

Spectral Feature Selection for Mining

Ultrahigh Dimensional Data

by

Zheng Zhao

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

ARIZONA STATE UNIVERSITY

August 2010

Spectral Feature Selection for Mining

Ultrahigh Dimensional Data

by

Zheng Zhao

has been approved

June 2010

Graduate Supervisory Committee:

Huan Liu, Co-Chair

Jieping Ye, Co-Chair

Subbarao Kambhampati

Guoliang Xue

ACCEPTED BY THE GRADUATE COLLEGE

ABSTRACT

The rapid advance of computer-based high-throughput technology and the ubiquitous use of the web have provided unparalleled opportunities for humans to expand their capabilities in production, services, communications, and research. In this process, immense quantities of high-dimensional data are accumulated, challenging the state-of-the-art machine learning techniques to efficiently produce useful results. Feature selection can effectively reduce data dimensionality by removing irrelevant and redundant features. It brings the immediate effects of speeding up data mining algorithms, and improving mining performance such as predictive accuracy and result comprehensibility. In the last decade, a large amount of relevance criteria have been developed to evaluate the utility of features in feature selection, and these criteria are largely studied separately according to the type of learning: supervised or unsupervised. This dissertation studies spectral feature selection, which is a novel general feature selection framework based on graph spectral analysis. It unifies both supervised and unsupervised feature selection, and can generate families of algorithms for both learning contexts. It also includes many existing algorithms as its special cases and allows their joint study to gain insights. A common issue of many existing algorithms is that they are univariate method, and thus cannot handle redundant features. The proposed spectral feature selection framework can be readily extended to conduct multivariate analysis for addressing the limitation effectively. One of the most challenging problems in feature selection research is the small sample problem, in which the lack of information further worsens the situation of high dimensionality. Spectral feature selection also provides a natural way to include domain knowledge from multiple sources to enrich information and address this problem. The resulted multi-source feature selection technique represents one of the latest development trends in feature selection research. Extensive experimental study is conducted and results demonstrate that spectral feature selection achieves superior performance in various learning contexts.

To My Lovely Wife And My Dear Parents

ACKNOWLEDGEMENTS

It has been a great pleasure working with faculty, staff, and students at the Arizona State university during my tenure as a doctoral student. This would have never been possible without the freedom I was given to pursue my research interests by my mentor and advisor, Prof. Huan Liu. I would like to thank him for his strong belief in me, invaluable feedback, kind advice, and incessant support all these years. I would also like to thank Prof. Jieping Ye, who is my co-advisor and committee co-chair, for his precious instruction and consistent help in the last five years.

My research has also benefited tremendously from various collaborations over the years. I would particularly like to thank Prof. Guoliang Xue (at Arizona State University), Prof. Subbarao Kambhampati (at Arizona State University), Prof. Yung Chang (at Arizona State University), Dr. Jiangxin Wang (at Arizona State University), Dr. Lei Wang (at Australian National University), Dr. Kari Torkkola (at Amazon.com), and Dr. Keshu Zhang (at Motorola Research Lab) for many thoughtful conversations.

In large part, my dissertation research has been sponsored by the National Science Foundation (NSF, Grant No. 0812551), the Graduate & Professional Student Association at Arizona State University, the government grants through the sponsorship from Mr. Leonard H. Montenegro, and the scholarships from the Computer Science and Engineering at Arizona State University.

TABLE OF CONTENTS

	Page
TABLE OF CONTENTS	vi
LIST OF FIGURES	viii
LIST OF TABLES	ix
1 INTRODUCTION	1
1.1 Spectral Feature Selection	4
1.2 Notations	9
2 FEATURE SELECTION VIA SPECTRAL ANALYSIS	12
2.1 Ranking Features on Graph	13
2.2 An Extension for Feature Ranking Functions	17
2.3 SPEC, A Unified Framework for Spectral Feature Selection	18
2.4 Robustness Analysis for SPEC	20
2.5 Study on Synthetic Data	26
2.6 Study on Real Data	30
2.6.1 Study of Unsupervised Cases	31
2.6.2 Study of Supervised Cases	34
2.7 Discussions	35
3 CONNECTIONS TO EXISTING ALGORITHMS	40
3.1 Reformulation for SPEC	41
3.2 Reformulation for Laplacian Score	44
3.3 Reformulation for Fisher Score	45
3.4 Reformulation of ReliefF	46
3.5 Reformulation for Trace Ratio Criterion	48
3.6 Reformulation for HSIC Criterion	50
3.7 Discussion	51
4 A MULTIVARIATE FORMULATION FOR SPECTRAL FEATURE SELECTION	52

Chapter	Page
4.1 Spectral Feature Selection with Sparse Multi-output Regression	53
4.2 MRSE, An Efficient Solver	57
4.3 Experimental Study	60
4.3.1 Study of Supervised Cases	62
4.3.2 Study of Unsupervised Cases	62
4.3.3 Study of Efficiency	63
4.4 discussion	63
5 INTEGRATING MULTIPLE KNOWLEDGE SOURCES IN FEATURE SELEC-	
TION	71
5.1 Categorization of Different Types of Knowledge	74
5.2 Knowledge Conversion	78
5.2.1 $\mathcal{H}_{SIM}^F \rightarrow \mathcal{H}_{SIM}^S$	78
5.2.2 $\mathcal{H}_{FUN}^F, \mathcal{H}_{CON}^F \rightarrow \mathcal{H}_{SIM}^S$	79
5.3 MSFS - The Framework	80
5.4 Experimental Study on Human Cancer Data	81
5.4.1 Results on 2C-DATA	84
5.4.2 Results on 4C-DATA	85
5.4.3 Incorporating Label Information	88
5.4.4 A further Study of Gene Biological Relevance	89
5.5 Discussion	92
6 CONCLUSION	95
BIBLIOGRAPHY	

LIST OF FIGURES

Figure	Page
2.1 An exemplification of the basic idea behind spectral feature selection.	13
2.2 Contours of the eigenvectors $\xi_1, \xi_2, \xi_3, \xi_4, \xi_5$, and ξ_{20}	16
2.3 Empirical study of the effect of perturbation on SPEC.	27
2.4 Study of unsupervised cases: accuracy (y axis) vs. different numbers of selected features (x axis). In the legend, F_i stands for $\hat{\phi}_i(\cdot)$	33
2.5 Study of supervised cases: accuracy (y axis) vs. different numbers of features (x axis). In the legend, F_i stands for $\hat{\phi}_i(\cdot)$. Y stands for the similarity matrix specified in Eq. (3.13), and L0 stands for AROM-SVM.	36
2.6 An exemplification of the idea behind semi-supervised feature selection.	39
4.1 The Running time of MRSF and the Nesterov method on each data set.	64
4.2 Study of supervised cases, plots for accuracy (y axis) vs. different numbers of selected features (x axis) on the six data sets. The higher the accuracy the better.	68
5.1 The framework of multi-source spectral feature selection.	73
5.2 An example of three different types of knowledge about genes (features): (a) Metabolic Pathway, (b) Gene Ontology Annotation, and (c) Gene Sequence.	75
5.3 Different types of knowledge about samples, (a) the class label information, (b) sample hierarchy, and (c) an example of the auxiliary data.	76
5.4 Calculating sample similarity using different types of knowledge of genes.	78
5.5 Charts (a,b,d): accuracy (y axis) vs. different numbers of genes (x axis). Chart (c): accuracy (y axis) vs. different combination coefficient (x axis).	86
5.6 KOGS, a framework for knowledge-oriented multi-source gene selection.	93

LIST OF TABLES

Table	Page
1.1 A summary of the notations used in the paper	11
2.1 The components for SPEC tried in this work. S_u and S_s stand for the similarity matrices used in unsupervised and supervised feature selection respectively.	20
2.2 Summary of six benchmark data sets used in the experiment.	31
2.3 Study of unsupervised cases: averaged accuracy. DIJ stands for Dijkstra kernel, and LPSR for Laplacian Score. Accuracy with bold face is the highest or the second highest one without significant difference with the highest.	34
2.4 Study of supervised cases: averaged accuracy. Accuracy with bold face is the highest or the second highest without significant difference with the highest.	35
2.5 Average accuracy improvements of the <i>1nn</i> classifier on the features selected by <i>sSelect</i> on the BASEHOCK data set. <i>L</i> is for number of labeled data.	39
3.1 The normalized similarity matrix and feature vector used in different algorithms.	51
4.1 Summary of the benchmark data sets	61
4.2 Study of supervised cases: aggregated accuracy, the higher the better. The number in the parentheses is the <i>p</i> -Val obtained from t-test.	69
4.3 Study of supervised cases: averaged redundancy rate, the lower the better. The number in the parentheses is the <i>p</i> -Val obtained from t-test.	69
4.4 Study of unsupervised cases: averaged Jaccard score, the higher the better. The number in the parentheses is the <i>p</i> -Val obtained from t-test.	70
4.5 Study of unsupervised cases: averaged redundancy rate, the lower the better. The number in the parentheses is the <i>p</i> -Val obtained from t-test.	70
5.1 Biologically relevant genes identified by two algorithms for childhood ALL.	72
5.2 The categories and examples of different types of knowledge.	77
5.3 A summary of the Human Cancer Data.	83

Table	Page
5.4 Results of accuracy and hit ratio. The numbers with bold typeface indicate the highest accuracy or hit ratio.	87
5.5 Upper: accuracy and hit ratio by supervised algorithms. <i>MSFS</i> with COMB-2 is unsupervised and listed for comparison. Lower: accuracy on 4C-DATA with genes selected by from 2C-DATA.	89
5.6 The top 20 genes selected by <i>MSFS</i> with COMB-2 on 2C-DATA. Genes with boldface names are known to be cancer related. Genes are ordered based on their relevance from highest to lowest.	91

Chapter 1

INTRODUCTION

The high dimensionality of data poses challenges to learning tasks due to the curse of dimensionality. In the presence of many irrelevant features, learning models tend to overfit and become less comprehensible. Feature selection is an important and frequently used technique in data mining for dimension reduction via removing irrelevant and redundant noisy. It brings the immediate effects of speeding up a data mining algorithm, improving learning accuracy, and enhancing model comprehensibility. Various studies show that features can be removed without performance deterioration [61, 17]. Feature selection has been an active area for decades in data mining, and has been widely applied to many research fields such as genomic analysis [35], text mining [22], image retrieval [26, 93], intrusion detection [45], to name a few. Given a rich literature exists for feature selection research, a systematical summarization and comparison studies are of necessity to facilitate the research and application of feature selection techniques. Recently, there have been many surveys published to serve this purpose. A comprehensive surveys of existing feature selection techniques and a general framework for their unification can be found in [50]. In [28], the authors reviewed feature selection algorithms from statistical learning point of view. In [72], the authors provided a good survey for applying feature selection techniques in bioinformatics. In [35], the authors reviewed and compared the filter with the wrapper model for feature selection. In [57], the authors explored the representative feature selection approaches based on sparse regularization, which is a branch of embedded feature selection techniques. Representative feature selection algorithms are also empirically evaluated in [52, 47, 90, 43, 56, 92, 59] under different problem settings and from different perspectives to provide insights on existing feature selection algorithms.

In the process of feature selection, the training data can be either labeled, unlabeled or partial labeled, leading to the development of supervised, unsupervised and semi-supervised feature selection algorithms. In the evaluation process, a supervised feature selection algorithm [80, 96, 85, 105] determines features' relevance by evaluating their correlation with the class or their utility for achieving accurate predication, and without labels, an unsupervised feature selection algorithm may exploit data variance or data distribution to evaluate features' relevance [20, 31]. A semi-supervised feature selection algorithm [110, 98] can use both labeled and unlabeled data. And the motivation of semi-supervised feature selection is to use a small amount of labeled data as additional information to improve the performance of unsupervised feature selection.

Feature selection algorithms designed with different strategies broadly fall into three categories: filter, wrapper and embedded models. To evaluate the utility of features, in the evaluation step, feature selection algorithms with filter model rely on analyzing the general characteristics of data and evaluating features without involving any learning algorithm. On the other hand, feature selection algorithms with wrapper model require a predetermined learning algorithm and uses its performance on the provided features in the evaluation step to identify relevant feature. Algorithms with embedded model, e.g., C4.5 [67], LARS [21], 1-norm support vector machine [115], and sparse logistic regression [10], incorporate feature selection as a part of the training process, and features' utility is obtained based on analyzing their utility for optimizing the objective function of the learning model. Compared to wrapper and embedded models, feature selection algorithms with filter model are independent to any learning model, therefore do not have bias on specific learner models, which is believed to be one advantage of the filter model. Another advantage of the filter model is that it has very simple structure, which usually contains a straightforward search strategy, such as backward elimination or forward selection, and a feature evaluation criterion designed according to certain criteria. The benefits of the simple structure are two

folders. First, it is easy to design, and after being implemented, it is also easy to understand for other researchers. This explains that why most feature selection algorithms are of filter model. Second, since its structure is simple, it is usually very fast. On the other hand, researcher also recognized that compared to the filter model, feature selection algorithms of wrapper and embedded models can usually select features that result in higher learning performance for the particular learning algorithms involved in the feature selection models. Comparing with wrapper model, feature selection algorithms of embedded model are usually more efficient, since they look into the structure of the learning algorithms and use their properties to guide feature evaluation and searching. In recent years, embedded model is gaining increasing interests in feature selection research due to its superior performance. Currently, most embedded feature selection algorithms are designed by applying L_0 norm [96, 33] or L_1 norm [53, 115, 113] constraint to existing learning models, such as support vector machine, logistic regression, and principle component analysis, to achieve sparse solution. When the constraint is of the form of L_1 , and the original problem is convex, existing optimization techniques can be applied to compute a unique global optimal solution for the problem efficiently [53].

