c^-}^{m})^T. \tag{6.14}$$

Let $\mathbf{u}$ be electric potential results of the nodes. According to Kirchhoff Law, Eqs. (6.10) and (6.11) hold, which lead to the solution:

$$\mathbf{u} = \mathbf{G}\mathbf{c},$$

or explicitly

$$\mathbf{u}_i = \sum_{j \in \partial V} \mathbf{G}_{ij} c^+ + \sum_{j \in V} \mathbf{G}_{ij} c^-, \quad i \in V. \tag{6.15}$$

The electric potential $\mathbf{u}_i$ provides a natural way to rank the putative candidate miRNAs, *i.e.* we pick up the candidates which have highest potential.

miRNAPred Algorithm

We explicitly summarize miRNAPred algorithm as following:

1) Construct the local similarity matrix

$$\mathbf{W}_{ij} = e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma^2}, \tag{6.16}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are the feature vectors of (putative) miRNAs i and j, (described in §6.3.6), $i, j = 1, 2, \cdots, N$, where N is the total number of miRNAs, including n labeled miRNAs (denoted by ∂V) and m putative candidates (denoted by V), and compute the graph Laplacian Ł $= \mathbf{D} - \mathbf{W}$, where $D = \mathbf{diag}\,(W\mathbf{e})$.

2) Compute $\mathbf{G} = \sum_{k=2}^{N} \mathbf{q}_k \mathbf{q}_k^T / \lambda_k$, where λ_k and $\mathbf{q}_k$ are the eigenvalues and eigenvectors of graph Laplacian L, see Equation (6.12).
3) Compute ranking scores for all unlabeled (putative candidate) miRNAs. $\mathbf{u}_i = \sum_{j \in \partial V} \mathbf{G}_{ij} c^+ + \sum_{j \in V} \mathbf{G}_{ij} c^-, i \in V$. Here we set $c^+ = 1$ and $c^- = -n/m$.
4) Rank the putative candidates. Sort the ranking scores u_i of candidates, and select top ranked candidates as the final predicted miRNAs.

We should notice the **W** is calculated by different ways in visualization and miRNAPred, because the aims of visualization and prediction are different: miRNAPred uses the structural similarity to predict the miRNAs, while visualization utilizes the targeting genes to analyze miRNA functionality.

6.3.4 Data Sources

Here we introduce how the data are prepared for the Green's function affinity component in Figure 6.10.

6.3.4.1 MiRNAs-target bipartite graph

For human miRNAs visualization, we use the collection of miRNAs:target pairs which can be found in website [273]. The data used in this chapter are downloaded on Sept. 20th, 2010. There are 851 unique miRNAs, 21,199 unique genes, and 685,813 targeting pairs in the *Homo sapiens* species. MiRNAs which are not found in the same species in miRBase (http://www.mirbase.org/) are ignored. Considering miRNAs and target genes as two sets of vertices, we construct a bipartite graph B as following: $B_{ij} = 1$ if the i-th miRNA has the j-th target gene. The final size of B is 711×21199 indicating 711 miRNAs are selected. The total number of targeting pairs is 568,070.

6.3.4.2 Human MicroRNA Prediction

The precursor sequences of *H. sapiens* are downloaded from the miRBase[4] ([267]). Genome sequences of H.sapiens are retrieved from UCSC Genome Browser[5]. We verify two versions of miRNAs datasets, one is the version of September 1, 2007, and the other one is that of Release 16 (Sept, 2010). The first one has 533 miRNAs, and this number increased to 720 in the second version.

We randomly extract non-overlapping fragments of 90 nt from the genome so that no genome annotation information is used. We first discard all fragments overlapping with known miRNA precursors in miRBase (Release 15, Sept, 2010). For the extracted fragments, we further predict their secondary structures using RNAfold ([274]). We select fragments with the following criteria: minimum 18 base pairings on the stem of the hairpin structure, maximum -0.25 $kcal/mol$ average free energy of the secondary structure and no multiple loops. These fragments (putative candidates) are pooled together with two versions of known human miRNA precursors (533 miRNAs and 720 miRNAs, respectively) which are all known human miRNA precursors except the ones serve as query samples in our experiments to form the pool of candidates. The two versions of datasets are name HSA1533 and HSA1720, respectively. The reason we add those known human miRNA precursors to this pool of samples is to evaluate the prediction performance of the miRNAPred algorithm in terms of both precision and recall.

The reason why we choose 90 nt and a threshold of -0.25 $kcal/mol$ for average free folding energy is that most of the miRNA precursors are about 90 nt in length and have lower average free energy than -0.25 $kcal/mol$, see Figure 6.12 for the statistics for the precursor length and average free folding energy for all the miRNAs in all species available at miRBase.

[4] http://www.mirbase.org/

[5] http://genome.ucsc.edu/

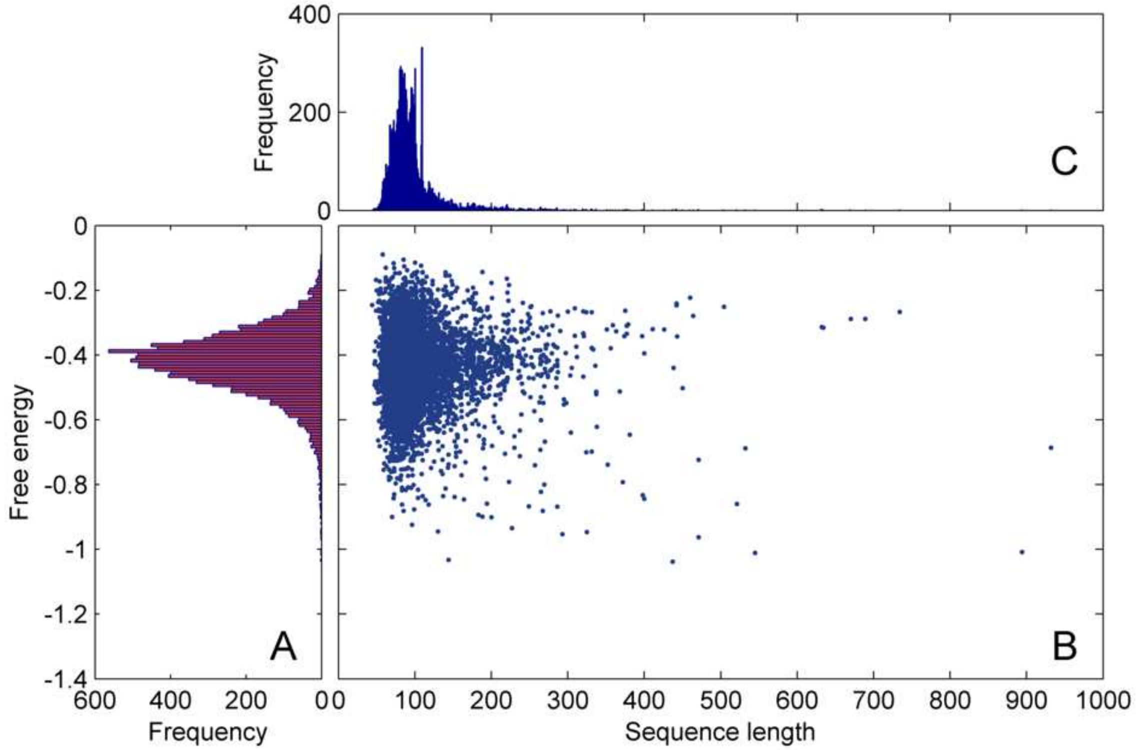


Figure 6.12. Statistics for precursor length and average free folding energy of the all miRNAs in all species available at miRBase. *A*: the distribution of average free folding energy. 96.92% of the miRNAs have lower average free folding free energy than -0.25. *B*: The correlation between average free folding energy and precursor length. *C*: Distributions of sequence length of miRNA precursors. Most of the miRNA precursors are about 90 nt in length. .

6.3.5 *Drosophila melanogaster*

The precursor sequences of *Drosophila melanogaster* are downloaded from the miRBase (http://www.mirbase.org/) with Release 16 (Sept, 2010). Every chromosome of *D. melanogaster* is fragmented, from 5'-end to 3'-end, by a sliding window of 90 nt and a shift increment of 45 nt. These fragments are folded by RNAfold ([274]), and hairpinned fragments are selected by the same criteria described above. The chosen hairpin sequences formed the initial candidate pool. In the fragmentation step, some putative candidates might be cut into two pieces, and have lost their hairpin structures, hence were excluded from the candidate pool. To avoid this, we further fragment, with the same sliding window and increment, the sequences between each pair of hairpinned fragments next to each other. The secondary structures of the new set of fragments are predicted and selected by the same tool and criteria. This process is iterated until no hairpinned fragment could be found.

6.3.6 Features used in miRNA prediction

We use the features introduced by [259] to describe miRNAs in a vector space. The entire hairpin structure of a miRNA precursor is characterized by 36 global and local intrinsic attributes that capture sequence, structural and topological properties of the miRNA precursor. Details can be found in [259].

CHAPTER 7

CONCLUSION

Interpretability is crucial to TCGA data analysis. The whole dissertation is dedicated to developVunderstandable Data Mining and Machine Learning techniques for TCGA data analysis.

We present the multi-subspace representation and discovery model, which is motivated by the multi-subspace discovery problem. We solve the multi-subspace discovery problem by providing block diagonal representation matrix where the data points are connected in the same subspace and disconnected for different subspace. We then extend our approach to handle noisy real world data which leads to the Multi-Subspace Representation. We develop an efficient algorithm for the presented model and a global optimizer is guaranteed. Empirical studies suggest that our method improves the quality of the data by sparse and low rank representation and the induced standalong classifier outperforms standard sparse representation approach.

In this dissertation, we also propose the explicit ℓ_2/ℓ_0-norm penalties and constraints to obtain structural sparsity models in multi-task learning and group sparsity learning. The induced learning problems are tackled by a novel general Lipschitz Auxiliary Function framework, which reduces the learning problem into simple iterative algorithms. We provide theoretical convergent guarantee, as well as the convergence rate guarantee. Empirical studies suggest that the explicit ℓ_2/ℓ_0-norm and group ℓ_2/ℓ_0-norm models achieve much lower objective values than ℓ_2/ℓ_1-norms under the same number selected joint variables.

It is also natural to extend our optimization techniques for even overlapped structural sparsity, such as grouped tree structure learning, or combined our methods with some other machine learning techniques such Structured Sparse Principal Component Analysis.

Our optimization techniques which are written in a general form, could also be useful in other machine learning models which involved both smooth and non-smooth objective functions. On the other hand, the concept of group lasso and the related group ℓ_2/ℓ_1 and ℓ_2/ℓ_0 norms can be extended to more general cases, *e.g.* for grouping in matrices and tensors, where our optimization techniques remain applicable.

We also present the Social Diffusion Process, which is motivated from the Matthew effect in social phenomenons. We develop the stochastic model by the assumption that social members tend to be together with someone who is familiar with. We also derive an graph evolution algorithm based on the presented mode. Empirical studies show significant improvement of the qualities of the graph data by the Social Diffusion Process, indicating that the assumptions in our model are natural in general. We also discover a new miRNA family in the experiment on miRNA functionality analysis.

In this dissertation, we propose *Scalable Orthogonal Regression* (SOR) to select low redundancy features. We propose an efficient iterative algorithm to resolve the problem and analyze its convergence rate. Furthermore, we also propose an extension of SOR to incorporate preselected features according to prior expertise knowledge. The effectiveness and efficiency of SOR is demonstrated on several benchmark data sets. Finally we also validate the usefulness of SOR on a real world clinical data set.

We also proposed novel computation tool to visualize and predict (miRNAPred) miRNAs using Green's function approach. The visualization tool embeds miRNAs into an Euclidean space, where the squared Euclidean distances naturally represent the functional relationship. We discover four tightly connected miRNAs patterns, two of which have been well studied in previous literature and two of which have not been explored yet.