Feature selection algorithms with filter and embedded models may return either a subset of selected features or the weights (measuring features' relevance) of all features. According to the type of the output, feature selection algorithms can be divided into either feature weighting algorithms or subset selection algorithms. Feature selection algorithms of wrapper model usually return feature subsets, therefore are subset selection algorithms. To the best of our knowledge, currently, most feature selection algorithms are designed to handle learning tasks with single data source, although the capability of using multiple data and knowledge sources in multi-source feature selection [111] may effectively enrich our information and enhance the reliability of relevance estimation [54, 109, 112].

1.1 Spectral Feature Selection

The chasm between supervised and unsupervised feature selection seems difficult to close as one works with class labels and the other does not. However, if we change the perspective and put less focus on class information, both supervised and unsupervised feature selection can be viewed as an effort to select features that are consistent with the target concept. In supervised learning the target concept is related to class affiliation, while in unsupervised learning the target concept is usually related to the innate cluster structures of the data. Essentially, in both cases, the target concept is related to partitioning samples into well separable subsets according to certain type of class or clustering affiliation. The feature evaluation criterion is a pivotal component of feature selection. It can take various forms: separability, information, dependency, consistency, learning model performance (used in wrapper model), and so on [50, 51]. In our recent study, we observe that a number of existing feature selection algorithms are essentially based on assessing features' capability of preserving sample similarity, which can be inferred from either label information or predefined distance metric. The representative feature selection algorithms in this category includes: Relief and ReliefF [80], Laplacian Score [31], Fisher Score [19], HSIC [85], and Trace Ratio [62]. These algorithms are designed to achieve different goals, for example, Fisher Score and ReliefF are designed to optimize sample separability, Laplacian Score is designed to retain sample locality, and HSIC is designed to maximize feature-class dependency. However, it turns out that all these algorithms actually select features to achieve sample similarity preservation via certain ways.

Pairwise sample similarity is widely used in both supervised and unsupervised learning to describe the relationships among samples. It forms an effective way to depict either sample cluster affiliation or sample class affiliation. For example, assume s_{ij} is the similarity between the i -th and the j -th samples, without class label information, a popular similarity

measurements is the *RBF* kernel function, which is defined as:

$$s_{ij} = e^{-\frac{\|x_1 - x_2\|^2}{(2\delta^2)}} \quad (1.1)$$

The function ensures samples which are from the same cluster have large similarity to each other and samples which are from different clusters have small similarity. On the other hand, when class label information is available, the sample similarity can be measured by:

$$s_{ij} = \begin{cases} \frac{1}{n_l}, & y_i = y_j = l \\ 0, & \text{otherwise} \end{cases} \quad (1.2)$$

where n_l denotes the number of samples in class l . This measurement ensures that samples from the same class have a nonnegative similarity, while samples from different classes have a zero similarity. Given n samples, the $n \times n$ matrix \mathbf{S} , is called a sample similarity matrix, when it contains the similarity of every sample pair, $\mathbf{S}_{ij} = s_{ij}$. \mathbf{S} is also call a kernel matrix, when any of its submatrix is also semi-positive-definite [74].

Given \mathbf{S} , selecting features that can preserve the sample similarity specified in \mathbf{S} helps retain sample affiliations in the dimensionality reduced space, therefore forms an effective way to identify relevant features that support the target concept. Also as mentioned above, pairwise sample similarity provides a common way to depict sample affiliation for both supervised and unsupervised learning, thus feature selection algorithms based on similarity preservation is applicable in both supervised and unsupervised learning context.

Given a sample similarity matrix \mathbf{S} , the spectrum of \mathbf{S} can be utilized to generate a low-dimensional space, where the sample similarity is best preserved [73]. Also according to spectral clustering theory, the affiliation of samples can be effectively discovered by analyzing its top eigenvectors of the graph induced from \mathbf{S} [95]. Based on spectral graph theory [12], we propose the novel concept of spectral feature selection [108, 113]. Spectral feature selection defines a unified framework for feature selection and utilize the spectrum of a given similarity matrix or its induced graph to evaluate feature relevance.

By designing different \mathbf{S} 's, the spectral feature selection framework can generate families of algorithms for both supervised and unsupervised feature selection. And when a small amount of labeled data is available with a large amount of unlabeled data, the spectral feature selection framework can be naturally extended to achieve semi-supervised feature selection [107, 98]. By applying the perturbation theory developed for symmetric linear system [15], its robustness can be theoretically analyzed, which provides insights on its behavior with random noise. We also prove that many existing successful feature selection algorithms, such as ReliefF, Laplacian Score, Fisher Score, HSIC, and Trace Ratio, are actually special cases of the proposed spectral feature selection framework.

Redundant features increase dimensionality unnecessarily [39], which worsens the learning performance. It has been empirically shown that removing redundant features can result in significant performance improvement [29, 16, 23, 100, 2, 18]. However, it turns out that most existing feature selection algorithms based on similarity preservation evaluate features individually, therefore are not capable to handle redundant features. The proposed spectral feature selection framework can effectively address the limitation via evaluating the utility of a set of features jointly. And this can be achieved by reformulating the original spectral feature selection problem as a multi-output regression problem [30], and feature selection is achieved by enforcing sparsity through applying $L_{2,1}$ -norm constraint on the solutions [63, 4]. We analyze its capability on redundancy removal and study the properties of its optimal solutions, which paves the way for an efficient path-following solver. By exploiting the necessary and sufficient conditions for the optimal solutions, we developed an very efficient solver for the proposed sparse multi-output regression problem, which can automatically adjust its parameters and generate a solution path for selecting a specific number of features. Empirical study in both supervised and unsupervised learning shows that the proposed formulation can effectively remove redundant features and result in a compact set of relevant features leading to superior learning performance.

One of the most challenging problem in many feature selection applications is the small sample problem [69], where the dimensionality of data is extremely high, while the sample size is very small. For instance, a typical cDNA microarray data [38] used in modern genetic analysis usually contains more than 30000 features (the oligonucleotide probes). But the sample size is usually less than 100. With so few samples, many features, which are irrelevant to the target concept, can easily gain their statistical relevance due to randomness [81]. With a data of this kind, most existing feature selection algorithms become unreliable by selecting many irrelevant features. For example in cancer study based on cDNA microarray, fold differences identified via statistical analysis offers limited or inaccurate selection of biological features [54, 81]. In real applications, the number of samples is usually impossible to be increased considerably, since the process of acquiring additional samples are costly. One way to address this problem is to include additional information sources to enhance our understanding of the data in hand. For instance, the recent developments in bioinformatics have made various knowledge sources available, including the KEEG pathway repository [37], the Gene Ontology database [8] and the NCI Gene-Cancer database [75], etc. Recent work has also revealed the existence of a class of small non-coding RNA species known as microRNAs, which are surprisingly informative for identifying cancerous tissues [54]. The availability of these various knowledge sources presents unprecedented opportunities to advance research solving previously unsolvable problems. However, to the best of the author's knowledge, currently most feature selection algorithms are designed to handle learning tasks with single data source, therefore can not benefit from any additional knowledge sources. To address this limitation, in this work we develop novel multi-source feature selection algorithms based on the proposed spectral feature selection framework. The developed algorithms can effectively integrate multiple knowledge sources with heterogeneous representations. They have been applied to various research domain including biomedical analysis and genetic analysis, and resulted in

interesting researching findings.

The key contributions of this work are highlighted as below:

- We proposed the concept of spectral feature selection, which forms a unified framework to enable the joint study of supervised and unsupervised feature selection. We study the effects of its components and show how to derive new algorithms with good performance by choosing proper components according to data characteristics.
- For data containing random noise, we conduct robustness analysis for the proposed spectral feature selection framework, and provide error upper bounds based on perturbation theory developed for the symmetric linear system [15].
- We show that many existing successful feature selection algorithms are actually special cases of the spectral feature selection framework. The joint study of these algorithm allow us to gain insights and achieve better understanding on them.
- We propose a multivariate formulation for spectral feature selection based on sparse multiple output regression and develop efficient techniques to solve the problem. Both theoretical and empirical study show that the formulation is able to remove redundant features, which overcomes a common drawback of the existing algorithms.
- Based on the spectral feature selection framework, we proposed multi-source feature selection. The technique can effectively integrate multiple knowledge sources with heterogeneous representations, which helps increase information in the evaluation process and improve the reliability of relevance estimation. Applying the technique in real applications results in interesting researching findings.

The rest of this paper is organized as below: we begin with the notations used in this work in Section 1.2. We then propose the general framework for spectral feature selection

in Section 2. We also study the effects of its components and show how to derive new algorithms with good performance in the section. In Section 2.4, We conduct robustness analysis for the proposed spectral feature selection framework, and provide error upper bounds for the framework, when data containing random noise. In Section 3, we show that many existing successful feature selection algorithms are actually special cases of the proposed spectral feature selection framework. We propose a multivariate formulation for spectral feature selection to handle redundant features in Section 4. We study how to incorporate multiple data and knowledge sources in spectral feature selection to achieve multi-source feature selection in Section 5, and we conclude in Section 6.

1.2 Notations

In this work, we use \mathbf{X} to denote a data set of n samples and m features: $\mathbf{X} = (\mathbf{x}_1^\top, \dots, \mathbf{x}_n^\top)^\top$, $\mathbf{x}_i \in \mathbb{R}^m$. We use F_1, F_2, \dots, F_m to denote the m features, and $\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_m$ are the corresponding feature vectors, $\mathbf{X} = (\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_m)$. For supervised learning, $\mathbf{y} = (y_1, y_2, \dots, y_n)$ are the class labels, $y_i \in \{1, 2, \dots, c\}$, and c is the number of classes. According to the distribution of data or class affiliation, a set of pairwise sample similarity, \mathbb{S} , and its corresponding similarity matrix \mathbf{S} , can be constructed to represent the relationships among samples. Given \mathbf{X} , we use $\mathbb{G}(V, E)$ to denote an undirected graph constructed from \mathbf{S} , where V is the vertex set, and E is the edge set. The i -th vertex v_i of \mathbb{G} corresponds to $\mathbf{x}_i \in \mathbf{X}$ and there is an edge between each vertex pair (v_i, v_j) . Given \mathbb{G} , its adjacency matrix \mathbf{A} is defined as $a_{ij} = s_{ij}$. Let \mathbf{d} denote the vector: $\mathbf{d} = \{d_1, d_2, \dots, d_n\}$, where $d_i = \sum_{k=1}^n a_{ik}$, the degree matrix \mathbf{D} of \mathbb{G} is defined by: $\mathbf{D}(i, j) = d_i$ if $i = j$, and 0 otherwise. Here d_i can be interpreted as an estimation of the density around \mathbf{x}_i , since the more data points that are close to \mathbf{x}_i , the larger the d_i . Given the adjacency matrix \mathbf{A} and the degree matrix \mathbf{D} of \mathbb{G} , the Laplacian matrix \mathbf{L} and the normalized Laplacian matrix \mathbb{L} are defined as:

$$\mathbf{L} = \mathbf{D} - \mathbf{A}; \quad \mathbb{L} = \mathbf{D}^{-\frac{1}{2}} \mathbf{L} \mathbf{D}^{-\frac{1}{2}} \quad (1.3)$$

It is easy to verify that \mathbf{D} and \mathbf{L} satisfy the following properties [12].

Proposition 1 *Given W, L and D of \mathbb{G} , we have:*

1. Let $\mathbf{1} = \{1, 1, \dots, 1\}^T$, $\mathbf{L} * \mathbf{1} = 0$.
2. $\forall \mathbf{f} \in \mathbb{R}^n$, $\mathbf{f}^T \mathbf{L} \mathbf{f} = \frac{1}{2} \sum_{v_i \sim v_j} a_{i,j} (f_i - f_j)^2$
3. $\forall \mathbf{f} \in \mathbb{R}^n, \forall t \in \mathbb{R}, (\mathbf{f} - t * \mathbf{1})^T \mathbf{L} (\mathbf{f} - t * \mathbf{1}) = \mathbf{f}^T \mathbf{L} \mathbf{f}$

In multiple output regression analysis [30], assuming $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_k) \in \mathbb{R}^{n \times k}$ denote the response matrix, and $\mathbf{W} = (\mathbf{w}_1, \dots, \mathbf{w}_k) \in \mathbb{R}^{m \times k}$ denote the weight matrix, the goal is to learn a weight matrix \mathbf{W} to minimize the following objective function:

$$\|\mathbf{Y} - \mathbf{X}\mathbf{W}\|_F^2 = \text{Trace} \left((\mathbf{Y} - \mathbf{X}\mathbf{W})(\mathbf{Y} - \mathbf{X}\mathbf{W})^\top \right),$$

where $\|\cdot\|_F$ denotes the Frobenius norm [25].

In this work, we use boldface characters in uppercase, such \mathbf{X} and \mathbf{S} , to denote matrices, use boldface characters in lowercase, such as \mathbf{x} and \mathbf{d} , to denote vectors, and use normal characters in lowercase to denote scalars. We use $\mathbf{1}$ to denote the vector with all its elements being 1 and \mathbf{I} to denote the identity matrix with all its diagonal elements being 1 and all other elements being 0. A summary of the notations can be found in Table 1.1.

Table 1.1: A summary of the notations used in the paper

n	the number of instances
m	the number of features
c	the number of classes
\mathbf{X}	the data matrix, $\mathbf{X} \in \mathbb{R}^{n \times m}$
\mathbf{y}	the class label vector, $\mathbf{y} \in \mathbb{R}^{n \times 1}$
\mathbf{Y}	the response matrix in regression, $\mathbf{Y} \in \mathbb{R}^{n \times k}$
\mathbf{W}	the weight matrix in regression, $\mathbf{W} \in \mathbb{R}^{m \times k}$
\mathbf{S}	the similarity matrix, $\mathbf{S} \in \mathbb{R}^{n \times n}$
\mathbf{A}	the affinity matrix, $\mathbf{S} = \mathbf{A} \in \mathbb{R}^{n \times n}$
\mathbf{D}	the degree matrix, $\mathbf{D} \in \mathbb{R}^{n \times n}$
\mathbf{L}	the Laplacian matrix, $\mathbf{L} \in \mathbb{R}^{n \times n}$
\mathbb{L}	the normalized Laplacian matrix, $\mathbb{L} \in \mathbb{R}^{n \times n}$
\mathbf{I}	the identity matrix
$\mathbf{1}$	the vector with all its elements to be 1
F_i	the i th feature
\mathbf{f}_i	the i th feature vector, $\mathbf{f}_i \in \mathbb{R}^n$
\mathbf{x}_i	the i th instance, $\mathbf{x}_i \in \mathbb{R}^m$
λ	the regularization parameter
t	the constraint parameter
\mathcal{H}^F	knowledge sources related to features
\mathcal{H}^S	knowledge sources related to samples

Chapter 2

FEATURE SELECTION VIA SPECTRAL ANALYSIS

Given a sample similarity matrix \mathbf{S} depicting the sample affiliations, a graph \mathbb{G} can be constructed to represent it. And the target concept, which defines the sample affiliations, is usually reflected by the structure of \mathbb{G} . For example, the samples of the same category usually form a cluster structure with dense inner connections. A feature is *consistent* to the target concept, if the feature assigns similar values to the samples from the same category defined by the target concept. Reflecting on the graph \mathbb{G} , it assigns similar values to the samples which are near to each other. Consistent features contain information about the target concept, therefore help categorize samples correctly. As shown in Fig. 2.1, the target concept specifies two categories indicated by the two ellipses $C1$ and $C2$. And different shapes and colors corresponds to the values of a feature on the samples. As we can see that feature F assigns similar values to the samples that are of the same category, while F' does not. Comparing with F' , by using F to cluster or classify samples, we have better chance to categorize samples correctly. This observation enables us to conclude that F is more relevant. Based on this intuition, we propose to measure features' relevance via measuring their consistency, which will be achieved by checking its consistency with the graph related to the target concept. Specifically, we check whether a feature assigns similar values to samples which are near to each other on the graph. It is easy to see that features selected according to this criterion must have a strong capability on preserving the sample similarity specified by \mathbf{S} , since they are consistent with the structure of graph \mathbb{G} .

Given a graph \mathbb{G} , we can derive a Laplacian matrix \mathbf{L} . According to spectral graph theory, the structural information of a graph can be obtained by studying its spectrum (all the eigenpairs of \mathbf{L}) [12]. For example it is well known that the leading eigenvectors of \mathbf{L}

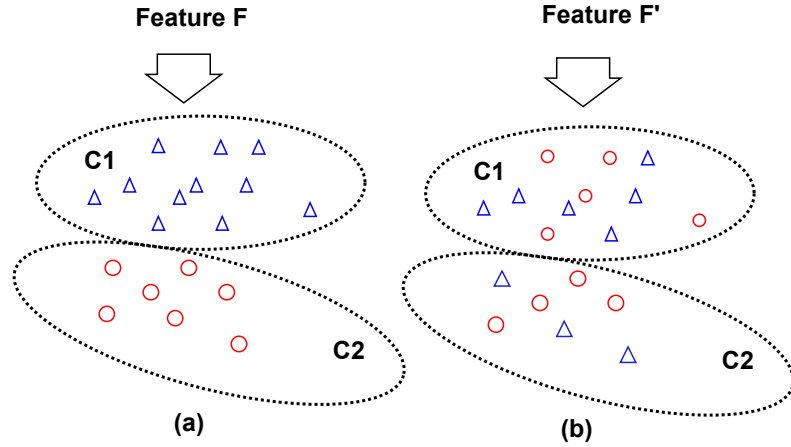


Figure 2.1: An exemplification of the basic idea behind spectral feature selection.

have a tendency to assign similar values to the samples that are near to each other on the graph. In this section we present the novel concept of spectral feature selection. In spectral feature selection we study how to measure feature relevance by using the spectrum of a graph, which is suitable for both supervised and unsupervised feature selection.

2.1 Ranking Features on Graph

We formalize the above idea using the concept of normalized cut for graph, derive two improved functions from the normalized cut formulation with the spectrum of the graph, and extend the three functions to their more general forms. These pave the way for constructing the unified framework for spectral feature selection.

Given \mathbb{G} , it can be shown that the Laplacian matrix of \mathbb{G} is a linear operator on vectors:

$$\langle \mathbf{f}, \mathbf{L}\mathbf{f} \rangle = \mathbf{f}^T \mathbf{L}\mathbf{f} = \frac{1}{2} \sum_{v_i \sim v_j} a_{ij} (f_i - f_j)^2, \quad \mathbf{f} = (f_1, f_2, \dots, f_m)^T \in \mathbb{R}^m. \quad (2.1)$$

The equation quantifies how much \mathbf{f} varies locally or how “smooth” it is over \mathbb{G} . More specifically, the smaller the value of $\langle \mathbf{f}, \mathbf{L}\mathbf{f} \rangle$, the smoother the vector \mathbf{f} on \mathbb{G} . A smooth vector \mathbf{f} assigns similar values to the samples that are close to each other on \mathbb{G} , thus it is consistent with the graph structure. This observation motivates us to apply \mathbf{L} on a feature

vector to measure its consistency with the graph structure. Given a feature vector \mathbf{f}_i and \mathbf{L} , two factors affect the value of $\langle \mathbf{f}_i, \mathbf{L}\mathbf{f}_i \rangle$: the norms of \mathbf{f}_i and \mathbf{L} . The two factors need to be removed, as they do not contain structure information of the data, but can cause the value of $\langle \mathbf{f}_i, \mathbf{L}\mathbf{f}_i \rangle$ to increase or decrease arbitrarily. The two factors can be removed via normalization. Based on the relationship between \mathbf{L} and \mathbb{L} , we have:

$$\langle \mathbf{f}_i, \mathbf{L}\mathbf{f}_i \rangle = \mathbf{f}_i^T \mathbf{L}\mathbf{f}_i = \mathbf{f}_i^T \mathbf{D}^{\frac{1}{2}} \mathbb{L} \mathbf{D}^{\frac{1}{2}} \mathbf{f}_i = (\mathbf{D}^{\frac{1}{2}} \mathbf{f}_i)^T \mathbb{L} (\mathbf{D}^{\frac{1}{2}} \mathbf{f}_i).$$

Let $\tilde{\mathbf{f}}_i = (\mathbf{D}^{\frac{1}{2}} \mathbf{f}_i)$ denote the weighted feature vector of F_i , and $\hat{\mathbf{f}}_i = \frac{\tilde{\mathbf{f}}_i}{\|\tilde{\mathbf{f}}_i\|}$ the normalized weighted feature vector. The score of F_i can be evaluated by the following function:

$$\varphi_1(F_i) = \hat{\mathbf{f}}_i^T \mathbb{L} \hat{\mathbf{f}}_i \quad (2.2)$$

Theorem 1 $\varphi_1(F_i)$ measures the value of the normalized cut [78] by using \mathbf{f}_i as the soft cluster indicator to partition the graph \mathbb{G} .

Proof: The theorem holds as:

$$\varphi_1(F_i) = \hat{\mathbf{f}}_i^T \mathbb{L} \hat{\mathbf{f}}_i = \frac{\mathbf{f}_i^T \mathbf{L}\mathbf{f}_i}{\mathbf{f}_i^T \mathbf{D}\mathbf{f}_i}$$

■

A soft cluster indicator is a cluster indicator that containing real values instead of categories. According to spectral clustering theorem, a good cluster indicator will result in a small normalized cut value by assigning similar values to samples that are near to each other on the graph. Also both theoretical and empirical results showed that the normalization step of normalized cut makes the method to be more robust to outliers [79]. Given the normalized Laplacian matrix \mathbb{L} , we can calculate its spectral decomposition (λ_i, ξ_i) , where λ_i is the eigenvalue and ξ_i is the eigenvector ($1 \leq i \leq n$). Assuming $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$, according to Proposition 1, we have: $\lambda_1 = 0$ and $\xi_1 = \frac{\mathbf{D}^{\frac{1}{2}} \mathbf{1}}{\|\mathbf{D}^{\frac{1}{2}} \mathbf{1}\|}$. (λ_1, ξ_1) is usually called

the trivial eigenpair of the graph. Also we can show that all the eigenvalues of \mathbb{L} are contained in $[0, 2]$. Given the spectral decomposition of \mathbb{L} , we can rewrite Eq. (2.2) using the eigensystem of \mathbb{L} to achieve a better understanding of the Equation.

Theorem 2 *Let (λ_j, ξ_j) , $1 \leq j \leq n$ be the eigensystem of \mathbb{L} , and $\alpha_j = \cos \theta_j$ where θ_j is the angle between $\widehat{\mathbf{f}}_i$ and ξ_j . Eq. (2.2) can be rewritten as:*

$$\varphi_1(F_i) = \sum_{j=1}^n \alpha_j^2 \lambda_j, \quad \text{where} \quad \sum_{j=1}^n \alpha_j^2 = 1 \quad (2.3)$$

Proof: Let $\Sigma = \text{DIAG}(\lambda_1, \lambda_2, \dots, \lambda_n)$ and $U = (\xi_1, \xi_2, \dots, \xi_n)$. As $\|\widehat{\mathbf{f}}_i\| = \|\xi_j\| = 1$, we have $\widehat{\mathbf{f}}_i^T \xi_j = \cos \theta_j$. We can rewrite $\widehat{\mathbf{f}}_i^T \mathbb{L} \widehat{\mathbf{f}}_i$ as:

$$\widehat{\mathbf{f}}_i^T \mathbb{L} \widehat{\mathbf{f}}_i = \widehat{\mathbf{f}}_i^T U \Sigma U^T \widehat{\mathbf{f}}_i = (\alpha_1, \dots, \alpha_n) \Sigma (\alpha_1, \dots, \alpha_n)^T = \sum_{i=1}^n \alpha_i^2 \lambda_i$$

Also $\sum_{j=1}^n \alpha_j^2 = 1$, as $UU^T = I$ and $\|\widehat{\mathbf{f}}_i\| = 1$. ■

Theorem 2 says that using Eq. (2.2), the score of F_i is calculated by combining the eigenvalues of \mathbb{L} , and $\cos \theta_1, \dots, \cos \theta_n$ are the combination coefficients, which measure the similarity between the feature vector and the eigenvectors. To study the properties of the eigen-system of \mathbb{L} , we construct an example, which contains samples forming three clusters. Figure 2.2 plots the contours of the eigenvectors $\xi_1, \xi_2, \xi_3, \xi_4, \xi_5$, and ξ_{20} of the \mathbb{L} constructed based on the example. The contours show how the eigenvectors assign values to the sample, and the darker the color, smaller the values the samples obtained. The figure shows that the first eigenvector capture the density information of the data. Since there are three clusters, the second and the third eigenvectors capture the cluster structure of the data. The fourth and the fifth eigenvectors capture the subcluster structure of the data. And the twentieth eigenvectors capture the subtle structure of the data, which may corresponding

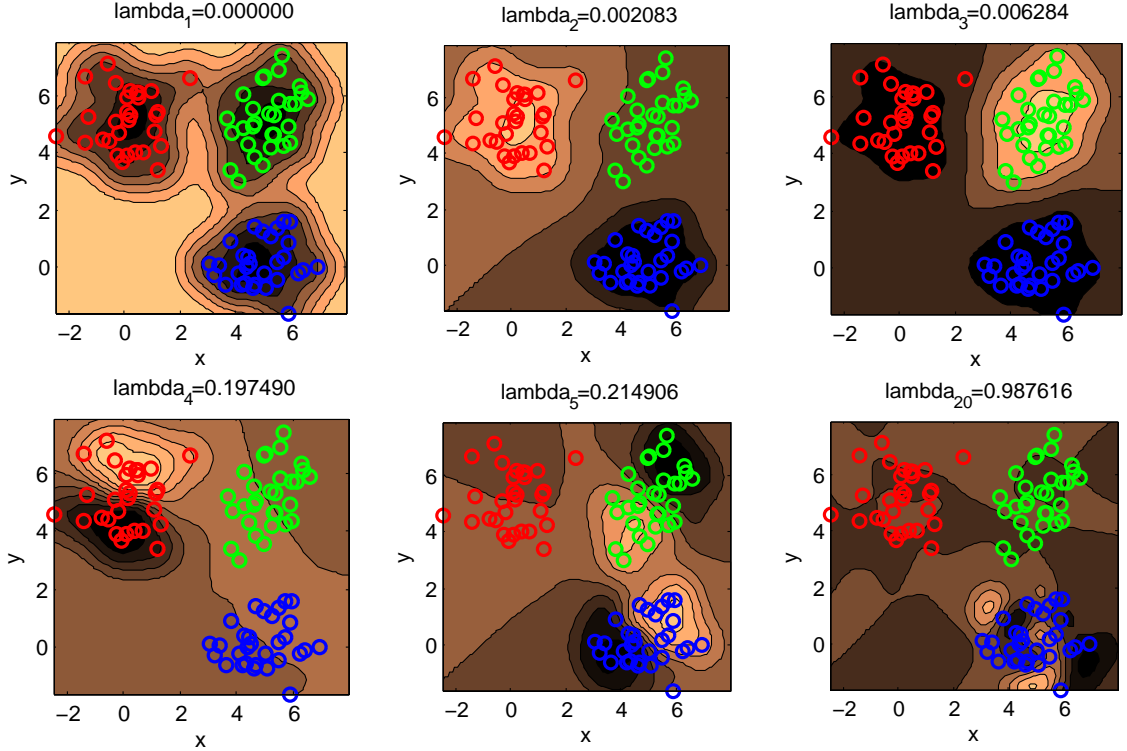


Figure 2.2: Contours of the eigenvectors $\xi_1, \xi_2, \xi_3, \xi_4, \xi_5$, and ξ_{20} .