By investigating the visualization results and combining with existing causal biological research and large scale micro-array experimental data, we discovered 20 pairs of miRNAs which are both causally verified by biological experiments to be involved in one or more in AML, prostate cancer, lung cancer, breast cancer, and ovarian cancer, and are nearby in the visualization results. Out of the 20 pairs, 9 of them are dissimilar in sequences and are discovered by different independent groups. We further discover 64 miRNAs which are near a causal verified miRNA and which have been tested using micro-array experiments. According to our analysis, these miRNAs are hypothesized involved in the corresponding five diseases we considered. The miRNAPred predictor, was tested to be robust even when few proportion of the miRNAs are used to retrieve unknown miRNAs. We successfully apply miRNAPred on *D. melanogaster*, and discover 30 novel miRNAs, out of which 14 are conserved in other animal species.

To summarize, we developed several integrative machine learning and data mining approaches from different point of views and it turns out that these approaches are consistent in discovering interesting and understandable patterns in TCGA data.

APPENDIX A

PROOF OF LEMMA 3.6.2

Proof Denote

$$J_{\mathbf{a},\mu}(\mathbf{u}) = \frac{1}{2}||\mathbf{u} - \mathbf{a}||^2 + \mu||\mathbf{u}||, \tag{A.1}$$

then,

$$J_{\mathbf{a},\mu}(\mathbf{u}) = \frac{1}{2}\mathbf{u}^T\mathbf{u} - \mathbf{a}^T\mathbf{u} + \frac{1}{2}\mathbf{a}^T\mathbf{a} + \mu\sqrt{\mathbf{u}^T\mathbf{u}}, \tag{A.2}$$

$$\frac{\partial J_{\mathbf{a},\mu}(\mathbf{u})}{\partial \mathbf{u}} = \mathbf{u} - \mathbf{a} + \frac{\mu}{\sqrt{\mathbf{u}^T\mathbf{u}}}\mathbf{u}, \tag{A.3}$$

By setting $\frac{\partial J_{\mathbf{a},\mu}(\mathbf{u})}{\partial \mathbf{x}} = 0$, and by denoting $b = \frac{1}{\sqrt{p\mathbf{u}^T\mathbf{u}}}$, we have

$$\mathbf{u} - \mathbf{a} + \mu b\mathbf{u}/p = 0,$$

or

$$\mathbf{u} = \frac{\mathbf{a}}{1 + \mu b}, \tag{A.4}$$

Equation (A.4) does not solve for **u**, since b is unknown. But we know that the optimal **u** can be always represented by the following form,

$$\mathbf{u} = \beta\mathbf{a}, \tag{A.5}$$

where β is scaler. By substituting (A.5) into (A.2), we have

$$J_{\mathbf{a},\mu}(\beta) = \frac{1}{2}||\mathbf{a}||^2\beta^2 + (\mu||\mathbf{a}|| - ||\mathbf{a}||^2)\beta + \frac{1}{2}||\mathbf{a}||^2. \tag{A.6}$$

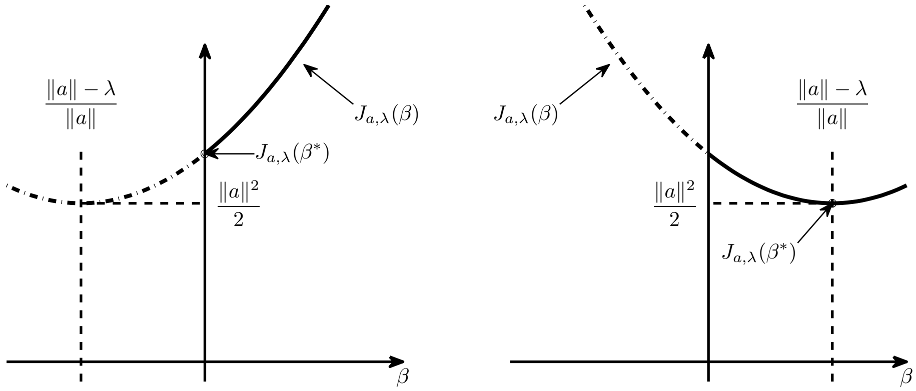


Figure A.1. The optimal solution of $J_{\mathbf{a},\mu}$. Left: the case of $p||\mathbf{a}|| - \mu \leq 0$. Right: the case of $p||\mathbf{a}|| - \mu \geq 0$. .

One can easily prove that when $\mathbf{u}$ is the optimal solution of (A.1), β is non-negative. To show this, we just assume $\mathbf{u} = \bar{\beta}\mathbf{a}, \bar{\beta} < 0$ is the optimal solution of (A.1). Then let $\mathbf{u}' = -\bar{\beta}\mathbf{a}$, and

$$\begin{aligned} J_{\mathbf{a},\mu}(\mathbf{u}') =& \frac{1}{2}(1+\bar{\beta})^2||\mathbf{a}||^2 + |\bar{\beta}|||\mathbf{a}|| \\ <& \frac{1}{2}(1-\bar{\beta})^2||\mathbf{a}||^2 + |\bar{\beta}|||\mathbf{a}|| \\ =& J_{\mathbf{a},\mu}(\mathbf{u}), \end{aligned}$$

which is impossible, since $\mathbf{u}$ is the optimal solution. Thus the optimization problem in (A.1) is to seek the $\beta \geq 0$, such that $\mathbf{u} = \beta\mathbf{a}$ gives the optimal solution, which leads to the following problem,

$$J_{\mathbf{a},\mu}(\beta) = \frac{1}{2}||\mathbf{a}||^2\beta^2 + (\mu||\mathbf{a}|| - ||\mathbf{a}||^2)\beta + \frac{1}{2}||\mathbf{a}||^2, \beta \geq 0. \tag{A.7}$$

One can check (see Figure A.1) that the optimal solution of (A.7) β^* is given the following,

$$\beta^* = \begin{cases} 0 & \text{if} \quad \mu \geq ||\mathbf{a}|| \\ \frac{||\mathbf{a}||-\mu}{||\mathbf{a}||} & \text{if} \quad \mu < ||\mathbf{a}|| \end{cases} \tag{A.8}$$

or

$$\mathbf{u}^* = \begin{cases} \mathbf{0} & \text{if} \quad \mu \geq ||\mathbf{a}|| \\ \frac{||\mathbf{a}||-\mu}{||\mathbf{a}||}\mathbf{a} & \text{if} \quad \mu < ||\mathbf{a}|| \end{cases} \tag{A.9}$$

which completes the proof.

APPENDIX B

PROOF OF THEOREM 3.7.1

For two consecutive solutions $\mathbf{X}^t, \mathbf{X}^{t+1}$, since $f(\mathbf{X})$ is convex and $\Phi(\mathbf{X})$ is convex on $\mathcal{D}$,

$$f(\mathbf{X}^*) \geq f(\mathbf{X}_t) + \langle \mathbf{X}^* - \mathbf{X}^t, \triangledown f(\mathbf{X}^t)\rangle, \tag{B.1}$$

$$\lambda\Phi(\mathbf{X}^*) \geq \lambda\Phi(\mathbf{X}^{t+1}) + \lambda\langle \mathbf{X}^* - \mathbf{X}^{t+1}, \mathbf{G}\rangle,$$

where $\mathbf{G} \in \partial\Phi(\mathbf{X}^{t+1})$ is any element in the sub-gradient of $\Phi(\mathbf{X})$ at $\mathbf{X}^{t+1}$. Since $\mathbf{X}^{t+1}$ is the optimal solution of

$$Z(\mathbf{X}, \mathbf{X}^t) = \frac{p}{2}\|\mathbf{X} - \mathbf{A}\|_F^2 + \lambda\Phi(\mathbf{X}) + C,$$

$\mathbf{0} \in \partial Z(\mathbf{X}^{t+1}, \mathbf{X}^t)$, or

$$\mathbf{0} \in p(\mathbf{X}^{t+1} - \mathbf{A}) + \partial\lambda\Phi(\mathbf{X}^{t+1}).$$

Obviously, $\mathbf{G} = p(\mathbf{A} - \mathbf{X}^{t+1})/\lambda$ must be in $\partial\Phi(\mathbf{X}^{t+1})$. Thus we have

$$\lambda\Phi(\mathbf{X}^*) \geq \lambda\Phi(\mathbf{X}^{t+1}) + \lambda\langle \mathbf{X}^* - \mathbf{X}^{t+1}, p(\mathbf{A} - \mathbf{X}^{t+1})/\lambda\rangle. \tag{B.2}$$

By combining (D.1) and (D.2) , we have

$$f(\mathbf{X}^*) + \lambda\Phi(\mathbf{X}^*) \geq \langle \mathbf{X}^* - \mathbf{X}^t, \triangledown f(\mathbf{X}^t)\rangle + \lambda\Phi(\mathbf{X}^{t+1}) + \lambda\langle \mathbf{X}^* - \mathbf{X}^{t+1}, p(\mathbf{A} - \mathbf{X}^{t+1})/\lambda\rangle,$$

By considering the fact that $Z(\mathbf{X}^{t+1}, \mathbf{X}^{t+1}) \leq Z(\mathbf{X}^{t+1}, \mathbf{X}^t)$, and that $Z(\mathbf{X}^{t+1}, \mathbf{X}^t) \leq Z(\mathbf{X}^t, \mathbf{X}^t)$, we have

$$f(\mathbf{X}^*) + \lambda\Phi(\mathbf{X}^*) \geq f(\mathbf{X}^{t+1}) + \lambda\Phi(\mathbf{X}^{t+1}) + \frac{p}{2}\|\mathbf{X}^{t+1} - \mathbf{X}^t\|_F^2 + p\langle \mathbf{X}^t - \mathbf{X}^*, \mathbf{X}^{t+1} - \mathbf{X}^t\rangle,$$

or

$$\begin{aligned}
J(\mathbf{X}^{t+1}) - J(\mathbf{X}^{*}) \leq & p\langle \mathbf{X}^{*} - \mathbf{X}^{t}, \mathbf{X}^{t+1} - \mathbf{X}^{t}\rangle - \frac{p}{2}\|\mathbf{X}^{t+1} - \mathbf{X}^{t}\|_F^2 \\
= & p\left(\langle \mathbf{X}^{*}, \mathbf{X}^{t+1}\rangle + \langle \mathbf{X}^{t}, \mathbf{X}^{t}\rangle - \langle \mathbf{X}^{*}, \mathbf{X}^{t}\rangle - \langle \mathbf{X}^{t}, \mathbf{X}^{t+1}\rangle\right) \\
& - \frac{p}{2}\left(\|\mathbf{X}^{t+1}\|_F^2 + \|\mathbf{X}^{t}\|_F^2 - 2\langle \mathbf{X}^{t}, \mathbf{X}^{t+1}\rangle\right) \\
= & \frac{p}{2}\left(\|\mathbf{X}^{t} - \mathbf{X}^{*}\|^2 - \|\mathbf{X}^{t+1} - \mathbf{X}^{*}\|^2\right).
\end{aligned}$$

According to Theorem 3.5.2, we have

$$J(\mathbf{X}^{T}) \leq J(\mathbf{X}^{T-1}) \leq J(\mathbf{X}^{T-2}) \leq \cdots \leq J(\mathbf{X}^{T_0}).$$

Thus

$$\begin{aligned}
\sum_{t=T_0}^{T-1} J(\mathbf{X}^{T}) - J(\mathbf{X}^{*}) \leq & \sum_{t=T_0}^{T-1} \frac{p}{2}\left(\|\mathbf{X}^{t} - \mathbf{X}^{*}\|_F^2 - \|\mathbf{X}^{t+1} - \mathbf{X}^{*}\|_F^2\right) \\
= & \frac{p}{2}\left(\|\mathbf{X}^{T_0} - \mathbf{X}^{*}\|_F^2 - \|\mathbf{X}^{T} - \mathbf{X}^{*}\|_F^2\right) \\
\leq & \frac{p}{2}\|\mathbf{X}^{T_0} - \mathbf{X}^{*}\|_F^2 \leq \frac{p_T}{2}\|\mathbf{X}^{T_0} - \mathbf{X}^{*}\|_F^2,
\end{aligned}$$

or

$$J(\mathbf{X}^{T}) - J(\mathbf{X}^{*}) \leq \frac{p_T}{2(T - T_0)}\|\mathbf{X}^{T_0} - \mathbf{X}^{*}\|^2. \tag{B.3}$$

Notice here we use the relation of

$$p_T \geq p_t, t = 1, 2, \cdots, T - 1.$$

APPENDIX C

PROOF OF THEOREM 4.4.1

We first introduce the following Lemma.