to the ones formed by noise. We also noticed that λ_1, λ_2 , and λ_3 are significantly smaller than the remaining eigenvalues. According to spectral graph theories, the eigenvalues of \mathbb{L} measure the smoothness of their corresponding eigenvectors¹. Since $\lambda_1 = 0$, Equation (2.3) can be rewritten as $\hat{\mathbf{f}}_i^T \mathbb{L} \hat{\mathbf{f}}_i = \sum_{j=2}^n \alpha_j^2 \lambda_j$, meaning that the value obtained from Eq. (2.2) evaluate the smoothness of $\hat{\mathbf{f}}_i$ by measuring the similarities between $\hat{\mathbf{f}}_i$ and those nontrivial eigenvectors of \mathbb{L} . Since $\sum_{j=1}^n \alpha_j^2 = 1$ and $\alpha_1 \geq 0$, we have $\sum_{j=2}^n \alpha_j^2 \leq 1$, and the bigger the α_1^2 , the smaller the $\sum_{j=2}^n \alpha_j^2$. The value of $\varphi_1(F_i)$ can be small, if $\hat{\mathbf{f}}_i$ is very similar with ξ_1 . However, in this case, a small $\varphi_1(F_i)$ value does not indicate better separability, since the trivial eigenvector ξ_1 does not carry any distribution information except the density around samples. To handle this case, we propose to use $\sum_{j=2}^n \alpha_j^2$ to normalize $\varphi_1(F_i)$, which gives

¹The smaller the value the more smooth the eigenvector.

us the following ranking function:

$$\varphi_2(F_i) = \frac{\sum_{j=2}^n \alpha_j^2 \lambda_j}{\sum_{j=2}^n \alpha_j^2} = \frac{\widehat{\mathbf{f}}_i^T \mathbb{L} \widehat{\mathbf{f}}_i}{1 - \left(\widehat{\mathbf{f}}_i^T \boldsymbol{\xi}_1\right)^2} \quad (2.4)$$

A small $\varphi_2(F_i)$ indicates that $\widehat{\mathbf{f}}_i$ aligns closely to those nontrivial eigenvectors with small eigenvalues, hence is smooth on the graph. According to spectral clustering theory, the leading k eigenvectors of \mathbb{L} form the optimal soft cluster indicators that separate \mathbb{G} into k parts. While the remaining eigenvectors corresponding to the subtle structures formed by noise. Therefore, if k is known, for instance, we know that the data contain samples from k different categories, which should form k dense clusters, we can also use the following function to estimate the relevance of the features:

$$\varphi_3(F_i) = \sum_{j=2}^k (2 - \lambda_j) \alpha_j^2 \quad (2.5)$$

By its definition, φ_3 assigns bigger scores to features which are more relevant, as achieving a big score entails that a feature aligns closely to nontrivial eigenvectors $\boldsymbol{\xi}_2, \dots, \boldsymbol{\xi}_k$, with $\boldsymbol{\xi}_2$ having the highest priority. By focusing on the leading eigenvectors, φ_3 can effectively reduce noise. Similar mechanism is widely used in Principle Component Analysis (PCA) [6] and spectral dimension reduction techniques [73] for noise reduction.

2.2 An Extension for Feature Ranking Functions

Laplacian matrix is also used by graph based learning models for designing regularization functions to penalize predictors that vary abruptly among adjacent vertices on graph. In [83], the authors related the eigenvectors of \mathbb{L} to a Fourier basis and extend the usage of \mathbb{L} to $\gamma(\mathbb{L})$, where $\gamma(\cdot)$ is a spectral matrix function defined as: $\gamma(\mathbb{L}) = \sum_{j=1}^n \gamma(\lambda_j) \boldsymbol{\xi}_j \boldsymbol{\xi}_j^T$. In the formulation, $\gamma(\lambda_j)$ is an increasing function that penalizes high frequency components². In

²Here λ_j is used to estimate frequency, since it measures how smooth the corresponding basis $\boldsymbol{\xi}_j$ is on the graph. The more smoother the basis, the lower the frequency.

[104], it is also pointed out that noise is able to flatten the spectrum of the Laplacian matrix of the data and causes performance degeneration. In practise, it is beneficial to use high order spectral matrix functions to penalize the high frequency components to alleviate the effect. In the this spirit, we extend our feature ranking functions to the following:

$$\widehat{\varphi}_1(F_i) = \widehat{\mathbf{f}}_i^T \gamma(\mathbb{L}) \widehat{\mathbf{f}}_i = \sum_{j=1}^n \alpha_j^2 \gamma(\lambda_j) \quad (2.6)$$

$$\widehat{\varphi}_2(F_i) = \frac{\sum_{j=2}^n \alpha_j^2 \gamma(\lambda_j)}{\sum_{j=2}^n \alpha_j^2} = \frac{\widehat{\mathbf{f}}_i^T \gamma(\mathbb{L}) \widehat{\mathbf{f}}_i}{1 - \left(\widehat{\mathbf{f}}_i^T \xi_1\right)^2} \quad (2.7)$$

$$\widehat{\varphi}_3(F_i) = \sum_{j=2}^k (\gamma(2) - \gamma(\lambda_j)) \alpha_j^2 \quad (2.8)$$

Calculating the spectral decomposition of \mathbb{L} can be expensive for data with a large number of samples. However, since $\gamma(\cdot)$ is usually a rational function, $\gamma(\mathbb{L})$ can be calculated efficiently by regarding \mathbb{L} as a variable and apply $\gamma(\cdot)$ on it. For example, assume $\gamma(\lambda) = \lambda^2$, then $\gamma(\mathbb{L}) = \mathbb{L}^2$. For $\widehat{\varphi}_3(\cdot)$, the k leading eigenpairs of \mathbb{L} can be obtained efficiently by using fast eigen-solvers such as the Implicitly Restarted Arnoldi method [68].

2.3 SPEC, A Unified Framework for Spectral Feature Selection

The proposed framework is built on spectral graph theory. In the framework, the relevance of a feature is determined by its consistency with the structure of the graph induced from \mathbb{S} . The three feature ranking functions, $\widehat{\varphi}_1(\cdot)$, $\widehat{\varphi}_2(\cdot)$, and $\widehat{\varphi}_3(\cdot)$ lay the foundation of the framework and enable us to derive families of supervised and unsupervised feature selection in a unified manner. We realize the unified framework in Algorithm 1. It selects features in three steps: (1) building similarity set \mathbb{S} and constructing its graph representation (Line 1-3); (2) evaluating features using the eigensystem of the graph (Line 4-6); and (3) rank-

ing features in descending order in terms of feature relevance³ (Line 7-8). We name the framework SPEC, stemming from the SPECTrum decomposition of \mathbb{L} .

The time complexity of SPEC largely depends on the cost of building the similarity matrix and the calculation of $\gamma(\cdot)$. If we use the RBF function to build the similarity matrix and $\gamma(\cdot)$ is in the form of \mathbb{L}^r , the time complexity of SPEC can be obtained as follow. First, we need $O(mn^2)$ operations to build \mathbf{S} , \mathbf{W} , \mathbf{D} , \mathbf{L} and \mathbb{L} . And we need $O(rn^3)$ operations to calculate $\gamma(\mathbb{L})$. Next, we need $O(n^2)$ operations to calculate $SF_{SPEC}(i)$ for each feature: transforming \mathbf{f}_i to $\hat{\mathbf{f}}_i$ requires $O(n)$ operations; calculating $\hat{\varphi}_1$, $\hat{\varphi}_2$ and $\hat{\varphi}_3$ need $O(n^2)$ operations⁴. Therefore, we need $O(mn^2)$ operations to calculate scores for m features. Last, we needs $O(m \log m)$ operations to rank the features. Hence, the overall time complexity of SPEC is $O((rn + m)n^2)$, or $O(mn^2)$ if $\gamma(\cdot)$ is not used.

Algorithm 1: SPEC

Input: \mathbf{X} , $\gamma(\cdot)$, k , $\hat{\varphi} \in \{\hat{\varphi}_1, \hat{\varphi}_2, \hat{\varphi}_3\}$

Output: the ranked feature list

- 1 construct \mathbf{S} , the similarity matrix from \mathbf{X} (and \mathbf{y});
- 2 construct graph \mathbb{G} from \mathbf{S} ;
- 3 build \mathbf{W} , \mathbf{D} and \mathbf{L} from \mathbb{G} ;
- 4 **for** each feature vector \mathbf{f}_i **do**
- 5 $\hat{\mathbf{f}}_i \leftarrow \frac{\mathbf{D}^{\frac{1}{2}} \mathbf{f}_i}{\|\mathbf{D}^{\frac{1}{2}} \mathbf{f}_i\|}$; $SF_{SPEC}(i) \leftarrow \hat{\varphi}(F_i)$;
- 6 **end**
- 7 ranking features according to SF_{SPEC} in ascending order for $\hat{\varphi}_1$ and $\hat{\varphi}_2$, or descending order for $\hat{\varphi}_3$;
- 8 **return** ranked list;

³Features selection is accomplished by choosing the desired number of features from returned list.

⁴For $\hat{\varphi}_3$, using Arnoldi method to calculate a few eigenpairs of a large sparse matrix needs roughly $O(n^2)$ operations, and calculating $\hat{\varphi}_3$ itself needs $O(k)$ operations.

SPEC is a general framework for feature selection. The framework can be used to systematically derive novel algorithm by using different similarity matrix \mathbf{S} obtained from different sample similarity measurements, different spectral matrix functions $\gamma(\cdot)$, and different rank functions $\widehat{\varphi}(\cdot)$. For example, given the options of \mathbf{S} and $\gamma(\cdot)$ in Table 2.1, we can generate families of new supervised and unsupervised feature selection algorithms. Below we analyze the effect of components in the SPEC framework; conduct experiments to evaluate how effective these algorithms are and how they fare in comparison with the baseline algorithms; and provide a guideline for users to choose the proper components according to the reality of different applications.

Table 2.1: The components for SPEC tried in this work. \mathbf{S}_u and \mathbf{S}_s stand for the similarity matrices used in unsupervised and supervised feature selection respectively.

$\widehat{\varphi}(\cdot)$	$\widehat{\varphi}_1(\cdot)$, $\widehat{\varphi}_2(\cdot)$ and $\widehat{\varphi}_3(\cdot)$
$\gamma(\cdot)$	$\gamma(r) = r$, $\gamma(r) = r^3$
\mathbf{S}_u	RBF kernel function, Dijkstra function [94]
\mathbf{S}_s	Similarity matrix defined in Equations 3.13

2.4 Robustness Analysis for SPEC

Being robust is an important character for feature selection algorithms [71, 101]. A feature selection algorithm is not robust if a small perturbation of the original data can cause a great change on its output. Although the perturbations can be various (eg. caused by various noise), the underlying target concept keep unchanged. Therefore a good feature selection algorithms should be robust to the potential perturbations. Below, we provide a robustness analysis for feature ranking functions $\varphi_1(\cdot)$, $\varphi_2(\cdot)$ and $\varphi_3(\cdot)$. The analysis is base on the perturbation theory developed for symmetric linear system [15], and can be extended to $\widehat{\varphi}_1(\cdot)$, $\widehat{\varphi}_2(\cdot)$ and $\widehat{\varphi}_3(\cdot)$ easily. We first present two theorems, which serve as the basis for the following analysis.

Theorem 3 (Weyl) *Let \mathbf{A} and \mathbf{E} be n -by- n symmetric matrices. Let $\lambda_1 \leq \dots \leq \lambda_n$ be the eigenvalues of \mathbf{A} and $\tilde{\lambda}_1 \leq \dots \leq \tilde{\lambda}_n$ be the eigenvalues of $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{E}$, then $\|\lambda_j - \tilde{\lambda}_j\| \leq \|\mathbf{E}\|_2$*

Theorem 4 *Let \mathbf{A} and \mathbf{E} be n -by- n symmetric matrices. Let $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top = \mathbf{Q}\text{diag}(\lambda_j)\mathbf{Q}^\top$ be an eigen decomposition of the \mathbf{A} . Let $\mathbf{A} + \mathbf{E} = \hat{\mathbf{A}} = \tilde{\mathbf{Q}}\tilde{\mathbf{\Lambda}}\tilde{\mathbf{Q}}^\top$ be the perturbed eigen decomposition. Write $\mathbf{Q} = [\xi_1, \dots, \xi_n]$ and $\hat{\mathbf{Q}} = [\tilde{\xi}_1, \dots, \tilde{\xi}_n]$, where ξ_j and $\tilde{\xi}_j$ are the unperturbed and perturbed unit eigenvectors, respectively. Let θ_j denote the acute angle between ξ_j and $\tilde{\xi}_j$. Provided that $\text{GAP}(j, \mathbf{A} + \mathbf{E}) > 0$ we have the following inequality:*

$$\frac{1}{2} \sin 2\theta_j \leq \frac{\|\mathbf{E}\|_2}{\text{GAP}(j, \mathbf{A} + \mathbf{E})}, \quad (2.9)$$

Note that when $\theta_j \ll 1$, then $\frac{1}{2} \sin 2\theta_j \approx \sin \theta_j \approx \theta_j$

Proof for the two theorems can be found in section 5.2 of [15]. In Theorem 4, $\text{GAP}(j, \mathbf{A} + \mathbf{E})$ denotes the eigen-gap of λ_j , where λ_j is the j th eigenvalue of $\mathbf{A} + \mathbf{E}$. Formally we can define $\text{GAP}(j, \mathbf{A} + \mathbf{E})$ as: $\text{GAP}(j, \mathbf{A} + \mathbf{E}) = \min_{i \neq j} |\lambda_i - \lambda_j|$. According to Theorem 3 and 4, the robustness of the eigenvalues is determined by the scale of the perturbation matrix E , which is measured by its norm. And the robustness of the eigenvectors is determined by the scale of the perturbation matrix E as well as the eigen-gap of the corresponding eigenvalue. Based on the two theorems, we provide an error upper bound for feature ranking functions $\varphi_1(\cdot)$, $\varphi_2(\cdot)$ and $\varphi_3(\cdot)$ when the original data is perturbed by noise. In general, noise causes two types of perturbation that will effect the outputs of ranking functions. They are (1) the perturbation of the Laplacian matrix \mathbb{L} , which is denoted as \mathbb{L}_ε ; and (2) the perturbation of the feature vector \mathbf{f} , which is denoted as \mathbf{f}_ε . Without loss of generality, for the original data and its perturbation, assume $\varepsilon_{\mathbb{L}} \geq 0$, we have the following specifications:

$$\tilde{\mathbb{L}} = \mathbb{L} + \mathbb{L}_\varepsilon, \|\tilde{\mathbb{L}}\|_2 = \|\mathbb{L}\|_2 = 1, \|\mathbb{L}_\varepsilon\|_2 \leq \varepsilon_{\mathbb{L}}. \quad (2.10)$$

$$\tilde{\mathbf{f}} = \mathbf{f} + \mathbf{f}_\varepsilon, \|\tilde{\mathbf{f}}\|_2 = \|\mathbf{f}\|_2 = 1, \|\mathbf{f}_\varepsilon\|_2 \leq \varepsilon_{\mathbf{f}}. \quad (2.11)$$

In above equations, \mathbb{L} is the original Laplacian matrix, $\tilde{\mathbb{L}}$ is the perturbed Laplacian matrix and \mathbb{L}_ε is the corresponding perturbation. Similar relationship holds for \mathbf{f} , \mathbf{f}_ε and $\tilde{\mathbf{f}}$, where \mathbf{f} is a feature vector. With theorem 5, we show that we can bound the perturbation errors of the three feature ranking functions $\varphi_1(\cdot)$, $\varphi_2(\cdot)$ and $\varphi_3(\cdot)$ by $\varepsilon_{\mathbb{L}}$, $\varepsilon_{\mathbf{f}}$ and the eigen-gaps of the eigenvalues of $\tilde{\mathbb{L}}$.

Theorem 5 Assume (λ_j, ξ_j) be the eigensystem of \mathbb{L} , and $\alpha_j = \cos \theta_j$, where θ_j is the angle between \mathbf{f} and ξ_j , also let $(\tilde{\lambda}_j, \tilde{\xi}_j)$ be the eigensystem of $\tilde{\mathbb{L}}$, and $\tilde{\alpha}_j = \cos \tilde{\theta}_j$ where $\tilde{\theta}_j$ is the angle between $\tilde{\mathbf{f}}$ and $\tilde{\xi}_j$, then we have:

$$\left(\sum_{j=1}^q \tilde{\alpha}_j^2 \tilde{\lambda}_j - \sum_{j=1}^q \alpha_j^2 \lambda_j \right) \leq \left(q \|\mathbb{L}_\varepsilon\|_2 + \sum_{j=1}^q \lambda_j \sin \left(\frac{1}{2} \arcsin 2\varepsilon_j, \theta \right) + \hat{\varepsilon}_f \sum_{j=1}^q \lambda_j \right), \quad (2.12)$$

where $\varepsilon_j, \theta = \frac{\|\mathbb{L}_\varepsilon\|_2}{\text{GAP}(j, \mathbb{L} + \mathbb{L}_\varepsilon)}$ and $\hat{\varepsilon}_f = 2\varepsilon_f + \varepsilon_f^2$. Note, when $\arcsin 2\varepsilon_j, \theta \ll 1$,

$$\begin{aligned} q \|\mathbb{L}_\varepsilon\|_2 + \sum_{j=1}^q \lambda_j \sin \left(\frac{1}{2} \arcsin 2\varepsilon_j, \theta \right) + \hat{\varepsilon}_f \sum_{j=1}^q \lambda_j \\ \approx \left(q \|\mathbb{L}_\varepsilon\|_2 + \sum_{j=1}^q \lambda_j \varepsilon_j, \theta + \hat{\varepsilon}_f \sum_{j=1}^q \lambda_j \right). \end{aligned} \quad (2.13)$$

Proof: We first provide an upper bound for $\tilde{\alpha}_j^2 \tilde{\lambda}_j$ using $\varepsilon_{\mathbb{L}}$, $\varepsilon_{\mathbf{f}}$ and the eigen gap of $\tilde{\mathbb{L}}$. For convenience, in the first part of the proof, we drop off the subscript j from $\tilde{\alpha}_j^2 \tilde{\lambda}_j$, if it does not cause confusion. Let $\tilde{\xi} = \xi + \xi_\varepsilon, \tilde{\lambda} = \lambda + \lambda_\varepsilon$. we have:

$$\begin{aligned} \tilde{\alpha}^2 \tilde{\lambda} &= \tilde{\lambda} \cos^2(\tilde{\theta}) = (\lambda + \lambda_\varepsilon) \left((\mathbf{f} + \mathbf{f}_\varepsilon)^T (\xi + \xi_\varepsilon) \right)^2 \\ &= \lambda \left((\mathbf{f} + \mathbf{f}_\varepsilon)^T (\xi + \xi_\varepsilon) \right)^2 + \lambda_\varepsilon \left((\mathbf{f} + \mathbf{f}_\varepsilon)^T (\xi + \xi_\varepsilon) \right)^2 \\ &\leq \lambda \left((\mathbf{f} + \mathbf{f}_\varepsilon)^T (\xi + \xi_\varepsilon) \right)^2 + \lambda_\varepsilon (\|\mathbf{f} + \mathbf{f}_\varepsilon\| \|\xi + \xi_\varepsilon\|)^2 \\ &= \lambda \left((\mathbf{f}^T (\xi + \xi_\varepsilon))^2 + (\mathbf{f}_\varepsilon^T (\xi + \xi_\varepsilon))^2 + 2\mathbf{f}^T (\xi + \xi_\varepsilon) \mathbf{f}_\varepsilon^T (\xi + \xi_\varepsilon) \right) + \lambda_\varepsilon \\ &\leq \lambda \left((\mathbf{f}^T (\xi + \xi_\varepsilon))^2 + (\|\mathbf{f}_\varepsilon\|_2 \|\xi + \xi_\varepsilon\|_2)^2 + 2\|\mathbf{f}\|_2 \|\xi + \xi_\varepsilon\|_2 \|\mathbf{f}_\varepsilon\|_2 \right) + \lambda_\varepsilon \end{aligned}$$

Since $\|\tilde{\xi}\|_2 = \|\xi\|_2 = 1$ and $\|\tilde{\mathbf{f}}\|_2 = \|\mathbf{f}\|_2 = 1$, we have:

$$\tilde{\alpha}^2 \tilde{\lambda} \leq \lambda \left((\mathbf{f}^T (\xi + \xi_\varepsilon))^2 + \varepsilon_f^2 + 2\varepsilon_f \right) + \lambda_\varepsilon$$

In the above derivations, we applied the Cauchy-Schwarz inequality: $\mathbf{x}^T \mathbf{y} \leq \|\mathbf{x}\| \|\mathbf{y}\|$. We notice that $\mathbf{f}^T (\xi + \xi_\varepsilon) \leq \cos(\theta - \theta_\varepsilon)$, where θ_ε is the angle between ξ and $\tilde{\xi}$, we have:

$$\begin{aligned} (\mathbf{f}^T (\xi + \xi_\varepsilon))^2 &\leq \cos^2(\theta - \theta_\varepsilon) - \cos^2(\theta) + \cos^2(\theta) \\ &= \sin(2\theta - \theta_\varepsilon) \sin(\theta_\varepsilon) + \cos^2(\theta) \\ &\leq |\sin(\theta_\varepsilon)| + \cos^2(\theta) \end{aligned}$$

Based on the two sets of results we just obtained, we have the following inequality:

$$\tilde{\alpha}^2 \tilde{\lambda} \leq \lambda_\varepsilon + \lambda (\varepsilon_f^2 + 2\varepsilon_f) + \lambda |\sin(\theta_\varepsilon)| + \lambda \cos^2(\theta),$$

and

$$\tilde{\alpha}^2 \tilde{\lambda} - \alpha^2 \lambda \leq \lambda_\varepsilon + \lambda (\varepsilon_f^2 + 2\varepsilon_f) + \lambda |\sin(\theta_\varepsilon)|.$$

According to Theorem 4, we know: $\lambda_\varepsilon \leq \|\mathbb{L}_\varepsilon\|$ and $|\sin(\theta_\varepsilon)| \leq \sin\left(\frac{1}{2} \arcsin 2\varepsilon_j, \theta\right)$, where $\varepsilon_j, \theta = \frac{\|\mathbb{L}_\varepsilon\|_2}{\text{GAP}(j, \mathbb{L} + \mathbb{L}_\varepsilon)}$. This allows us to obtain the following result:

$$\left(\sum_{j=1}^q \tilde{\alpha}_j^2 \tilde{\lambda}_j - \sum_{j=1}^q \alpha_j^2 \lambda_j \right) \leq \left(q \|\mathbb{L}_\varepsilon\|_2 + \sum_{j=1}^q \lambda_j \sin\left(\frac{1}{2} \arcsin 2\varepsilon_j, \theta\right) + \hat{\varepsilon}_f \sum_{j=1}^q \lambda_j \right).$$

Also, by noticing that when $\theta \ll 1$, $\frac{1}{2} \sin 2\theta \approx \sin \theta \approx \theta$, we can obtain Eq. (2.13). ■

Comparing with $\varphi_i(\cdot)$, $\hat{\varphi}_i(\cdot)$ applies $\gamma(\cdot)$ to rescale the eigenvalues of \mathbb{L} before calculating the feature scores. Based on Theorem 5, we provide a robustness analysis for $\hat{\varphi}_i(\cdot)$. In the analysis we assume $\gamma(\cdot)$ is a rational function and has the form $\gamma(\lambda) = \lambda^r$.