Lemma 2. *If* $\mathbf{A}$ *and* $\mathbf{B}$ *are semi-positive definite, then* $\mathbf{A} \odot \mathbf{B}$ *is also semi-positive definite.*

Proof. We first notice that if $\mathbf{u}$ and $\mathbf{v}$ are vectors, then

$$\left[(\mathbf{u}\mathbf{u}^T) \odot (\mathbf{v}\mathbf{v}^T)\right]_{ij} = u_i u_j v_i v_j = (u_i v_i)(u_j v_j),$$

Let $w_i = u_i v_i$, or $\mathbf{w} = \mathbf{u} \odot \mathbf{v}$ then

$$\left[(\mathbf{u}\mathbf{u}^T) \odot (\mathbf{v}\mathbf{v}^T)\right] = \mathbf{w}\mathbf{w}^T.$$

Since $\mathbf{A}$ is semi-positive definite, there exist $\mathbf{U}$ such that $\mathbf{A} = \mathbf{U}\mathbf{U}^T$. For the same reason, let $\mathbf{B} = \mathbf{V}\mathbf{V}^T$. Let $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \cdots \mathbf{u}_r], \mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_s]$ where r and s are the ranks of $\mathbf{A}$ and $\mathbf{B}$, respectively, and $\mathbf{w}_{ij} = \mathbf{u}_i \odot \mathbf{v}_j$, then

$$\mathbf{A} \odot \mathbf{B} = \sum\nolimits_{ij} \mathbf{u}_i \mathbf{u}_i^T \odot \mathbf{v}_j \mathbf{v}_j^T = \sum\nolimits_{ij} \mathbf{w}_{ij} \mathbf{w}_{ij}^T = \mathbf{W}\mathbf{W}^T,$$

where $\mathbf{W} = [\mathbf{w}_{11}, \cdots, \mathbf{w}_{1s}, \mathbf{w}_{21}, \cdots, \mathbf{w}_{rs}]$. Thus $\mathbf{A} \odot \mathbf{B}$ is semi-positive definite. □

Then we can prove the convexity of $f(\boldsymbol{\alpha})$ by showing the Hessian matrix of $f(\boldsymbol{\alpha})$ is positive semi-definite. From the gradient of Eq. (4.17), we can compute the Hessian of $f(\boldsymbol{\alpha})$ as

$$\mathbf{H}_{pq} = \frac{\partial\,(\nabla f(\boldsymbol{\alpha}))_p}{\partial \alpha_q} = \mathbf{G}_{pq} + \beta \frac{\partial \sum_j \alpha_p \alpha_j (\mathbf{x}_p^T \mathbf{x}_j)^2 \alpha_j}{\partial \alpha_q} \tag{C.1}$$

Let $\tilde{H}_{pq} = \frac{\partial \sum_j \alpha_p \alpha_j \mathbf{x}_i^T \mathbf{x}_j \alpha_j}{\partial \alpha_q}$, then

$$\tilde{H}_{pq} = \begin{cases} 2\alpha_p^2 (\mathbf{x}_p^T \mathbf{x}_p)^2 + \sum\nolimits_j \alpha_j^2 (\mathbf{x}_p^T \mathbf{x}_j)^2 & \text{if } p = q \\ 2\alpha_p \alpha_q (\mathbf{x}_p^T \mathbf{x}_p)^2 & \text{if } p \neq q \end{cases}$$

Thus the Hessian matrix is

$$\mathbf{H} = \mathbf{G} + 2\beta\mathbf{A} \odot \mathbf{G} \odot \mathbf{G} + 2\beta\mathbf{diag}(a_1, a_2, \cdots, a_n),$$

where $\mathbf{A} = \boldsymbol{\alpha}\boldsymbol{\alpha}^T$ and $a_i = \sum_j \alpha_j(\mathbf{x}_i^T\mathbf{x}_j)^2, i = 1, 2, \cdots, n$. Since $a_1 \geq 0$, $\mathbf{diag}(a_1, a_2, \cdots, a_n)$ is positive semi-definite. And according to Lemma 2, $\mathbf{A} \odot \mathbf{G} \odot \mathbf{G}$ is also positive semi-definite. Thus $\mathbf{H}$ is positive semi-definite. Thus $\mathbf{H}$ is positive semi-definite, and $f(\boldsymbol{\alpha})$ is convex. Obviously, $||\boldsymbol{\alpha}||_1$ is convex, hence $J(\boldsymbol{\alpha})$ is convex.

APPENDIX D

PROOF OF THEOREM 4.4.2

The following proof is similar to paper [5]. For two consecutive solutions $\boldsymbol{\alpha}^t, \boldsymbol{\alpha}^{t+1}$, since $f(\boldsymbol{\alpha})$ is convex and $||\boldsymbol{\alpha}||_1$ is convex,

$$f(\boldsymbol{\alpha}^*) \geq f(\boldsymbol{\alpha}^t) + (\boldsymbol{\alpha}^* \boldsymbol{\alpha}^t)^T \nabla f(\boldsymbol{\alpha}^t), \tag{D.1}$$

$$\lambda||\boldsymbol{\alpha}^*||_1 \geq \lambda||\boldsymbol{\alpha}^{t+1}||_1 + \lambda\boldsymbol{\alpha}^* - \mathbf{g}^T \boldsymbol{\alpha}^{t+1},$$

where $\mathbf{g} \in \partial||\boldsymbol{\alpha}||_1$ is any element in the sub-gradient of $||\boldsymbol{\alpha}||_1$ at $\boldsymbol{\alpha}^{t+1}$. Since $\boldsymbol{\alpha}^{t+1}$ is the optimal solution of

$$Z(\boldsymbol{\alpha}, \boldsymbol{\alpha}^t) = \frac{L}{2}||\boldsymbol{\alpha} - \mathbf{a}||^2 + \lambda||\boldsymbol{\alpha}||_1 + C,$$

$\mathbf{0} \in \partial Z(\boldsymbol{\alpha}^{t+1}, \boldsymbol{\alpha}^t)$, or

$$\mathbf{0} \in L(\boldsymbol{\alpha}^{t+1} - \mathbf{a}) + \partial\lambda\phi(\mathbf{a}^{t+1}).$$

Obviously, $G = L(\mathbf{a} - \boldsymbol{\alpha}^{t+1})/\lambda$ must be in $\partial\phi(\boldsymbol{\alpha}^{t+1})$. Thus we have

$$\lambda||\boldsymbol{\alpha}^*||_1 \geq \lambda||\boldsymbol{\alpha}^{t+1}|| + L(\boldsymbol{\alpha}^* - \boldsymbol{\alpha}^{t+1})^T(\mathbf{a} - \boldsymbol{\alpha}^{t+1}). \tag{D.2}$$

By combining (D.1) and (D.2) , we have

$$\begin{aligned} & f(\boldsymbol{\alpha}^*) + \lambda||\boldsymbol{\alpha}^*|| \\ \geq & (\boldsymbol{\alpha}^* - \boldsymbol{\alpha}^t)^T \nabla f(\boldsymbol{\alpha}^t) + \lambda||\boldsymbol{\alpha}^{t+1}||_1 + L(\boldsymbol{\alpha}^* - \boldsymbol{\alpha}^{t+1})^T(\mathbf{a} - \boldsymbol{\alpha}^{t+1}), \end{aligned}$$

By considering the fact that $Z(\boldsymbol{\alpha}^{t+1}, \boldsymbol{\alpha}^{t+1}) \leq Z(\boldsymbol{\alpha}^{t+1}, \boldsymbol{\alpha}^t)$, and that $Z(\boldsymbol{\alpha}^{t+1}, \boldsymbol{\alpha}^t) \leq Z(\boldsymbol{\alpha}^t, \boldsymbol{\alpha}^t)$, we have

$$\begin{aligned} f(\boldsymbol{\alpha}^*) + \lambda||\boldsymbol{\alpha}^*||_1 \geq & f(\boldsymbol{\alpha}^{t+1}) + \lambda||\boldsymbol{\alpha}^{t+1}||_1 \\ & + \frac{L}{2}||\boldsymbol{\alpha}^{t+1} - \boldsymbol{\alpha}^t||^2 + L(\boldsymbol{\alpha}^t - \boldsymbol{\alpha}^*)^T(\boldsymbol{\alpha}^{t+1} - \boldsymbol{\alpha}^t), \end{aligned} \tag{D.3}$$

then

$$
\begin{aligned}
&J(\boldsymbol{\alpha}^{t+1}) - J(\boldsymbol{\alpha}^*) \leq L(\boldsymbol{\alpha}^* - \boldsymbol{\alpha}^t)^T(\boldsymbol{\alpha}^{t+1} - \boldsymbol{\alpha}^t) - \frac{L}{2}||\boldsymbol{\alpha}^{t+1} - \boldsymbol{\alpha}^t||^2 \\
&= L\left(\boldsymbol{\alpha}^{*T}\boldsymbol{\alpha}^{t+1} + \boldsymbol{\alpha}^{tT}\boldsymbol{\alpha}^t - \boldsymbol{\alpha}^{*T}\boldsymbol{\alpha}^t - \boldsymbol{\alpha}^{tT}\boldsymbol{\alpha}^{t+1}\right) \\
&-\frac{L}{2}\left(||\boldsymbol{\alpha}^{t+1}||^2 + ||\boldsymbol{\alpha}^t||^2 - 2\boldsymbol{\alpha}^{tT}\boldsymbol{\alpha}^{t+1}\right) \\
&= \frac{L}{2}\left(||\boldsymbol{\alpha}^t - \boldsymbol{\alpha}^*||^2 - ||\boldsymbol{\alpha}^{t+1} - \boldsymbol{\alpha}^*||^2\right).
\end{aligned}
$$

According to Eq. (4.13), we have$J(\boldsymbol{\alpha}^T) \leq J(\boldsymbol{\alpha}^{T-1}) \leq \cdots \leq J(\boldsymbol{\alpha}^0)$. Thus

$$
\begin{aligned}
&\sum\nolimits_{t=T_0}^{T-1} J(\boldsymbol{\alpha}^T) - J(\boldsymbol{\alpha}^*) \leq \sum\nolimits_{t=0}^{T-1} \frac{L}{2}\left(||\boldsymbol{\alpha}^t - \boldsymbol{\alpha}^*||_F^2 - ||\boldsymbol{\alpha}^{t+1} - \boldsymbol{\alpha}^*||_F^2\right) \\
&= \frac{L}{2}\left(||\boldsymbol{\alpha}^0 - \boldsymbol{\alpha}^*||_F^2 - ||\boldsymbol{\alpha}^T - \boldsymbol{\alpha}^*||_F^2\right) \leq \frac{L}{2}||\boldsymbol{\alpha}^0 - \boldsymbol{\alpha}^*||_F^2 \leq \frac{L_T}{2}||\boldsymbol{\alpha}^0 - \boldsymbol{\alpha}^*||_F^2,
\end{aligned}
$$

or

$$
J(\boldsymbol{\alpha}^T) - J(\boldsymbol{\alpha}^*) \leq \frac{L_T}{2T}||\boldsymbol{\alpha}^0 - \boldsymbol{\alpha}^*||^2. \qquad \text{(D.4)}
$$

Notice here we use the relation of

$$
L_T \geq L_t, t = 1, 2, \cdots, T - 1.
$$

REFERENCES

[1] R. Siegel, D. Naishadham, and A. Jemal, “Cancer statistics, 2012,” *CA: A Cancer Journal for Clinicians*, 2012.

[2] R. Jenatton, G. Obozinski, and F. Bach, “Structured sparse principal component analysis,” in *Proc. AISTATS*. Citeseer, 2009.

[3] H. Zou and T. Hastie, “Regularization and variable selection via the elastic net,” *Journal of the Royal Statistical Society: Series B*, vol. 67, pp. 301 – 320, 2005.

[4] A. Beygelzimer, J. Kephart, and I. Rish, “Evaluation of optimization methods for network bottleneck diagnosis,” *ICAC*, 2007.

[5] D. Luo, C. Ding, and H. Huang, “Towards structural sparsity: An explicit ℓ_2/ℓ_0 approach,” in *2010 IEEE International Conference on Data Mining*. IEEE, 2010, pp. 344–353.