Theorem 6 Let (λ_j, ξ_j) be the eigensystem of \mathbb{L} , and $\alpha_j = \cos \theta_j$ where θ_j is the angle between \mathbf{f} and ξ_j , let $(\tilde{\lambda}_j, \tilde{\xi}_j)$ be the eigensystem of $\tilde{\mathbb{L}}$, and $\tilde{\alpha}_j = \cos \tilde{\theta}_j$ where $\tilde{\theta}_j$ is the angle between $\tilde{\mathbf{f}}$ and $\tilde{\xi}_j$. Also let $\tilde{\lambda}_j = \lambda_j + \lambda_{\varepsilon,j}$, and assume $\rho \lambda_j \geq \lambda_{\varepsilon,j}$ for all $1 \leq i \leq n$, with $\gamma(\lambda) = \lambda^r$ we have the following inequality holds:

$$\begin{aligned} & \left(\sum_{j=1}^q \tilde{\alpha}_j^2 \tilde{\lambda}_j^r - \sum_{j=1}^q \alpha_j^2 \lambda_j^r \right) \\ & \leq \sum_{j=1}^q \left(\|\mathbb{L}_\varepsilon\|_2 \frac{(\rho+1)^r - 1}{\rho} \lambda_j^{r-1} + \sin \left(\frac{1}{2} \arcsin 2\varepsilon_j, \theta \right) \lambda_j^r + \hat{\varepsilon}_f \lambda_j^r \right) \end{aligned} \quad (2.14)$$

where $\varepsilon_j, \theta = \frac{\|\mathbb{L}_\varepsilon\|_2}{\text{GAP}(j, \mathbb{L} + \mathbb{L}_\varepsilon)}$ and $\hat{\varepsilon}_f = 2\varepsilon_f + \varepsilon_f^2$. Note, when $\arcsin \varepsilon_j, \theta \ll 1$, we have:

$$\begin{aligned} & \sum_{j=1}^q \left(\|\mathbb{L}_\varepsilon\|_2 \frac{(\rho+1)^r - 1}{\rho} \lambda_j^{r-1} + \sin \left(\frac{1}{2} \arcsin 2\varepsilon_j, \theta \right) \lambda_j^r + \hat{\varepsilon}_f \lambda_j^r \right) \\ & \leq \sum_{j=1}^q \left(\|\mathbb{L}_\varepsilon\|_2 \frac{(\rho+1)^r - 1}{\rho} \lambda_j^{r-1} + \varepsilon_j, \theta \lambda_j^r + \hat{\varepsilon}_f \lambda_j^r \right). \end{aligned} \quad (2.15)$$

Proof: This is true, since we have that the following inequality holds:

$$\begin{aligned} \tilde{\alpha}^2 \tilde{\lambda}^r &= (\lambda + \lambda_\varepsilon)^r \cos^2(\tilde{\theta}) - \lambda^r \cos^2(\tilde{\theta}) + \lambda^r \cos^2(\tilde{\theta}) \\ &= ((\lambda + \lambda_\varepsilon)^r - \lambda^r) \cos^2(\tilde{\theta}) + \lambda^r \cos^2(\tilde{\theta}) \\ &= \lambda_\varepsilon (\lambda^{r-1} + \lambda_\varepsilon \lambda^{r-2} + \dots + \lambda_\varepsilon^{r-2} \lambda + \lambda_\varepsilon^{r-1}) \cos^2(\tilde{\theta}) + \lambda^r \cos^2(\tilde{\theta}) \\ &\leq \lambda_\varepsilon \lambda^{r-1} (1 + (1+\rho) + \dots + (1+\rho)^{r-2} + (1+\rho)^{r-1}) \cos^2(\tilde{\theta}) + \lambda^r \cos^2(\tilde{\theta}) \\ &= \lambda_\varepsilon \lambda^{r-1} \frac{(1+\rho)^r - 1}{\rho} \cos^2(\tilde{\theta}) + \lambda^r \cos^2(\tilde{\theta}) \\ &\leq \lambda_\varepsilon \lambda^{r-1} \frac{(1+\rho)^r - 1}{\rho} + \lambda^r \cos^2(\tilde{\theta}) \end{aligned}$$

■

When original data is perturbed by noise, feature scores will change accordingly. The two theorems tell that, before and after the perturbation, the difference of the feature scores

is bounded by $\varepsilon_{\mathbb{L}}$, $\varepsilon_{\mathbf{f}}$ and the eigen gap of $\tilde{\mathbb{L}}$. Among these factors, $\varepsilon_{\mathbb{L}}$ corresponds to the matrix perturbation, $\hat{\varepsilon}_{\mathbf{f}}$ corresponds to the feature vector perturbation and the eigen-gap of $\tilde{\mathbb{L}}$ corresponds to matrix stability [15], which measures how robust the matrix is to noise. We have the following points for the robustness of the feature ranking functions:

a.) Discarding the tail eigenpairs in feature evaluation helps increase robustness. It is known that for a graph with well separable cluster structures, its tail eigenvalues are usually packed in a small range, thus have small eigen-gaps [60]. Eq. (2.14) suggests including eigenpairs with small eigen-gaps causes increasing sensitivity on noise. Also it is known that for a graph, its tail eigenvectors usually correspond to the subtle structures formed due to noise [95]. Therefore removing them helps improve robustness.

b.) Among the three feature ranking functions, $\varphi_3(\cdot)$ is more robust than $\varphi_1(\cdot)$ and $\varphi_2(\cdot)$, since it discards the less robustness tail eigenpairs in evaluation. $\varphi_1(\cdot)$ and $\varphi_2(\cdot)$ are equally robust, since λ_1 and ξ_1 are constants. Note, although $\varphi_3(\cdot)$ is more robust in theory, to performs well, it needs a proper threshold to determine which tail eigenpairs should be discarded. Discarding either too many or too few eigenpairs may cause either losing information or including unnecessary noise. In [60] the authors proposed to use spectrum gap (the eigen-gap that divides eigenvalues to two well separable groups) or the known number of clusters to determine the threshold. In this work we adopt the later one, as the benchmark data sets used in our experiment do not have clear spectrum gap.

c.) Theorem 6 suggests that if only leading eigenpairs are used for computing feature scores, a high order rational function will increase robustness, since the leading eigenvalues are usually smaller than one. But, if all eigenpairs are used the robustness may decrease, since in this case we usually have $\sum \lambda_j^r \geq \sum \lambda_j$. However, in our experiments, we found that even all eigenpairs are involved, a high order rational function still helps improve performance. Two points may support our observation: first $\sum \lambda_j^r$ actually does not increase much comparing with $\sum \lambda_j$, for example, in our experiments, we found $\sum \lambda_j^3$ usually causes

an increases of the value by no more than 25 percents. Second, with a high order rational function, $\widehat{\varphi}_i(\cdot)$ s penalize the high frequency components in a more harsh way, and effectively increase the gap between the scores of relevant and irrelevant features, which makes the algorithm more robust to perturbations. In our experiments, we found the effect of gap increasement is usually more observable. Therefore, in reality, it is reasonable to use high order rational functions to improve performance. In the following, we empirically evaluate the performance of *SPEC* with both synthetic and real date.

2.5 Study on Synthetic Data

We provide evidences supporting our analysis in Section 2.4 with a synthetic test case. We will mainly focus on the effect of matrix perturbation, since the effect of feature perturbation is relatively straightforward⁵. To exam the effect of matrix perturbation, we construct the test case in the following way: (1) we generate a mixture of three gaussians in a 3D space as the target concept. The center of the three gaussians are $(2,0,0)$, $(0,2,0)$ and $(0,0,2)$, and they all have unit variance. The three dimensions containing the gaussians are denoted as F_1, F_2 , and F_3 and forms the set of relevant features. (2) Beside the 3 relevant features, we also generate 3 irrelevant features⁶ which have uniform distribution and are denoted as F_4, F_5 and F_6 . (3) We randomly sample 30 samples from each gaussian. (4) We generate the original Laplacian matrix \mathbb{L} based on F_1, F_2 , and F_3 , and the perturbation matrix \mathbb{L}_ϵ based on F_4, F_5 and F_6 . The similarity among samples are calculated using RBF kernel functions, see Equation 1.1. (5) The perturbed Laplacian matrix is generated by $\widetilde{\mathbb{L}} = (1 - \alpha)\mathbb{L} + \alpha\mathbb{L}_\epsilon$, where $0 \leq \alpha \leq 0.5$ and is used to control the scale of the perturbation.

⁵Feature perturbation may also cause matrix perturbation. As the Laplacian matrix can be formed in many different way, to simply the analysis, we decouple the two types of perturbations.

⁶Although only three irrelevant features are used for constructing the perturbation matrix, the obtained observations are valid for high dimensional case. This is because, our experiment is based on the scale of the perturbation matrix, which makes the effect of the irrelevant features less dependent on the feature number. Also as most filter algorithms, *SPEC* evaluates features individually.

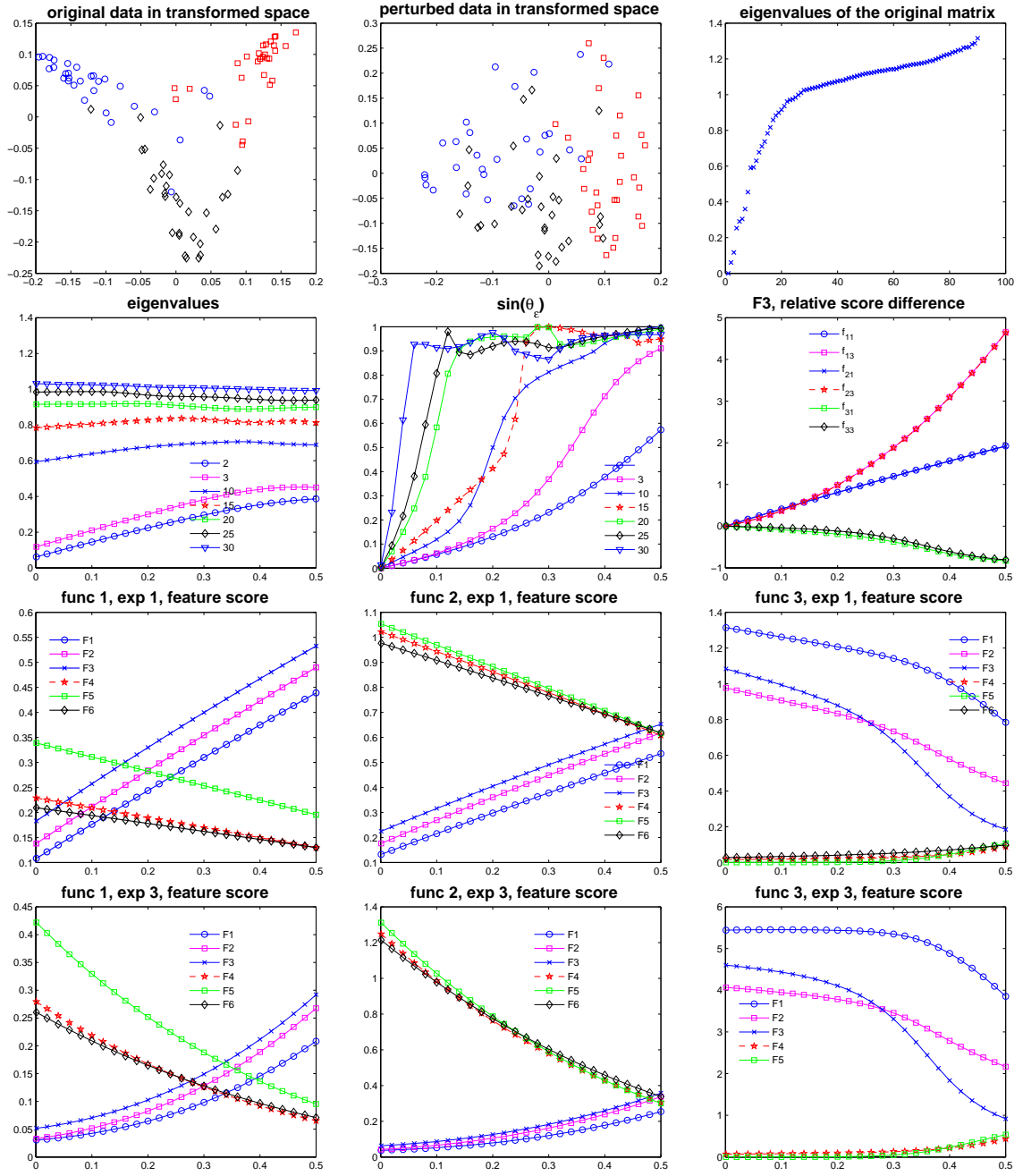


Figure 2.3: Empirical study of the effect of perturbation on SPECT.

Figure 2.3 shows the effect of perturbations with different scales. In the figure, the charts in the first row show the original and perturbed data (obtained by plotting the 2nd and the 3rd eigenvectors of \mathbb{L} and $\mathbb{L} + \mathbb{L}_\epsilon$) and the eigenvalues of \mathbb{L} , respectively. The charts in the second row depict the changes of the eigenvalues, the eigenvectors and the feature score, respectively. The remaining charts plot the feature score (Y axis) vs. different noise scale (X axis) with different feature ranking functions. Among the 6 features, F_1, F_2, F_3 are relevant, and the others are irrelevant. For $\varphi_1(\cdot)$ and $\varphi_2(\cdot)$, a smaller feature score indicates more relevant. While for $\varphi_3(\cdot)$, a bigger feature score indicates more relevant. Below we use chart- i - j to denote the chart at the i th row and j th column of the figure.

Chart-1-1 shows the distribution of the unperturbed data in the transformed space⁷. In the transformed space, samples form three well separable cluster structures. Chart-1-2, plots the distribution of the samples when the original data is perturbed using $\alpha = 0.5$, that is the perturbation has similar scale as the original matrix. The chart shows that with a big perturbation, the cluster structures become blur. Chart-1-3 shows the distribution of the eigenvalues of the original data. We can observe that the tail eigenvalues are packed in a small range. Comparing with them, the leading eigenvalues do have bigger eigen-gaps.

Chart-2-1 plots the values of the $\lambda_2, \lambda_3, \lambda_{10}, \lambda_{15}, \lambda_{25}$ and λ_{30} under perturbations of different scales. The chart shows that when more noise is added, the leading eigenvalues of $\tilde{\mathbb{L}}$ become bigger and the gap between leading and tail eigenvalues become smaller. This corresponds to the fact that when more noise is added, the cluster structures of the perturbed data become blur. Chart-2-2 plots the values of $\sin(\theta_\epsilon)$ when the scale of the perturbation varies, where θ_ϵ is the angle between ξ and $\tilde{\xi}$. The chart shows that the leading eigenvectors are more robust to noise, since they have bigger eigen-gap, which is consistent with the results presented in Theorem 4. Chart-2-3 plots the relative feature

⁷Since the dimensionality of the original data is greater than two, the data is plotted by using its 2nd and the 3rd eigenvectors of \mathbb{L} , which forms the Laplacian embedding [55] of the original data.

score differences related to different ranking functions for feature F_3 , when the scale of the perturbation varies. The relative feature score differences is define as: $c_{relative} = \frac{c-\tilde{c}}{c}$, where c is the feature score corresponding to the original data, and \tilde{c} is the feature score corresponding to the perturbed data. We use relative feature score differences, since the scores returned by $\varphi_1(\cdot)$ and $\varphi_2(\cdot)$ have different scale with $\varphi_3(\cdot)$. In the chart, f_{ij} denotes using $\varphi_i(\cdot)$ to rank features and using $\gamma(\cdot) = \lambda^j$ to rescale eigenvalues. From chart-2-3, it can be observed that $\varphi_3(\cdot)$ is more robust to noise, while $\varphi_1(\cdot)$ and $\varphi_2(\cdot)$ are of similar robustness, which is coherent to our analysis. From the chart it can also be observed that by using high order rational function to rescale the eigenvalues, the robustness for $\varphi_1(\cdot)$ and $\varphi_2(\cdot)$ decreases when the scale of the perturbation become bigger, while the robustness of $\varphi_3(\cdot)$ increases. Similar trends can also be observed on other features.

The chart-3-1 to chart-4-3 plot feature scores returned by different feature ranking functions when the scale of the perturbation varies. In the plots, “func i ” denotes $\varphi_i(\cdot)$ and “exp i ” denotes $\gamma(\cdot) = \lambda^i$. For $\varphi_1(\cdot)$ and $\varphi_2(\cdot)$, a smaller feature score means more relevant, while for $\varphi_3(\cdot)$, a bigger feature score means more relevant. From the charts, three observations can be obtained. First, among the three feature ranking functions, $\varphi_3(\cdot)$ is the most robust one to perturbations. As we can observe that when $\alpha = 0.5$, both $\varphi_1(\cdot)$ and $\varphi_2(\cdot)$ mix up relevant features with irrelevant ones, while $\varphi_3(\cdot)$ can still separate them. Second, comparing with $\varphi_1(\cdot)$, $\varphi_2(\cdot)$ is more robust due to the fact that the scores returned by $\varphi_2(\cdot)$ offer bigger gaps between the relevant and the irrelevant features. We also notice that the feature scores returned by $\varphi_1(\cdot)$ and $\varphi_2(\cdot)$ actually change in similar trend as the scale of the perturbation varies. It can be verified that the different behavior of $\varphi_1(\cdot)$ and $\varphi_2(\cdot)$ is caused by the mechanism of removing the trivial eigenpair, $(\lambda_1, \xi_1) = (0, \mathbf{D}^{\frac{1}{2}}\mathbf{1})$, from consideration. Recall $\varphi_2(\cdot) = \hat{\mathbf{f}}_i^T \mathbb{L} \hat{\mathbf{f}}_i \left(1 - (\hat{\mathbf{f}}_i^T \xi_1)^2 \right)^{-1}$, it turns out that the relevant features are usually far from ξ_1 ⁸, therefore have denominators near to 1. While, in contrast, the

⁸In terms of the angle between the two vectors.

irrelevant ones are often relatively closer to ξ_1 and have smaller denominators. This difference effectively increase the score gaps between the relevant and the irrelevant features. Third, a high order rational functions helps to increase the score gap too. From the chart we can observe that, although $\varphi_1(\lambda)$ mixes up the relevant with irrelevant features when $\alpha \approx 0.13$, with the help of $\gamma(\cdot) = \lambda^3$, $\tilde{\varphi}_1(\lambda)$ does not mixes up the relevant with irrelevant ones until $\alpha \approx 0.3$. Similar effect can also be observed on the other two feature ranking functions. In the next section we exam the efficacy of the feature selection methods derived from the proposed framework on real data for both supervised and unsupervised learning.

2.6 Study on Real Data

We compared the algorithms specified in Table 2.1 with two representative unsupervised feature selection algorithms: Laplacian Score and SEPER [13]; and three representative supervised feature selection algorithms: Fisher Score, ReliefF [80] and AROM-SVM [96]. Among the five baseline algorithms, AROM-SVM is of embedded model and the others are of filter model. All five algorithms provide weights for features to evaluate their relevance, and are feature weighting algorithms. The baseline algorithms used for comparison are all state-of-the-art feature selection algorithms, comparing with them enables us to examine the efficacy of the algorithms derived from SPEC. All algorithms are implemented with the spider toolbox⁹ under the MATLAB environment. We apply 1-nearest-neighbor (1NN) classifier on data sets with selected features, and use its accuracy to measure the quality of feature sets. All results reported in the paper are obtained by averaging the accuracy from 10 trials of experiments. In the experiments, for $\hat{\varphi}_3$, we set k equal to the number of different classes of the corresponding data set. The strategy is also commonly used for evaluating spectral clustering approaches [60].

⁹<http://www.kyb.tuebingen.mpg.de/bs/people/spider/>

Six benchmark data sets are used for experiments: HOCKBASE, RELATHE¹⁰, PIE10P¹¹, PIX10P¹², DATABASEC¹³ and COLON¹⁴. HOCKBASE & RELATHE are text data sets generated from the 20-new-group data: BASEBALL vs. HOCKEY (HOCKBASE) and RELIGION vs. ATHEISM (RELATHE). PIE10P & PIX10P are face image data sets containing 10 persons in each. DATABASEC and COLON are both gene expression data sets. All benchmark datasets are of very high dimensionality. A summary of the datasets is listed in Table 2.2.

Table 2.2: Summary of six benchmark data sets used in the experiment.

Data Set	Sample	Feature	Classes
HOCKBASE	1993	4862	2
RELATHE	1427	4322	2
PIE10P	210	10000	10
PIX10P	100	10000	10
DATABASEC	60	7129	2
COLON	62	2000	2

2.6.1 Study of Unsupervised Cases

In the experiment, we use weighted 10-nearest neighborhood graph to represent the similarity among samples. Figure 2.4 contains 12 plots of accuracy vs. different numbers of selected features, different ranking functions, different $\gamma(\cdot)$ and different similarity measures. As shown in the plots, in most cases, the majority of the algorithms proposed in Table 1 work better than the benchmark algorithms. Working with the Dijkstra function [94], SPEC achieves much better accuracy than using RBF kernel function on HOCKBASE and RELATHE data. While with RBF kernel, SPEC works better on PIE10P data. In general, the more features we select, the better accuracy we can achieve. However, in many cases, this

¹⁰<http://people.csail.mit.edu/jrennie/20Newsgroups/>

¹¹http://www.ri.cmu.edu/projects/project_418.html

¹²<http://peipa.essex.ac.uk/ipa/pix/faces/manchester/>

¹³<http://www.broad.mit.edu/mpr/publications/projects/CNS/>

¹⁴<http://microarray.princeton.edu/oncology/affydata/index.html>

trend is less pronounced when more than 40 features are selected. Using $\hat{\varphi}_2(\cdot)$, $\gamma(\mathbb{L}) = \mathbb{L}$ and the RBF kernel function, SPEC works exactly the same as Laplacian Score, showing the potential equivalence between this special case of SPEC and the Laplacian Score, which will be confirmed in Section 3. Table 2.3 shows the averaged accuracy when 10, 20, 30, 40, 50 features are selected. In the table, the accuracy with bold face is the highest or the second highest one without significant difference with the highest determined by t-test ($p\text{-val} > 0.1$). From the results we can observe that the differences between Laplacian Score and the algorithms performing best on each data set are: 0.17 for HOCKBASE, 0.07 for RELATHE, 0.15 for PIE10P, 0.18 for PIX10P, 0.09 for DATABASEC and 0.09 for COLON. And the differences for SEPER are: 0.19 for HOCKBASE, 0.07 for RELATHE, 0.24 for PIE10P, 0.17 for PIX10P, 0.16 for DATABASEC and 0.11 for COLON. A trend can be observed in the table is that $\hat{\varphi}_1(\cdot)$ performs well on the two text data sets, which contain binary classes; $\hat{\varphi}_3(\cdot)$ performances well on the two gene expression data sets and one image data set PIX10P, while $\hat{\varphi}_2(\cdot)$ works robustly on most data sets. We also observed that comparing with using \mathbb{L} , using \mathbb{L}^3 never downgrades performance significantly, while in certain cases, applying it can improve the performance by a big margin. For example on PIX10P with RBF kernel, by using φ_2, r^3 , we are able to achieve an improvement of 0.16 comparing with using φ_2, r . This observation shows that the high order spectral matrix function is helpful on improving learning performance by reducing noise.

According to the table, most of the highest accuracies are achieved when r^3 is used, this fact further justified the use of high order spectral matrix function to adjust the eigenvalues for the Laplacian matrix. From the experiment results, we can also observe that the high order spectral matrix function, r^3 , provides better performance than truncating the spectrum of the Laplacian matrix, $\hat{\varphi}_3$. For $\hat{\varphi}_3$ to perform well, one need to determine a proper threshold for discarding tail eigenpairs. In the experiments, we use the first k eigenpairs for calculating feature score, where k is equal to the number of different classes in the

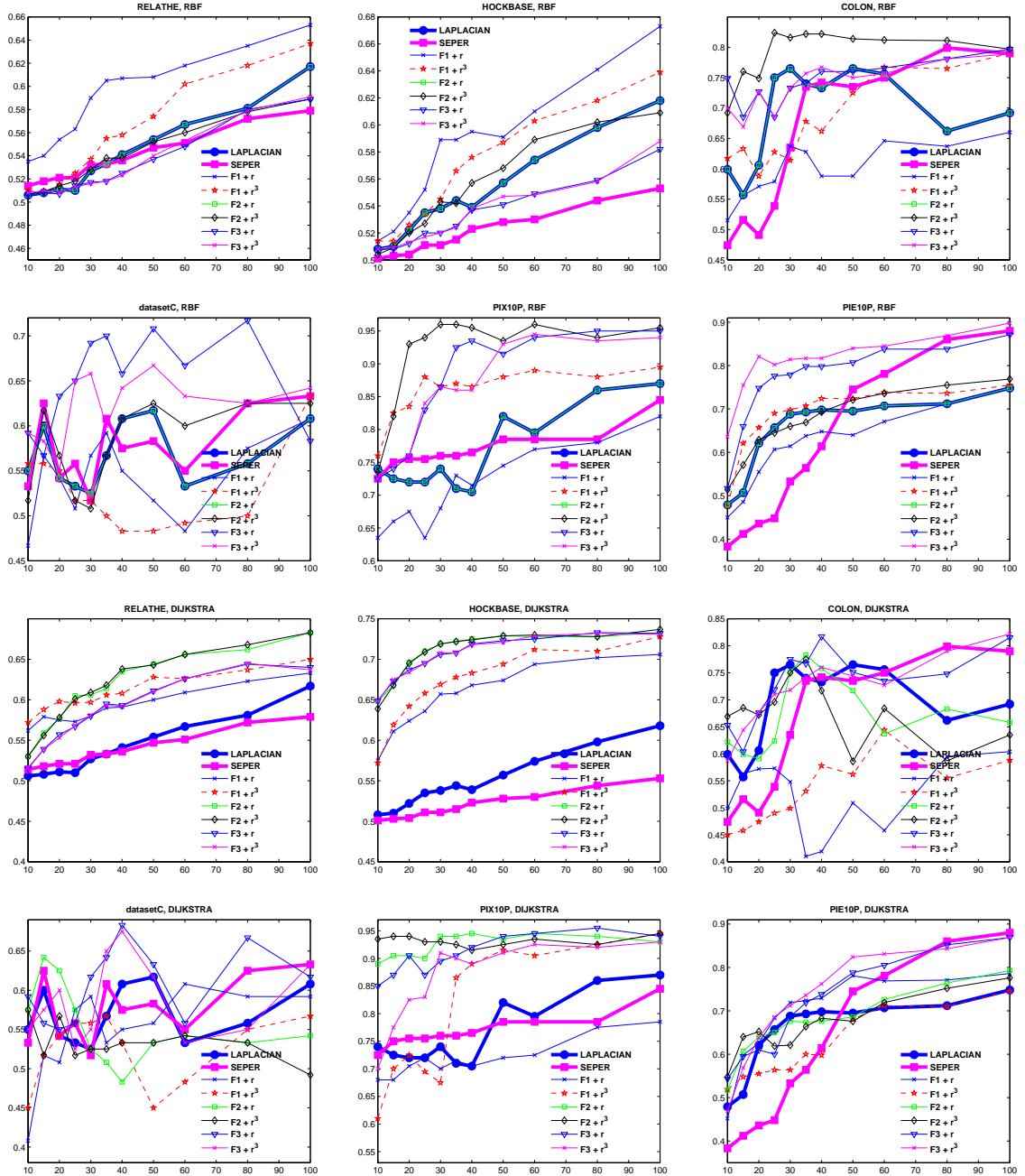


Figure 2.4: Study of unsupervised cases: accuracy (y axis) vs. different numbers of selected features (x axis). In the legend, F_i stands for $\hat{\phi}_i(\cdot)$.

Table 2.3: Study of unsupervised cases: averaged accuracy. DIJ stands for Dijkstra kernel, and LPSR for Laplacian Score. Accuracy with bold face is the highest or the second highest one without significant difference with the highest.