[6] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, “Robust face recognition via sparse representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[7] S. Roweis and L. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[8] D. Luo, F. Nie, C. Ding, and H. Huang, “Multi-subspace representation and discovery,” *Machine Learning and Knowledge Discovery in Databases*, pp. 405–420, 2011.

[9] G. Liu, Z. Lin, and Y. Yu, “Robust subspace segmentation by low-rank representation,” in *Proceedings of the 26th International Conference on Machine Learning, Haifa, Israel*. Citeseer, 2010.

[10] B. Olshausen and D. Field, “Sparse coding with an overcomplete basis set: A strategy employed by V1?” *Vision research*, vol. 37, no. 23, pp. 3311–3325, 1997.

[11] R. Tibshirani, “Regression shrinkage and selection via the LASSO,” *J. Royal. Statist. Soc B.*, vol. 58, pp. 267–288, 1996.

[12] W. Vinje and J. Gallant, “Sparse coding and decorrelation in primary visual cortex during natural vision,” *Science*, vol. 287, no. 5456, p. 1273, 2000.

[13] F. Bach and M. Jordan, “Predictive low-rank decomposition for kernel methods,” in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 33–40.

[14] E. Candes, X. Li, Y. Ma, and J. Wright, “Robust principal component analysis,” *preprint*, 2009.

[15] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. Altman, “Missing value estimation methods for DNA microarrays,” *Bioinformatics*, vol. 17, no. 6, p. 520, 2001.

[16] F. Wang, P. Li, A. König, and M. Wan, “Improving clustering by learning a bi-stochastic data similarity matrix,” *Knowledge and Information Systems*, pp. 1–32, 2011.

[17] W. Liu and S. Chang, “Robust multi-class transductive learning with graphs,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 381–388.

[18] A. Lewis, “The mathematics of eigenvalue optimization,” *Mathematical Programming*, vol. 97, no. 1, pp. 155–176, 2003.

[19] G. Watson, “Characterization of the subdifferential of some matrix norms,” *Linear Algebra and its Applications*, vol. 170, pp. 33–45, 1992.

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

[21] J. Shi and J. Malik, "Normalized cuts and image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 8, pp. 888–905, 2002.

[22] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml

[23] F. Nie, D. Xu, I. Tsang, and C. Zhang, "Spectral embedded clustering," in *Proceedings of the 21st international jont conference on Artifical intelligence*. Morgan Kaufmann Publishers Inc., 2009, pp. 1181–1186.

[24] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," *Proc. Neural Info. Processing Systems*, 2003.

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

[26] B. Efron, T. Hastie, L. Johnstone, and R. Tibshirani, "Least angle regression," *Annals of Statistics*, vol. 32, pp. 407–499, 2004.

[27] H. Dressman, A. Berchuck, G. Chan, J. Zhai, A. Bild, R. Sayer, J. Cragun, J. Clarke, R. Whitaker, L. Li, *et al.*, "An integrated genomic-based approach to individualized treatment of patients with advanced-stage ovarian cancer," *Journal of clinical oncology*, vol. 25, no. 5, pp. 517–525, 2007.

[28] E. Candès and T. Tao, "Decoding by linear programming," *IEEE Transactions on Information Theory*, vol. 51, pp. 4203–4215, 2005.

[29] ——, "Rejoinder: Statistical estimation when p is much larger than n," *Annuals of Statistics*, vol. 35, pp. 2392–2404, 2004.

[30] P. Zhao and B. Yu, "On model selection consistency of lasso," *Journal of Machine Learning Research*, vol. 7, pp. 2541– 2563, 2006.

[31] H. Kim and H. Park, “Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis,” *Bioinformatics*, vol. 23, no. 12, pp. 1495–1502, 2007.

[32] S. K. Shevade and S. S. Keerthi, “A simple and efficient algorithm for gene selection using sparse logistic regression,” *Bioinformatics*, vol. 19, no. 17, pp. 2246–2253, 2003.

[33] K. Huang, Y. Ying, and C. Campbell, “Generalized sparse metric learning with relative comparisons,” *Knowledge and Information Systems*, vol. 28, no. 1, pp. 25–45, 2011.

[34] S. Simmuteit, F. Schleif, T. Villmann, and B. Hammer, “Evolving trees for the retrieval of mass spectrometry-based bacteria fingerprints,” *Knowledge and Information Systems*, vol. 25, no. 2, pp. 327–343, 2010.

[35] C. Ding, D. Zhou, X. He, and H. Zha, “R1-PCA: Rotational invariant L1-norm principal component analysis for robust subspace factorization,” *Proc. Int’l Conf. Machine Learning (ICML)*, June 2006.

[36] E. J. Candès and J. K. Romberg, “Quantitative robust uncertainty principles and optimally sparse decompositions,” *Foundations of Computational Mathematics*, vol. 6, no. 2, pp. 227–254, 2006.

[37] E. Candès and M. WAKIN, “An introduction to compressive sensing,” *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[38] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, “Online learning for matrix factorization and sparse coding,” *Journal of Machine Learning Research*, vol. 11, pp. 10–60, 2010.

[39] A. Argyriou, T. Evgeniou, and M. Pontil, “Convex multi-task feature learning,” *Machine Learning*, vol. 73, no. 3, pp. 243–272, 2008.

[40] J. Peng, J. Zhu, A. Bergamaschi, W. Han, D.-Y. Noh, J. R. Pollack, and P. Wang, "Regularized multivariate regression for identifying master predictors with application to integrative genomics study of breast cancer," *The Annals of Applied Statistics*, vol. 2, no. 1, pp. 53–77, 2010.

[41] R. Tibshirani, "Spatial smoothing and hot spot detection for cgh data using the fused lasso," *Biostatistics*, vol. 9, no. 1, pp. 18–29, 2008.

[42] S. Huang, J. Li, L. Sun, J. Ye, A. Fleisher, T. Wu, K. Chen, and E. Reiman, "Learning brain connectivity of alzheimers disease by sparse inverse covariance estimation," *NeuroImage*, vol. 50, pp. 935–949, 2010.

[43] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," in *ICCV09*, 2009, pp. 2114–2121.

[44] L. Sun, R. Patel, J. Liu, K. Chen, T. Wu, J. Li, E. Reiman, and J. Ye, "Mining brain region connectivity for alzheimer's disease study via sparse inverse covariance estimation," in *SIGKDD09*, 2009, pp. 1335–1344.

[45] J. Liu, J. Chen, and J. Ye, "Large-scale sparse logistic regression," in *SIGKDD09*, 2009, pp. 547–556.

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

[47] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of The Royal Statistical Society Series B*, vol. 68, no. 1, pp. 49–67, 2006.

[48] G. Obozinski, B. Taskar, and M. I. Jordan, "Joint covariate selection and joint subspace selection for multiple classification problems," *Statistics and Computing*, vol. 20, pp. 231–252, 2010.

[49] X. Chen, Q. Lin, S. Kim, and E. Xing, “An efficient proximal-gradient method for single and multi-task regression with structured sparsity,” *stat*, vol. 1050, p. 26, 2010.

[50] L. Sun, J. Liu, J. Chen, and J. Ye, “Efficient recovery of jointly sparse vectors,” in *Advances in Neural Information Processing Systems 22*, 2009, pp. 1812–1820.

[51] P. Zhao, G. Rocha, and B. Yu, “Grouped and hierarchical model selection through composite absolute penalties,” *Annals of Statistics*, vol. 37, no. 6A, pp. 3468–3497, 2009.

[52] M. Stojnic, “ ℓ_2/ℓ_1-optimization in block-sparse compressed sensing and its strong thresholds,” *IEEE Journal of Selected Topics in Signal Processing*, 2009.

[53] J. Liu, S. Ji, and J. Ye., “Multi-task feature learning via efficient $l_{2,1}$-norm minimization,” in *UAI2009*, 2009.

[54] F. Nie, H. Huang, X. Cai, and C. Ding, “Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization,” in *NIPS*, 2010.

[55] J.-F. Cai, E. J. Candès, and Z. Shen, “A singular value thresholding algorithm for matrix completion.” *SIAM J. on Optimization*, vol. 20, no. 4, pp. 1956–1982.

[56] M. R. Osborne, B. Presnell, and B. A. Turlach, “On the lasso and its dual,” *Journal of Computational and Graphical Statistics*, vol. 9, no. 2, pp. 319–337, 2000.

[57] J. Tropp, “Just relax: Convex programming methods for subset selection and sparse approximation,” *ICES report*, pp. 04–04, 2004.

[58] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, “Multiple kernel learning, conic duality, and the smo algorithm,” *ICML*, 2004.

[59] F. R. Bach, “Bolasso: model consistent lasso estimation through the bootstrap,” in *ICML*, 2008, pp. 33–40.

[60] H. Zuo, “The adaptive lasso and its oracle properties,” *Journal of the American Statistical Association*, vol. 110, no. 476.

[61] J. Fan and R. Li, “Variable selection via nonconcave penalized likelihood and its oracle properties,” *J. Amer. Statist. Assoc.*, vol. 96, no. 456, pp. 1348–1360, 2003.

[62] 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*, vol. 67, no. 1, pp. 91 – 108, 2004.

[63] R. Tibshirani, “Spatial smoothing and hot spot detection for cgh data using the fused lasso,” *Biostatistics*, 2007.

[64] S. Mallat and Z. Zhang, “Matching pursuit in a time-frequency dictionary,” *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[65] W. J. Fu., “Penalized regressions: The bridge versus the lasso,” *Journal of Computational and Graphical Statistics*, vol. 7, no. 3, pp. 397–416, 2000.

[66] J. Friedman, T. Hastie, H. Hölfling, and R. Tibshirani, “Pathwise coordinate optimization,” *Annals of statistics*, vol. 1, no. 2, pp. 302–332, 2007.

[67] Y. Nesterov, “Gradient methods for minimizing composite objective function,” *Technical report*, vol. CORE, 2007.

[68] J. Tropp and A. Gilbert, “Signal recovery from random measurements via orthogonal matching pursuit,” *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[69] G. Davis, S. Mallat, and M. Avellaneda, “Greedy adaptive approximation,” *Journal of Constructive Approximation*, vol. 13, pp. 57–98, 1997.

[70] C. J. Leggetter and P. C. Woodland, “Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models,” *Computer Speech and Language*, vol. 9, no. 2, pp. 171–185, Apr. 1995.

[71] D. D. Lee and H. S. Seung, “Learning the parts of objects by non-negative matrix factorization,” *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[72] C. H. Q. Ding, T. Li, W. Peng, and H. Park, “Orthogonal nonnegative matrix t-factorizations for clustering,” in *KDD*, 2006, pp. 126–135.

[73] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*. Kluwer Academic Publishers, 2003.

[74] F. R. Bach, R. Thibaux, and M. I. Jordan, “Computing regularization paths for learning multiple kernels,” *NIPS*, 2004.

[75] D. D. Lee and H. S. Seung, “A method for solving a convex programming problem with convergence rate$o(1/k^2)$,” *Soviet Math. Dokl.*, vol. 27, pp. 372–376, 1983.

[76] A. F. Smeaton, P. Over, and W. Kraaij, “Evaluation campaigns and trecvid,” 2006.

[77] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. Vlahavas, “Multilabel classification of music into emotions,” in *Proc. 9th International Conference on Music Information Retrieval (ISMIR 2008), Philadelphia, PA, USA*, vol. 2008, 2008.

[78] E. Baralis, G. Bruno, and A. Fiori, “Measuring gene similarity by means of the classification distance,” *Knowledge and Information Systems*, vol. 29, no. 1, pp. 81–101, 2011.

[79] A. El Akadi, A. Amine, A. El Ouardighi, and D. Aboutajdine, “A two-stage gene selection scheme utilizing mrmr filter and ga wrapper,” *Knowledge and Information Systems*, vol. 26, no. 3, pp. 487–500, 2011.

[80] Z. Z. *et. al.*, “Imputation of missing genotypes: an empirical evaluation of impute,” *BMC Genetics*, vol. 9, no. 85, 2008.

[81] H. Liu and H. Motoda, *Feature Selection for Knowledge Discovery and Data Mining*. Springer, 1998.

[82] M. Petrik, G. Taylor, R. Parr, and S. Zilberstein, “Feature selection using regularization in approximate linear programs for markov decision processes,” *ICML*, 2010.

[83] P. Langley, “Selection of relevant features in machine learning,” in *AAAI Fall Symposium on Relevance*, 1994, pp. 140–144.

[84] R. Kohavi and G. H. John, “Wrappers for feature subset selection,” *Artificial Intelligence*, vol. 97, no. 1-2, pp. 273–324, 1997. [Online]. Available: citeseer.nj.nec.com/kohavi96wrappers.html

[85] Y. Yang and J. O. Pederson, “A comparative study on feature selection in text categorization,” in *ICML-97*, 1997.

[86] L. Yu and H. Liu, “Feature selection for high-dimensional data: A fast correlation-based filter solution,” in *ICML*, 2003, pp. 856–863.

[87] E. P. Xing, M. I. Jordan, and R. M. Karp, “Feature selection for high-dimensional genomic microarray data,” in *Proc. 18th International Conf. on Machine Learning*. Morgan Kaufmann, San Francisco, CA, 2001, pp. 601–608. [Online]. Available: citeseer.nj.nec.com/article/xing01feature.html

[88] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *The Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[89] N. Chuzhanova, A. Jones, and S. Margetts, “Feature selection for genetic sequence classification.” *Bioinformatics*, vol. 14, no. 2, p. 139, 1998.

[90] H. Peng, F. Long, and C. Ding, “Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, 2005.

[91] D. Luo, F. Wang, J. Sun, M. Markatou, J. Hu, and S. Ebadollahi, “Sor: Scalable orthogonal regression for low-redundancy feature selection and its healthcare applications,” in *Siam Data Mining*, 2012, p. to appear.

[92] S. Ma and J. Huang, “Regularized roc method for disease classification and biomarker selection with microarray data,” *Bioinformatics*, vol. 21, no. 24, p. 4356, 2005.

[93] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, “Least angle regression,” *Annals of statistics*, vol. 32, no. 2, pp. 407–451, 2004.

[94] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[95] W. Zhang, A. Surve, X. Fern, and T. Dietterich, "Learning non-redundant codebooks for classifying complex objects," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 1241–1248.

[96] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course (Applied Optimization)*, 1st ed. Springer Netherlands, 2003.

[97] N. Garofalo and D. Nhieu, "Lipschitz continuity, global smooth approximations and extension theorems for sobolev functions in carnot-carathéodory spaces," *Journal d'Analyse Mathématique*, vol. 74, no. 1, pp. 67–97, 1998.

[98] J. Liu, S. Ji, and J. Ye, "Multi-task feature learning via efficient l 2, 1-norm minimization," in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2009, pp. 339–348.

[99] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml

[100] S. Nene, S. Nayar, and H. Murase, "Columbia object image library (coil-100)," *Techn. Rep. No. CUCS-006-96, dept. Comp. Science, Columbia University*, 1996.

[101] S. Armstrong, J. Staunton, L. Silverman, R. Pieters, M. den Boer, M. Minden, S. Sallan, E. Lander, T. Golub, and S. Korsmeyer, "Mll translocations specify a distinct gene expression profile that distinguishes a unique leukemia," *Nature genetics*, vol. 30, no. 1, pp. 41–47, 2001.

[102] J. Khan, J. Wei, M. Ringner, L. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. Antonescu, C. Peterson, *et al.*, "Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks," *Nature medicine*, vol. 7, no. 6, pp. 673–679, 2001.

[103] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, “Gene selection for cancer classification using support vector machines,” *Machine learning*, vol. 46, no. 1, pp. 389–422, 2002.

[104] A. Rakotomamonjy, “Variable selection using svm based criteria,” *The Journal of Machine Learning Research*, vol. 3, pp. 1357–1370, 2003.

[105] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[106] A. Kalton, P. Langley, K. Wagstaff, and J. Yoo, “Generalized clustering, supervised learning, and data assignment,” in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2001, pp. 299–304.

[107] M. Belkin and P. Niyogi, “Laplacian eigenmaps for dimensionality reduction and data representation,” *Neural computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[108] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Conf. Computer Vision and Pattern Recognition*, June 1997.

[109] A. Y. Ng, M. I. Jordan, and Y. Weiss, “On spectral clustering: Analysis and an algorithm,” in *NIPS*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds. MIT Press, 2001, pp. 849–856.

[110] P. K. Chan, M. D. F. Schlag, and J. Y. Zien, “Spectral K -way ratio-cut partitioning and clustering,” in *DAC*, 1993, pp. 749–754.

[111] L. W. Hagen and A. B. Kahng, “New spectral methods for ratio cut partitioning and clustering,” *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 11, no. 9, pp. 1074–1085, 1992.

[112] K. Nowicki and T. A. B. Snijders, “Estimation and prediction for stochastic blockstructures,” *Journal of the American Statistical Association*, vol. 96, no. 455, pp. 1077–1088, 2001.

[113] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, “Mixed membership stochastic blockmodels,” pp. 1981–2014, Sept. 2008.

[114] J. Pitman, “Combinatorial stochastic processes,” Springer-Verlag, New York, Tech. Rep., Aug. 30 2002.

[115] D. Blei, T. Griffiths, and M. Jordan, “The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies,” *Journal of the ACM*, vol. 57, pp. 21 – 30, 2010.

[116] D. Blei and P. Frazier, “Distance dependent chinese restaurant processes,” in *ICML*, 2010.

[117] M. I. Jordan, Z. Ghahramani, T. Jaakkola, and L. K. Saul, “An introduction to variational methods for graphical models,” *Machine Learning*, vol. 37, no. 2, pp. 183–233, 1999.

[118] H. Ishwaran and L. F. James, “Generalized weighted chinese restaurant processes for species sampling mixture models,” *STATISTICA SINICA*, vol. 13, pp. 1211–1236, 2003.

[119] A. Ahmed and E. P. Xing, “Dynamic non-parametric mixture models and the recurrent chinese restaurant process: with applications to evolutionary clustering,” in *SDM*. SIAM, 2008, pp. 219–230.

[120] D. Luo, C. Ding, and H. Huang, “Graph evolution via social diffusion processes,” *Machine Learning and Knowledge Discovery in Databases*, pp. 390–404, 2011.

[121] S. van Dongen, “Graph Clustering by Flow Simulation,” *PhD thesis, University of Utrecht*, 2000.

[122] R. Merton, “The matthew effect in science,” *Science*, vol. 159, no. 3810, pp. 56–63, 1968.

[123] M. W. Rossiter, “The matthew matilda effect in science,” *Social Studies of Science*, vol. 23, no. 2, pp. 325–341, 1993.

[124] K. E. Stanovich, “Matthew effects in reading: Some consequences of individual differences in the acquisition of literacy,” *Reading Research Quarterly*, vol. 21, no. 4, pp. 360–407, 1986.

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

[126] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Scholkopf, “Learning with local and global consistency,” in *NIPS*, 2003, p. 321.

[127] D. P. Bartel, “MicroRNAs: target recognition and regulatory functions,” *Cell*, vol. 136, pp. 215–233, 2009.

[128] J. L. *et al*. Bearfoot, “Genetic analysis of cancer-implicated microrna in ovarian cancer.” *Clinical cancer research : an official journal of the American Association for Cancer Research*, vol. 14, no. 22, pp. 7246–7250, 2008.

[129] C. *et al*. Blenkiron, “MicroRNA expression profiling of human breast cancer identifies new markers of tumor subtype,” *Genome Biology*, vol. 8, no. 10, pp. R214+, 2007.

[130] K. Ng and S. Mishra, “De novo SVM classification of precursor microRNAs from genomic pseudo hairpins using global and intrinsic folding measures. bioinformatics,” vol. 23, pp. 1321–1330, 2007.

[131] J. C. Huang, B. J. Frey, and Q. Morris, “Comparing sequence and expression for predicting microRNA targets using genMIR3,” in *Pacific Symposium on Biocomputing*, R. B. Altman, A. K. Dunker, L. Hunter, T. Murray, and T. E. Klein, Eds. World Scientific, 2008, pp. 52–63.

[132] N. Dahiya and P. Morin, “MicroRNAs in ovarian carcinomas,” *Endocrine-Related Cancer*, vol. In press, pp. DOI: 10.1677/ERC–09–0203, 2009.

[133] E. *et al*. Berezikov, “Approaches to microRNA discovery.” *Nat. Genet.*, vol. 38 (Suppl), pp. S2–S7, 2006.

[134] Q. *et al*. Jiang, “miR2Disease: a manually curated database for microRNA deregulation in human disease,” *Nucleic Acids Res.*, vol. 37, pp. D98–104, 2009.

[135] G. *et al*. Hu, “*MicroRNA-98* and *let-7* confer cholangiocyte expression of cytokine-inducible src homology 2-containing protein in response to microbial challenge,” *J Immunol*, vol. 183, pp. 1617–1624, 2009.

[136] A. *et al*. Abbott, “The *let-7* microrna family members *mir-48, mir-84*, and *mir-241* function together to regulate developmental timing in caenorhabditis elegans,” *Dev. Cell*, vol. 9, pp. 403–414, 2005.

[137] R. C. Lee, R. L. Feinbaum, and V. Ambros, “The c. elegans heterochronic gene *lin-4* encodes small rnas with antisense complementarity to *lin-14*.” *Cell*, vol. 75, no. 5, pp. 843–854, 1993.

[138] B. *et al*. Wheeler, “The deep evolution of metazoan microRNAs,” *Evolution & Development*, vol. 11, pp. 50–68, 2009.

[139] B. Cullen, “Transcription and processing of human microRNA precursors,” *Molecular Cell*, vol. 16, no. 6, pp. 5861–865, 2004.

[140] Y. Zeng and B. R. Cullen, “Structural requirements for pre-microRNA binding and nuclear export by exportin 5.” *Nucleic Acids Res*, vol. 32, no. 16, pp. 4776–4785, 2004.

[141] A. Pratt and I. MacRae, “The RNA-induced silencing complex: a versatile genesilencing machine,” *J Biol Chem*, vol. 284, pp. 17 897–17 901, 2009.

[142] A. Esquela-Kerscher and F. Slack, “Oncomirs-microRNAs with a role in cancer,” *Nature Reviews Cancer*, vol. 6, no. 4, pp. 259–269, 2006.

[143] A. Cimmino, G. Calin, M. Fabbri, M. Iorio, M. Ferracin, M. Shimizu, S. Wojcik, R. Aqeilan, S. Zupo, M. Dono, *et al.*, “miR-15 and miR-16 induce apoptosis by targeting BCL2,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 39, p. 13944, 2005.

[144] G. A. Calin and C. M. Croce, “MicroRNA signatures in human cancers.” *Nature reviews. Cancer*, vol. 6, no. 11, pp. 857–866, 2006.

[145] S. *et al*. Ambs, “Genomic profiling of microrna and messenger rna reveals deregulated microrna expression in prostate cancer,” *Cancer Research*, vol. 68, no. 15, pp. 6162–6170, August 2008.

[146] *et al*. Hoffman, Aaron E., “microrna *mir-196a-2* and breast cancer: A genetic and epigenetic association study and functional analysis,” *Cancer Res*, vol. 69, no. 14, pp. 5970–5977, 2009.

[147] S. *et al*. Eacker, “Understanding microRNAs in neurodegeneration,” *Nature Reviews Neuroscience*, vol. advance online publication, p. doi:10.1038/nrn2726, 2009.

[148] P. *et al*. Soon, “*miR-195* and *mir-483-5p* identified as predictors of poor prognosis in adrenocortical cancer,” *Clin. Cancer Res.*, vol. 15(24), pp. 7684 – 7692, December 15, 2009.

[149] X. *et al*. Yang, “*miR-449a* and *mir-449b* are direct transcriptional targets of e2f1 and negatively regulate pRb-E2F1 activity through a feedback loop by targeting CDK6 and CDC25A,” *Genes & Dev.*, vol. 23(20), pp. 2388 – 2393, 2009.

[150] A. Schetter, N. Heegaard, and C. Harris, “Inflammation and cancer: interweaving microRNA, free radical, cytokine and p53 pathways,” *Carcinogenesis*, vol. 31, no. 1, p. 37, 2010.

[151] P. Voorhoeve, “MicroRNAs: Oncogenes, tumor suppressors or master regulators of cancer heterogeneity?” *Biochimica et Biophysica Acta (BBA)-Reviews on Cancer*, vol. 1805, no. 1, pp. 72–86, 2010.

[152] V. Davalos and M. Esteller, “MicroRNAs and cancer epigenetics: a macrorevolution,” *Current opinion in oncology*, vol. 22, no. 1, p. 35, 2010.

[153] S. Hu, M. Huang, Z. Li, F. Jia, Z. Ghosh, M. Lijkwan, P. Fasanaro, N. Sun, X. Wang, F. Martelli, *et al.*, “MicroRNA-210 as a Novel Therapy for Treatment of Ischemic Heart Disease,” *Circulation*, vol. 122, no. 11_suppl_1, p. S124, 2010.

[154] S. Ikeda, T. Pu, *et al.*, “Expression and function of MicroRNAs in heart disease,” *Current Drug Targets*, vol. 11, no. 8, pp. 913–925, 2010.

[155] I. Alvarez-Garcia and E. Miska, “MicroRNA functions in animal development and human disease,” *Development*, vol. 132, no. 21, p. 4653, 2005.

[156] R. Garzon, G. Calin, and C. Croce, “MicroRNAs in cancer,” *Annual review of medicine*, vol. 60, pp. 167–179, 2009.

[157] L. He, J. Thomson, M. Hemann, E. Hernando-Monge, D. Mu, S. Goodson, S. Powers, C. Cordon-Cardo, S. Lowe, G. Hannon, *et al.*, “A microRNA polycistron as a potential human oncogene,” *Nature*, vol. 435, no. 7043, pp. 828–833, 2005.

[158] S. Volinia, G. Calin, C. Liu, S. Ambs, A. Cimmino, F. Petrocca, R. Visone, M. Iorio, C. Roldo, M. Ferracin, *et al.*, “A microRNA expression signature of human solid tumors defines cancer gene targets,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 103, no. 7, p. 2257, 2006.

[159] J. Kota, R. Chivukula, K. O’Donnell, E. Wentzel, C. Montgomery, H. Hwang, T. Chang, P. Vivekanandan, M. Torbenson, K. Clark, *et al.*, “Therapeutic microRNA delivery suppresses tumorigenesis in a murine liver cancer model,” *Cell*, vol. 137, no. 6, pp. 1005–1017, 2009.

[160] C. Croce, “Causes and consequences of microRNA dysregulation in cancer,” *Nature Reviews Genetics*, vol. 10, no. 10, pp. 704–714, 2009.

[161] M. Hafner, M. Landthaler, L. Burger, M. Khorshid, J. Hausser, P. Berninger, A. Rothballer, M. Ascano Jr, A. Jungkamp, M. Munschauer, *et al.*, “Transcriptome-wide identification of RNA-binding protein and microRNA target sites by PAR-CLIP,” *Cell*, vol. 141, no. 1, pp. 129–141, 2010.

[162] M. Poy, L. Eliasson, J. Krutzfeldt, S. Kuwajima, X. Ma, P. MacDonald, S. Pfeffer, T. Tuschl, N. Rajewsky, P. Rorsman, *et al.*, "A pancreatic islet-specific microRNA regulates insulin secretion," *Nature*, vol. 432, no. 7014, pp. 226–230, 2004.

[163] W. Dunn, P. Trang, Q. Zhong, E. Yang, C. van Belle, and F. Liu, "Human cytomegalovirus expresses novel microRNAs during productive viral infection," *Cellular microbiology*, vol. 7, no. 11, pp. 1684–1695, 2005.

[164] J. Weidhaas, I. Babar, S. Nallur, P. Trang, S. Roush, M. Boehm, E. Gillespie, and F. Slack, "MicroRNAs as potential agents to alter resistance to cytotoxic anticancer therapy," *Cancer research*, vol. 67, no. 23, p. 11111, 2007.

[165] J. Rossi, "New hope for a microRNA therapy for liver cancer," *Cell*, vol. 137, no. 6, pp. 990–992, 2009.

[166] M. Negrini, M. Ferracin, S. Sabbioni, and C. Croce, "MicroRNAs in human cancer: from research to therapy." *Journal of cell science*, vol. 120, no. Pt 11, p. 1833, 2007.

[167] A. Tong and J. Nemunaitis, "Modulation of miRNA activity in human cancer: a new paradigm for cancer gene therapy?" *Cancer Gene Therapy*, vol. 15, no. 6, pp. 341–355, 2008.

[168] B. Lewis, C. Burge, and D. Bartel, "Conserved seed pairing, often flanked by adenosines, indicates that thousands of human genes are microRNA targets," *Cell*, vol. 120, no. 1, pp. 15–20, 2005.

[169] A. Krek, D. Grün, M. Poy, R. Wolf, L. Rosenberg, E. Epstein, P. MacMenamin, I. da Piedade, K. Gunsalus, M. Stoffel, *et al.*, "Combinatorial microRNA target predictions," *Nature genetics*, vol. 37, no. 5, pp. 495–500, 2005.

[170] A. Enright, B. John, U. Gaul, T. Tuschl, C. Sander, and D. Marks, "MicroRNA targets in Drosophila," *Genome biology*, vol. 5, no. 1, pp. 1–1, 2004.

[171] R. Friedman, K. Farh, C. Burge, and D. Bartel, "Most mammalian mRNAs are conserved targets of microRNAs," *Genome research*, vol. 19, no. 1, p. 92, 2009.

[172] A. Gavin, M. Bösche, R. Krause, P. Grandi, M. Marzioch, A. Bauer, J. Schultz, J. Rick, A. Michon, C. Cruciat, *et al.*, “Functional organization of the yeast proteome by systematic analysis of protein complexes,” *Nature*, vol. 415, no. 6868, pp. 141–147, 2002.

[173] Y. Ho, A. Gruhler, A. Heilbut, G. Bader, L. Moore, S. Adams, A. Millar, P. Taylor, K. Bennett, K. Boutilier, *et al.*, “Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry,” *Nature*, vol. 415, no. 6868, pp. 180–183, 2002.

[174] G. Rigaut, A. Shevchenko, B. Rutz, M. Wilm, M. Mann, and B. Séraphin, “A generic protein purification method for protein complex characterization and proteome exploration,” *Nature biotechnology*, vol. 17, no. 10, pp. 1030–1032, 1999.

[175] S. Bandyopadhyay, R. Kelley, N. Krogan, and T. Ideker, “Functional maps of protein complexes from quantitative genetic interaction data,” *PLoS Comput Biol*, vol. 4, no. 4, p. e1000065, 2008.

[176] G. Hannum, R. Srivas, A. Guénolé, H. Van Attikum, N. Krogan, R. Karp, and T. Ideker, “Genome-wide association data reveal a global map of genetic interactions among protein complexes,” *PLoS Genet*, vol. 5, no. 12, p. e1000782, 2009.

[177] C. Choudhary, C. Kumar, F. Gnad, M. Nielsen, M. Rehman, T. Walther, J. Olsen, and M. Mann, “Lysine acetylation targets protein complexes and co-regulates major cellular functions,” *Science’s STKE*, vol. 325, no. 5942, p. 834, 2009.

[178] V. Spirin and L. Mirny, “Protein complexes and functional modules in molecular networks,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 21, p. 12123, 2003.

[179] S. Johnson, H. Grosshans, J. Shingara, M. Byrom, R. Jarvis, A. Cheng, E. Labourier, K. Reinert, D. Brown, and F. Slack, “RAS is regulated by the let-7 microRNA family,” *Cell*, vol. 120, no. 5, pp. 635–647, 2005.

[180] N. Yanaihara, N. Caplen, E. Bowman, M. Seike, K. Kumamoto, M. Yi, R. Stephens, A. Okamoto, J. Yokota, T. Tanaka, *et al.*, “Unique microRNA molecular profiles in lung cancer diagnosis and prognosis,” *Cancer cell*, vol. 9, no. 3, pp. 189–198, 2006.

[181] J. Takamizawa, H. Konishi, K. Yanagisawa, S. Tomida, H. Osada, H. Endoh, T. Harano, Y. Yatabe, M. Nagino, Y. Nimura, *et al.*, “Reduced expression of the let-7 microRNAs in human lung cancers in association with shortened postoperative survival,” *Cancer Research*, vol. 64, no. 11, p. 3753, 2004.

[182] C. Mayr, M. Hemann, and D. Bartel, “Disrupting the pairing between let-7 and Hmga2 enhances oncogenic transformation,” *Science*, vol. 315, no. 5818, p. 1576, 2007.

[183] Y. Lee, K. Jeon, J. Lee, S. Kim, and V. Kim, “MicroRNA maturation: stepwise processing and subcellular localization,” *The EMBO journal*, vol. 21, no. 17, pp. 4663–4670, 2002.

[184] Y. Altuvia, P. Landgraf, G. Lithwick, N. Elefant, S. Pfeffer, A. Aravin, M. Brownstein, T. Tuschl, and H. Margalit, “Clustering and conservation patterns of human microRNAs,” *Nucleic acids research*, vol. 33, no. 8, p. 2697, 2005.

[185] G. Stefani and F. Slack, “Small non-coding RNAs in animal development,” *Nature Reviews Molecular Cell Biology*, vol. 9, no. 3, pp. 219–230, 2008.

[186] A. Dixon-McIver, P. East, C. Mein, J. Cazier, G. Molloy, T. Chaplin, T. Lister, B. Young, and S. Debernardi, “Distinctive patterns of microRNA expression associated with karyotype in acute myeloid leukaemia,” *PLoS One*, vol. 3, no. 5, p. 2141, 2008.

[187] Q. Huang, K. Gumireddy, M. Schrier, C. Le Sage, R. Nagel, S. Nair, D. Egan, A. Li, G. Huang, A. Klein-Szanto, *et al.*, “The microRNAs miR-373 and miR-520c promote tumour invasion and metastasis,” *Nature cell biology*, vol. 10, no. 2, pp. 202–210, 2008.

[188] K. To, R. Robey, T. Knutsen, Z. Zhan, T. Ried, and S. Bates, “Escape from hsa-miR-519c enables drug-resistant cells to maintain high expression of ABCG2,” *Molecular cancer therapeutics*, vol. 8, no. 10, p. 2959, 2009.

[189] J. Ren, P. Jin, E. Wang, F. Marincola, and D. Stroncek, “MicroRNA and gene expression patterns in the differentiation of human embryonic stem cells,” *Journal of Translational Medicine*, vol. 7, no. 1, p. 20, 2009.

[190] S. Mees, W. Mardin, C. Wendel, N. Baeumer, E. Willscher, N. Senninger, C. Schleicher, M. Colombo-Benkmann, and J. Haier, “EP300 – A miRNA-regulated metastasis suppressor gene in ductal adenocarcinomas of the pancreas,” *International Journal of Cancer*, vol. 126, no. 1, pp. 114–124, 2010.

[191] J. Piriyapongsa and I. Jordan, “A family of human microRNA genes from miniature inverted-repeat transposable elements,” *PLoS One*, vol. 2, no. 2, p. 203, 2007.

[192] D. Gibbons, W. Lin, C. Creighton, Z. Rizvi, P. Gregory, G. Goodall, N. Thilaganathan, L. Du, Y. Zhang, A. Pertsemlidis, *et al.*, “Contextual extracellular cues promote tumor cell EMT and metastasis by regulating miR-200 family expression,” *Genes & development*, vol. 23, no. 18, p. 2140, 2009.

[193] Y. Hayashita, H. Osada, Y. Tatematsu, H. Yamada, K. Yanagisawa, S. Tomida, Y. Yatabe, K. Kawahara, Y. Sekido, and T. Takahashi, “A polycistronic microRNA cluster, miR-17-92, is overexpressed in human lung cancers and enhances cell proliferation,” *Cancer research*, vol. 65, no. 21, p. 9628, 2005.

[194] M. Fabbri, R. Garzon, A. Cimmino, Z. Liu, N. Zanesi, E. Callegari, S. Liu, H. Alder, S. Costinean, C. Fernandez-Cymering, *et al.*, “MicroRNA-29 family reverts aberrant methylation in lung cancer by targeting DNA methyltransferases 3A and 3B,” *Proceedings of the National Academy of Sciences*, vol. 104, no. 40, p. 15805, 2007.

[195] R. Garzon, F. Pichiorri, T. Palumbo, M. Visentini, R. Aqeilan, A. Cimmino, H. Wang, H. Sun, S. Volinia, H. Alder, *et al.*, “MicroRNA gene expression dur-

ing retinoic acid-induced differentiation of human acute promyelocytic leukemia," *Oncogene*, vol. 26, no. 28, pp. 4148–4157, 2007.

[196] D. Bhaumik, G. Scott, S. Schokrpur, C. Patil, J. Campisi, and C. Benz, "Expression of microRNA-146 suppresses NF-κB activity with reduction of metastatic potential in breast cancer cells," *Oncogene*, vol. 27, no. 42, pp. 5643–5647, 2008.

[197] P. Gregory, A. Bert, E. Paterson, S. Barry, A. Tsykin, G. Farshid, M. Vadas, Y. Khew-Goodall, and G. Goodall, "The miR-200 family and miR-205 regulate epithelial to mesenchymal transition by targeting ZEB1 and SIP1," *Nature cell biology*, vol. 10, no. 5, pp. 593–601, 2008.

[198] J. Zhao, J. Lin, H. Yang, W. Kong, L. He, X. Ma, D. Coppola, and J. Cheng, "MicroRNA-221/222 negatively regulates estrogen receptorα and is associated with tamoxifen resistance in breast cancer," *Journal of Biological Chemistry*, vol. 283, no. 45, p. 31079, 2008.

[199] X. Hu, D. Macdonald, P. Huettner, Z. Feng, I. El Naqa, J. Schwarz, D. Mutch, P. Grigsby, S. Powell, and X. Wang, "A miR-200 microRNA cluster as prognostic marker in advanced ovarian cancer," *Gynecologic oncology*, vol. 114, no. 3, pp. 457–464, 2009.

[200] X. Chen, J. Gong, H. Zeng, N. Chen, R. Huang, Y. Huang, L. Nie, M. Xu, J. Xia, F. Zhao, *et al.*, "MicroRNA145 Targets BNIP3 and Suppresses Prostate Cancer Progression," *Cancer research*, vol. 70, no. 7, p. 2728, 2010.

[201] R. Prueitt, M. Yi, R. Hudson, T. Wallace, T. Howe, H. Yfantis, D. Lee, *et al.*, "Expression of microRNAs and protein-coding genes associated with perineural invasion in prostate cancer," *The Prostate*, vol. 68, no. 11, pp. 1152–1164, 2008.

[202] H. Matsubara, T. Takeuchi, E. Nishikawa, K. Yanagisawa, Y. Hayashita, H. Ebi, H. Yamada, M. Suzuki, M. Nagino, Y. Nimura, *et al.*, "Apoptosis induction by anti-

sense oligonucleotides against miR-17-5p and miR-20a in lung cancers overexpressing miR-17-92," *Oncogene*, vol. 26, no. 41, pp. 6099–6105, 2007.

[203] J. Foekens, A. Sieuwerts, M. Smid, M. Look, V. De Weerd, A. Boersma, J. Klijn, E. Wiemer, and J. Martens, "Four miRNAs associated with aggressiveness of lymph node-negative, estrogen receptor-positive human breast cancer," *Proceedings of the National Academy of Sciences*, vol. 105, no. 35, p. 13021, 2008.

[204] S. Reddy, K. Ohshiro, S. Rayala, and R. Kumar, "MicroRNA-7, a homeobox D10 target, inhibits p21-activated kinase 1 and regulates its functions," *Cancer research*, vol. 68, no. 20, p. 8195, 2008.

[205] R. Webster, K. Giles, K. Price, P. Zhang, J. Mattick, and P. Leedman, "Regulation of epidermal growth factor receptor signaling in human cancer cells by microRNA-7," *Journal of Biological Chemistry*, vol. 284, no. 9, p. 5731, 2009.

[206] P. Hsu, D. Deatherage, B. Rodriguez, S. Liyanarachchi, Y. Weng, T. Zuo, J. Liu, A. Cheng, and T. Huang, "Xenoestrogen-induced epigenetic repression of microRNA-9-3 in breast epithelial cells," *Cancer research*, vol. 69, no. 14, p. 5936, 2009.

[207] Y. Saito, G. Liang, G. Egger, J. Friedman, J. Chuang, G. Coetzee, and P. Jones, "Specific activation of microRNA-127 with downregulation of the proto-oncogene BCL6 by chromatin-modifying drugs in human cancer cells," *Cancer cell*, vol. 9, no. 6, pp. 435–443, 2006.

[208] X. Li, P. Yan, and Z. Shao, "Downregulation of miR-193b contributes to enhance urokinase-type plasminogen activator (uPA) expression and tumor progression and invasion in human breast cancer," *Oncogene*, vol. 28, no. 44, pp. 3937–3948, 2009.

[209] V. Findlay, D. Turner, O. Moussa, and D. Watson, "MicroRNA-mediated inhibition of prostate-derived Ets factor messenger RNA translation affects prostate-derived

Ets factor regulatory networks in human breast cancer," *Cancer Research*, vol. 68, no. 20, p. 8499, 2008.

[210] K. Taganov, M. Boldin, K. Chang, and D. Baltimore, "NF-κB-dependent induction of microRNA miR-146, an inhibitor targeted to signaling proteins of innate immune responses," *Proceedings of the National Academy of Sciences*, vol. 103, no. 33, p. 12481, 2006.

[211] J. Shen, C. Ambrosone, R. Dicioccio, K. Odunsi, S. Lele, and H. Zhao, "A functional polymorphism in the miR-146a gene and age of familial breast/ovarian cancer diagnosis," *Carcinogenesis*, 2008.

[212] S. Tavazoie, C. Alarcón, T. Oskarsson, D. Padua, Q. Wang, P. Bos, W. Gerald, and J. Massagué, "Endogenous human microRNAs that suppress breast cancer metastasis," *Nature*, vol. 451, no. 7175, pp. 147–152, 2008.

[213] I. Guttilla and B. White, "Coordinate regulation of FOXO1 by miR-27a, miR-96, and miR-182 in breast cancer cells," *Journal of Biological Chemistry*, vol. 284, no. 35, p. 23204, 2009.

[214] V. Tryndyak, F. Beland, and I. Pogribny, "E-cadherin transcriptional down-regulation by epigenetic and microRNA-200 family alterations is related to mesenchymal and drug-resistant phenotypes in human breast cancer cells," *International Journal of Cancer*, vol. 126, no. 11, pp. 2575–2583, 2010.

[215] D. Cochrane, N. Spoelstra, E. Howe, S. Nordeen, and J. Richer, "MicroRNA-200c mitigates invasiveness and restores sensitivity to microtubule-targeting chemotherapeutic agents," *Molecular cancer therapeutics*, vol. 8, no. 5, p. 1055, 2009.

[216] Y. Shimono, M. Zabala, R. Cho, N. Lobo, P. Dalerba, D. Qian, M. Diehn, H. Liu, S. Panula, E. Chiao, *et al.*, "Downregulation of miRNA-200c links breast cancer stem cells with normal stem cells," *Cell*, vol. 138, no. 3, pp. 592–603, 2009.

[217] A. Chao, C. Tsai, P. Wei, S. Hsueh, A. Chao, C. Wang, C. Tsai, Y. Lee, T. Wang, and C. Lai, “Decreased expression of microRNA-199b increases protein levels of SET (protein phosphatase 2A inhibitor) in human choriocarcinoma,” *Cancer letters*, vol. 291, no. 1, pp. 99–107, 2010.

[218] T. Li, D. Li, J. Sha, P. Sun, and Y. Huang, “MicroRNA-21 directly targets MARCKS and promotes apoptosis resistance and invasion in prostate cancer cells,” *Biochemical and biophysical research communications*, vol. 383, no. 3, pp. 280–285, 2009.

[219] J. Ribas, X. Ni, M. Haffner, E. Wentzel, A. Salmasi, W. Chowdhury, T. Kudrolli, S. Yegnasubramanian, J. Luo, R. Rodriguez, *et al.*, “miR-21: An Androgen Receptor–Regulated MicroRNA that Promotes Hormone-Dependent and Hormone-Independent Prostate Cancer Growth,” *Cancer research*, vol. 69, no. 18, p. 7165, 2009.

[220] S. Varambally, Q. Cao, R. Mani, S. Shankar, X. Wang, B. Ateeq, B. Laxman, X. Cao, X. Jing, K. Ramnarayanan, *et al.*, “Genomic loss of microRNA-101 leads to overexpression of histone methyltransferase EZH2 in cancer,” *Science*, vol. 322, no. 5908, p. 1695, 2008.

[221] H. Huang, “Commentary on Genomic loss of microRNA-101 leads to overexpression of histone methyltransferase EZH2 in cancer:: Varambally S, Cao Q, Mani RS, Shankar S, Wang X, Ateeq B, Laxman B, Cao X, Jing X, Ramnarayanan K, Brenner JC, Yu J, Kim JH, Han B, Tan P, Kumar-Sinha C, Lonigro RJ, Palanisamy N, Maher CA, Chinnaiyan AM, Michigan Center for Translational Pathology, University of Michigan Medical School, Ann Arbor, MI 48109,” in *Urologic Oncology: Seminars and Original Investigations*, vol. 27, no. 2. Elsevier, 2009, p. 230.

[222] P. Cao, Z. Deng, M. Wan, W. Huang, S. Cramer, J. Xu, M. Lei, and G. Sui, “MicroRNA-101 negatively regulates Ezh 2 and its expression is modulated by an-

drogen receptor and HIF-1 α/HIF-1 β," *Molecular Cancer*, vol. 9, no. 1, p. 108, 2010.

[223] G. Wang, W. Mao, and S. Zheng, "MicroRNA-183 regulates Ezrin expression in lung cancer cells," *FEBS letters*, vol. 582, no. 25-26, pp. 3663–3668, 2008.

[224] T. Melkamu, X. Zhang, J. Tan, Y. Zeng, and F. Kassie, "Alteration of microRNA expression in vinyl carbamate-induced mouse lung tumors and modulation by the chemopreventive agent indole-3-carbinol," *Carcinogenesis*, vol. 31, no. 2, p. 252, 2010.

[225] M. Crawford, K. Batte, L. Yu, X. Wu, G. Nuovo, C. Marsh, G. Otterson, and S. Nana-Sinkam, "MicroRNA 133B targets pro-survival molecules MCL-1 and BCL2L2 in lung cancer," *Biochemical and biophysical research communications*, vol. 388, no. 3, pp. 483–489, 2009.

[226] I. Barshack, G. Lithwick-Yanai, A. Afek, K. Rosenblatt, H. Tabibian-Keissar, M. Zepeniuk, L. Cohen, H. Dan, O. Zion, Y. Strenov, *et al.*, "MicroRNA expression differentiates between primary lung tumors and metastases to the lung," *Pathology-Research and Practice*, 2010.

[227] L. Du, J. Schageman, M. Subauste, B. Saber, S. Hammond, L. Prudkin, I. Wistuba, L. Ji, J. Roth, J. Minna, *et al.*, "miR-93, miR-98, and miR-197 regulate expression of tumor suppressor gene FUS1," *Molecular Cancer Research*, vol. 7, no. 8, p. 1234, 2009.

[228] Z. Li, J. Lu, M. Sun, S. Mi, H. Zhang, R. Luo, P. Chen, Y. Wang, M. Yan, Z. Qian, *et al.*, "Distinct microRNA expression profiles in acute myeloid leukemia with common translocations," *Proceedings of the National Academy of Sciences*, vol. 105, no. 40, p. 15535, 2008.

[229] S. Mi, J. Lu, M. Sun, Z. Li, H. Zhang, M. Neilly, Y. Wang, Z. Qian, J. Jin, Y. Zhang, *et al.*, "MicroRNA expression signatures accurately discriminate acute lymphoblas-

tic leukemia from acute myeloid leukemia," *Proceedings of the National Academy of Sciences*, vol. 104, no. 50, p. 19971, 2007.

[230] R. Garzon, M. Garofalo, M. Martelli, R. Briesewitz, L. Wang, C. Fernandez-Cymering, S. Volinia, C. Liu, S. Schnittger, T. Haferlach, *et al.*, "Distinctive microRNA signature of acute myeloid leukemia bearing cytoplasmic mutated nucleophosmin," *Proceedings of the National Academy of Sciences*, vol. 105, no. 10, p. 3945, 2008.

[231] A. Saumet, G. Vetter, M. Bouttier, E. Portales-Casamar, W. Wasserman, T. Maurin, B. Mari, P. Barbry, L. Vallar, E. Friederich, *et al.*, "Transcriptional repression of microRNA genes by PML-RARA increases expression of key cancer proteins in acute promyelocytic leukemia," *Blood*, vol. 113, no. 2, p. 412, 2009.

[232] D. Schaar, D. Medina, D. Moore, R. Strair, and Y. Ting, "miR-320 targets transferrin receptor 1 (CD71) and inhibits cell proliferation," *Experimental hematology*, vol. 37, no. 2, pp. 245–255, 2009.

[233] M. Pigazzi, E. Manara, E. Baron, and G. Basso, "miR-34b Targets Cyclic AMP–Responsive Element Binding Protein in Acute Myeloid Leukemia," *Cancer research*, vol. 69, no. 6, p. 2471, 2009.

[234] L. Ma, J. Teruya-Feldstein, and R. Weinberg, "Tumour invasion and metastasis initiated by microRNA-10b in breast cancer," *Nature*, vol. 449, no. 7163, pp. 682–688, 2007.

[235] M. Iorio, M. Ferracin, C. Liu, A. Veronese, R. Spizzo, S. Sabbioni, E. Magri, M. Pedriali, M. Fabbri, M. Campiglio, *et al.*, "MicroRNA gene expression deregulation in human breast cancer," *Cancer research*, vol. 65, no. 16, p. 7065, 2005.

[236] J. Zhang, Y. Du, Y. Lin, Y. Chen, L. Yang, H. Wang, and D. Ma, "The cell growth suppressor, mir-126, targets IRS-1," *Biochemical and biophysical research communications*, vol. 377, no. 1, pp. 136–140, 2008.

[237] U. Lehmann, B. Hasemeier, M. Christgen, M. Müller, D. Römermann, F. Länger, and H. Kreipe, “Epigenetic inactivation of microRNA gene hsa-mir-9-1 in human breast cancer,” *The Journal of pathology*, vol. 214, no. 1, pp. 17–24, 2008.

[238] U. Lehmann, B. Hasemeier, D. Römermann, M. Müller, F. Länger, and H. Kreipe, “Epigenetic inactivation of microRNA genes in mammary carcinoma,” *Verhandlungen der Deutschen Gesellschaft für Pathologie*, vol. 91, p. 214, 2007.

[239] L. Yan, X. Huang, Q. Shao, M. Huang, L. Deng, Q. Wu, Y. Zeng, and J. Shao, “MicroRNA miR-21 overexpression in human breast cancer is associated with advanced clinical stage, lymph node metastasis and patient poor prognosis,” *Rna*, vol. 14, no. 11, p. 2348, 2008.

[240] W. Kong, H. Yang, L. He, J. Zhao, D. Coppola, W. Dalton, and J. Cheng, “MicroRNA-155 is regulated by the transforming growth factor {beta}/Smad pathway and contributes to epithelial cell plasticity by targeting RhoA,” *Molecular and cellular biology*, vol. 28, no. 22, p. 6773, 2008.

[241] M. Iorio, P. Casalini, C. Piovan, G. Di Leva, A. Merlo, T. Triulzi, S. Ménard, C. Croce, and E. Tagliabue, “microRNA-205 regulates HER3 in human breast cancer,” *Cancer research*, vol. 69, no. 6, p. 2195, 2009.

[242] D. Pandey and D. Picard, “miR-22 inhibits estrogen signaling by directly targeting the estrogen receptor {alpha} mRNA,” *Molecular and cellular biology*, vol. 29, no. 13, p. 3783, 2009.

[243] S. Mertens-Talcott, S. Chintharlapalli, X. Li, and S. Safe, “The oncogenic microRNA-27a targets genes that regulate specificity protein transcription factors and the G2-M checkpoint in MDA-MB-231 breast cancer cells,” *Cancer research*, vol. 67, no. 22, p. 11001, 2007.

[244] S. Reddy, S. Pakala, K. Ohshiro, S. Rayala, and R. Kumar, “MicroRNA-661, ac/EBPα Target, Inhibits Metastatic Tumor Antigen 1 and Regulates Its Functions,” *Cancer research*, vol. 69, no. 14, p. 5639, 2009.

[245] Y. Pan, M. Morris, and A. Yu, “MicroRNA-328 negatively regulates the expression of breast cancer resistance protein (BCRP/ABCG2) in human cancer cells,” *Molecular pharmacology*, vol. 75, no. 6, p. 1374, 2009.

[246] M. Ozen, C. Creighton, M. Ozdemir, and M. Ittmann, “Widespread deregulation of microRNA expression in human prostate cancer,” *Oncogene*, vol. 27, no. 12, pp. 1788–1793, 2007.

[247] K. Porkka, M. Pfeiffer, K. Waltering, R. Vessella, T. Tammela, and T. Visakorpi, “MicroRNA expression profiling in prostate cancer,” *Cancer research*, vol. 67, no. 13, p. 6130, 2007.

[248] A. Musiyenko, V. Bitko, and S. Barik, “Ectopic expression of miR-126*, an intronic product of the vascular endothelial EGF-like 7 gene, regulates prostein translation and invasiveness of prostate cancer LNCaP cells,” *Journal of Molecular Medicine*, vol. 86, no. 3, pp. 313–322, 2008.

[249] S. Lin, A. Chiang, D. Chang, and S. Ying, “Loss of mir-146a function in hormone-refractory prostate cancer,” *Rna*, vol. 14, no. 3, p. 417, 2008.

[250] D. Bonci, V. Coppola, M. Musumeci, A. Addario, R. Giuffrida, L. Memeo, L. D’Urso, A. Pagliuca, M. Biffoni, C. Labbaye, *et al.*, “The miR-15a–miR-16-1 cluster controls prostate cancer by targeting multiple oncogenic activities,” *Nature medicine*, vol. 14, no. 11, pp. 1271–1277, 2008.

[251] A. Schaefer, M. Jung, H. Mollenkopf, I. Wagner, C. Stephan, F. Jentzmik, K. Miller, M. Lein, G. Kristiansen, and K. Jung, “Diagnostic and prognostic implications of microRNA profiling in prostate carcinoma,” *International Journal of Cancer*, vol. 126, no. 5, pp. 1166–1176, 2010.

[252] S. Galardi, N. Mercatelli, E. Giorda, S. Massalini, G. Frajese, S. Ciafrè, and M. Farace, “miR-221 and miR-222 expression affects the proliferation potential of human prostate carcinoma cell lines by targeting p27Kip1,” *Journal of Biological Chemistry*, vol. 282, no. 32, p. 23716, 2007.

[253] A. Tong, P. Fulgham, C. Jay, P. Chen, I. Khalil, S. Liu, N. Senzer, A. Eklund, J. Han, and J. Nemunaitis, “MicroRNA profile analysis of human prostate cancers,” *Cancer gene therapy*, vol. 16, no. 3, pp. 206–216, 2008.

[254] K. Lee, Y. Chen, S. Yeh, M. Hsiao, J. Lin, Y. Goan, and P. Lu, “MicroRNA-330 acts as tumor suppressor and induces apoptosis of prostate cancer cells through E2F1-mediated suppression of Akt phosphorylation,” *Oncogene*, vol. 28, no. 38, pp. 3360–3370, 2009.

[255] Y. Fujita, K. Kojima, N. Hamada, R. Ohhashi, Y. Akao, Y. Nozawa, T. Deguchi, and M. Ito, “Effects of miR-34a on cell growth and chemoresistance in prostate cancer PC3 cells,” *Biochemical and biophysical research communications*, vol. 377, no. 1, pp. 114–119, 2008.

[256] S. Josson, S. Sung, K. Lao, L. Chung, and P. Johnstone, “Radiation modulation of microRNA in prostate cancer cell lines,” *The Prostate*, vol. 68, no. 15, pp. 1599–1606, 2008.

[257] D. Corney, A. Flesken-Nikitin, A. Godwin, W. Wang, and A. Nikitin, “MicroRNA-34b and MicroRNA-34c are targets of p53 and cooperate in control of cell proliferation and adhesion-independent growth,” *Cancer research*, vol. 67, no. 18, p. 8433, 2007.

[258] N. Dahiya, C. Sherman-Baust, T. Wang, B. Davidson, I. Shih, Y. Zhang, W. Wood III, K. Becker, and P. Morin, “MicroRNA expression and identification of putative miRNA targets in ovarian cancer,” *PLoS One*, vol. 3, no. 6, p. 2436, 2008.

[259] Y. *et al*. Xu, "MicroRNA prediction with a novel ranking algorithm based on random walks," *Bioinformatics*, vol. 24, pp. 50–58, 2008.

[260] J. *et al*. Nam, "Human microRNA prediction through a probabilistic co-learning model of sequence and structure," *Nucl. Acids Res.*, vol. 33, pp. 3570–3581, 2005.

[261] K. Spackman, "Signal detection theory: Valuable tools for evaluating inductive learning," *Proceedings of the Sixth International Workshop on Machine Learning*, pp. 160–163, 1989.

[262] D. Cozma, L. Lukes, J. Rouse, T. Qiu, E. Liu, and K. Hunter, "A bioinformatics-based strategy identifies c-Myc and Cdc25A as candidates for the Apmt mammary tumor latency modifiers," *Genome research*, vol. 12, no. 6, p. 969, 2002.

[263] T. Wu, "Analysing gene expression data from DNA microarrays to identify candidate genes," *The Journal of Pathology*, vol. 195, no. 1, pp. 53–65, 2001.

[264] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, p. 2507, 2007.

[265] A. Vecchione and C. Croce, "Apoptomirs: small molecules have gained the license to kill," *Endocrine-related cancer*, 2009.

[266] B. Schwikowski, P. Uetz, and S. Fields, "A network of protein–protein interactions in yeast," *Nature Biotechnology*, vol. 18, no. 12, pp. 1257–1261, 2000.

[267] *et al*. Griffiths-Jones S, "MiRBase: tools for microRNA genomics," *NAR*, vol. 36(Database Issue), pp. D154–D158, 2008.

[268] M. Lu, Q. Zhang, M. Deng, J. Miao, Y. Guo, W. Gao, and Q. Cui, "An analysis of human microRNA and disease associations," *PLoS One*, vol. 3, no. 10, p. 3420, 2008.

[269] K. Goh, M. Cusick, D. Valle, B. Childs, M. Vidal, and A. Barabási, "The human disease network," *Proceedings of the National Academy of Sciences*, vol. 104, no. 21, p. 8685, 2007.

[270] D. P. Bartel, “MicroRNAs: genomics, biogenesis, mechanism, and function,” *Cell*, vol. 116, pp. 281–297, 2004.

[271] F. Gobel and A. A. Jagers, “Random walks on graphs,” *Stochastic processes and their applications*, vol. 2, pp. 311–336, 1974.

[272] C. Ding, “A learning framework using green’s function and kernel regularization with application to recommender system,” *KDD*, pp. 260–269, 2007.

[273] MicroCosm, “http://www.ebi.ac.uk/enright-srv/microcosm/cgi-bin/targets/v5/download.pl,” 2009.

[274] I. Hofacker, “Vienna RNA secondary structure server,” *Nucl. Acids Res.*, vol. 31, pp. 3429–431, 2003.

BIOGRAPHICAL STATEMENT

Dijun Luo is currently a PhD candidate in Computer Science and Engineering Department, University of Texas at Arlington. He received B.S. and M.S. degrees in Zhejiang University in 2004 and 2006, respectively. Dijuns current research interests include Machine Learning, Data Mining, Medical Informatics, and Computer Vision. Dijun has published 25 research papers in top conferences and journals, such as ICML, KDD, CVPR, ICCV, and Machine Learning Journal etc. among which 2 papers received best paper award and one received best paper finalis. Dijun has served as PC member for 5 conferences/workshops and reviewer for 8 journals in areas of Machine Learning and Data Mining. He also served as research intern in IBM T. J. Watson Research Center during his PhD study. Prior to his PhD, Dijun was the founder of ZJUBase and co-founder of Minitech.