Kernel	LPSR	SEPER	$\hat{\varphi}_1, r$	$\hat{\varphi}_1, r^3$	$\hat{\varphi}_2, r$	$\hat{\varphi}_2, r^3$	$\hat{\varphi}_3, r$	$\hat{\varphi}_3, r^3$
RELATHE								
RBF	0.528	0.530	0.579	0.539	0.528	0.528	0.518	0.519
DIJ	0.528	0.530	0.582	0.601	0.598	0.600	0.571	0.570
HOCKBASE								
RBF	0.533	0.513	0.565	0.550	0.533	0.538	0.524	0.525
DIJ	0.533	0.513	0.640	0.652	0.703	0.701	0.697	0.696
PIE10P								
RBF	0.636	0.542	0.581	0.656	0.636	0.643	0.73	0.786
DIJ	0.636	0.542	0.661	0.583	0.639	0.636	0.673	0.681
PIX10P								
RBF	0.745	0.758	0.69	0.841	0.745	0.903	0.84	0.828
DIJ	0.745	0.758	0.702	0.763	0.923	0.929	0.902	0.847
COLON								
RBF	0.694	0.615	0.580	0.641	0.694	0.779	0.746	0.735
DIJ	0.694	0.615	0.509	0.513	0.688	0.679	0.734	0.697
DATASET C								
RBF	0.568	0.55	0.529	0.518	0.568	0.565	0.657	0.622
DIJ	0.568	0.55	0.523	0.507	0.548	0.547	0.615	0.598

corresponding data set. Since a data set may have subtle structures in each its cluster, we conjecture that using only the first k eigenpairs may cause losing useful information. In fact, in our experiment we did obtain better performance by manually tuning the threshold for discarding tail eigenpairs, which says that determining a proper threshold for spectrum truncation is crucial for $\hat{\varphi}_3$. We will conduct analysis on this issue in the next chapter.

2.6.2 Study of Supervised Cases

The similarity matrix defined in Eq. (3.13) is used with SPEC for supervised feature selection. It can be verified that this similarity matrix is a block matrix with rank k . And in this case the first k eigenvalues of \mathbb{L} are 0 and the others are 1. Hence, varying $\gamma(\cdot)$ will not affect the values of the three feature ranking functions, therefore we skip $\gamma(\cdot) =$

r^3 for supervised feature selection. Figure 2.5 shows the six plots for supervised feature selection. Table 2.4 shows the averaged accuracy when 10, 20, 30, 40 and 50 features are selected. In the table, accuracy with bold face is the highest or the second highest without significant difference with the highest determined by t-test ($p\text{-val} > 0.1$). From the results we can observe (1) generally supervised feature selection algorithms perform better than unsupervised feature selection algorithms, as they use label information; (2) generally, with the the similarity matrix defined in Equation 3.13, $\hat{\varphi}_2(\cdot)$ works the best, followed by $\hat{\varphi}_3(\cdot)$ and $\hat{\varphi}_1(\cdot)$ ¹⁵; and (3) with the the similarity matrix defined in Equation 3.13, $\hat{\varphi}_2(\cdot)$ works exactly the same as Fisher Score, showing the potential equivalence between this special case of SPEC and the Fisher Score, which will be confirmed in Section 3. We averaged the accuracy over different numbers of selected features and different data sets. Results show that SPEC($\hat{\varphi}_2$) and Fisher Score perform the best with an averaged accuracy of 0.78, followed by ReliefF whose accuracy is 0.76. The accuracy of SPEC($\hat{\varphi}_3$), SPEC($\hat{\varphi}_1$) and AROM-SVM are 0.73, 0.71 and 0.67, respectively.

Table 2.4: Study of supervised cases: averaged accuracy. Accuracy with bold face is the highest or the second highest without significant difference with the highest.

Data Set	ReliefF	Fisher	AROM-SVM	$\hat{\varphi}_1$	$\hat{\varphi}_2$	$\hat{\varphi}_3$
RELATHE	0.648	0.649	0.532	0.588	0.649	0.589
HOCKBASE	0.759	0.815	0.503	0.688	0.815	0.660
PIE10P	0.916	0.924	0.855	0.892	0.924	0.907
PIX10P	0.902	0.908	0.936	0.930	0.908	0.958
COLON	0.753	0.807	0.640	0.580	0.807	0.694
DATABASEC	0.553	0.577	0.467	0.507	0.577	0.537

2.7 Discussions

Feature selection algorithms can be either supervised or unsupervised [50]. We propose a general spectral feature selection framework, SPEC, for both supervised and unsupervised

¹⁵There is an exception on PIX10P data, where SPEC($\hat{\varphi}_3$) performs the best.

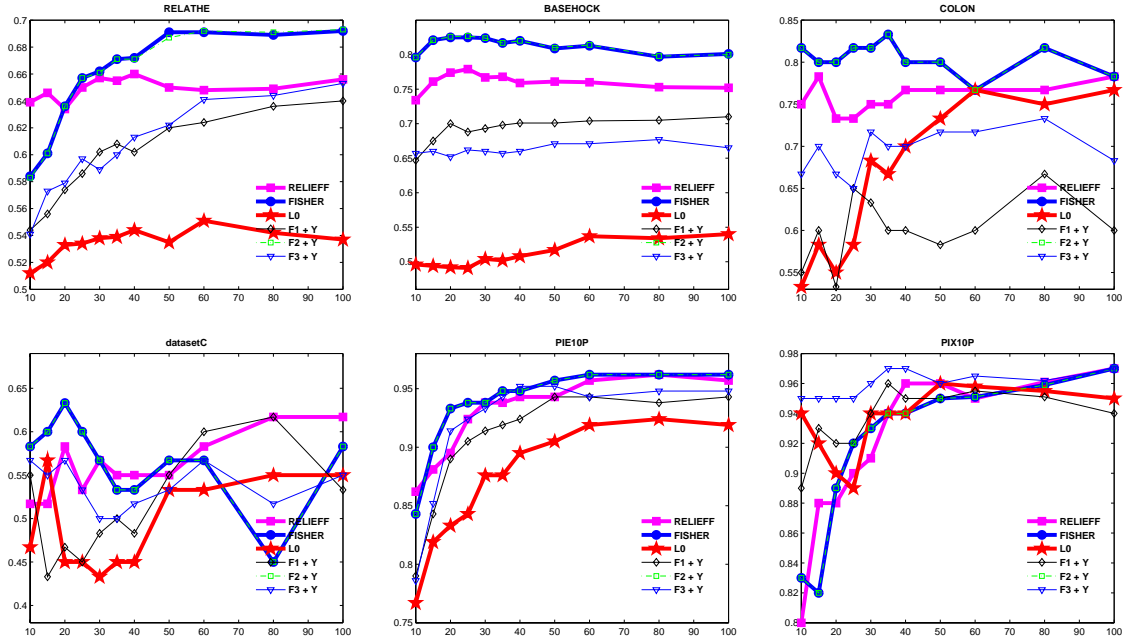


Figure 2.5: Study of supervised cases: accuracy (y axis) vs. different numbers of features (x axis). In the legend, F_i stands for $\hat{\phi}_i(\cdot)$. Y stands for the similarity matrix specified in Eq. (3.13), and L0 stands for AROM-SVM.

learning, which facilitates the joint study of supervised and unsupervised feature selection. We showed that families of effective algorithms can be derived from the framework. We conducted robustness analysis based on perturbation theory. The analysis enabled us to obtain better understanding about the behavior of the SPEC in a noisy learning environment. Extensive experiments exhibited the generality and usability of SPEC. One piece of work related to SPEC is [97], in which an unsupervised feature selection algorithm is proposed based on iteratively calculating the soft cluster indicator matrix and feature weight vector. The algorithm can be extended to handling data with labels. However the input of the algorithm restricts to covariance matrix, it does not handle general similarity matrix and cannot be extended to a general framework as the SPEC does.

The proposed framework consists of several components, the similarity matrix \mathbf{S} , the ranking function $\hat{\phi}(\cdot)$ and the spectral function $\gamma(\cdot)$. A proper configuration of the frame-

work ensures good performance. Based on our experimental results and observations, we offer the following guidelines for configuring SPEC. (1) The similarity matrix depicts the relationship among samples. A matrix which reflects the eccentrical relationships among samples is important for SPEC to select good features. An example is the HOCKBASE data, by using Dijkstra kernel, we achieved an improvement of 23% comparing with using the RBF kernel. (2) In noisy learning environment, either $\hat{\phi}_3(\cdot)$ or an high order rational function $\gamma(\cdot)$ is helpful for removing noise. (3) For data with clear spectrum gap, $\hat{\phi}_3(\cdot)$ may be very effective. Otherwise $\hat{\phi}_2(\cdot)$ could be a better way, since comparing with $\hat{\phi}_3(\cdot)$, $\hat{\phi}_2(\cdot)$ is less aggressive, and usually provide robust performance. (4) SPEC generates feature weighting algorithms. As most feature weighting algorithms, it does not consider feature redundancy, which may hurt the learning performance [102]. To address the problem, we will propose a multivariate formulation for spectral feature selection in Section 4.

SPEC works with general similarity matrices. It is natural to extend the framework for various applications. One interesting extension is the semi-supervised feature selection. In many real applications, such as text mining and image processing, data are abundant, but labeled data are costly to obtain. It is common to have a high dimensional data with large amount of unlabeled samples but only a few labeled samples. The data sets of this kind present a serious challenge, the so-called “small labeled-sample problem” [36], to supervised feature selection. That is, when the labeled sample size is too small to provide sufficient information about the target concept, supervised feature selection algorithms fail with either unintentionally removing many relevant features or selecting irrelevant features, which seems to be significant only on the small labeled data. Unsupervised feature selection algorithms can be an alternative in this case, as they are able to use the large amount of unlabeled data. However, as these algorithms ignore label information, important hints from labeled data are left out and this will generally downgrades the performance of unsupervised feature selection algorithms. Under the assumption that labeled and unlabeled

data are sampled from the same population generated by the target concept, using both labeled and unlabeled data is expected to better estimate feature relevance. The task of learning from mixed labeled and unlabeled data is of semi-supervised learning [11]. Spectral feature selection SPEC, can be naturally extended to achieve semi-supervised feature selection through a regularization framework, in which a feature’s relevance is evaluated by its consistency with both labeled and unlabeled data. The idea is formulated in [110] as:

$$\lambda \hat{\phi}(F_i) + (1 - \lambda)(1 - NMI(F_i, \mathbf{y})). \quad (2.16)$$

In Eq. (2.16), $NMI(F_i, \mathbf{y})$ is the normalized mutual information [102] between feature F_i and \mathbf{y} . The first term of Eq. (2.16) calculates the consistency of feature F_i on \mathbf{X} using one of the feature ranking function proposed in Section 2.2. The second term estimates the consistency of feature F_i with the labeled data. To be identified as relevant, a feature must be consistent according to the distribution of both the large amount of unlabeled data and the small amount of labeled data, which is illustrated in Figure 2.6. In the figure features assign similar values to samples that are in the same cluster, which is denoted by the ellipses. As we can see, since the cluster structure of the unlabeled data is ambiguous, feature F and feature F' are equally consistent to the distribution of unlabeled data(smooth). However, feature F is more consistent to labeled data, which suggests that F is more relevant.

The idea formulated in Eq. (2.16) forms one of the first semi-supervised feature selection algorithm, which is called *sSelect* [110]. Table 2.5 shows the averaged accuracy¹⁶ improvements (in percent) of the *1nn* classifier on the features selected by *sSelect* compared to the case when Fisher Score (supervised) and Laplacian Score (unsupervised) is used. The result suggest that using both labeled and unlabeled data does help feature selection. We refer readers to [106] for more detailed theoretical analysis and experimental results on semi-supervised feature selection algorithm *sSelect*.

¹⁶The averaged accuracy is obtained by averaging the accuracy achieved by the *1nn* classifier when different number of features are selected by the feature selection algorithms

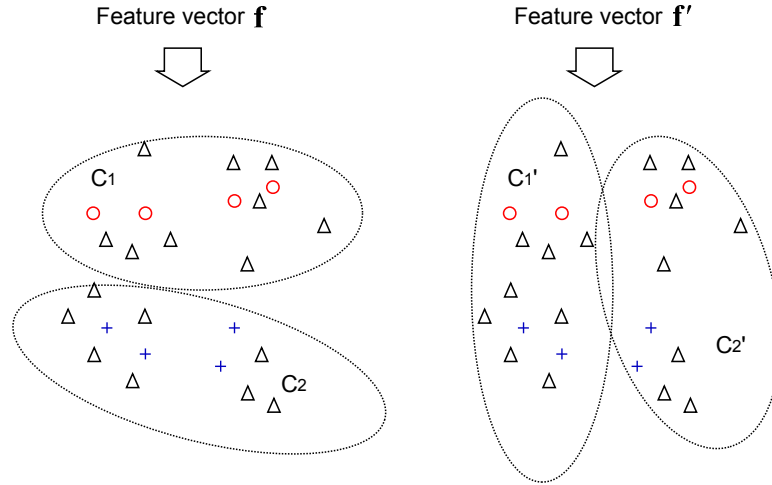


Figure 2.6: An exemplification of the idea behind semi-supervised feature selection.

Table 2.5: Average accuracy improvements of the $1nn$ classifier on the features selected by $sSelect$ on the BASEHOCK data set. L is for number of labeled data.

L	Fisher Score	Laplacian Score
2	5.60%	13.89%
6	8.36%	17.75%
10	11.50%	19.97%

One issue related to the framework formulated in Eq. (2.16) is that the regularization parameter λ is data dependent. That is for different data, the best value for the regularization parameter may vary. Therefore to find a proper value for the regularization parameter λ is crucial to ensure the performance of $sSelect$. In [106], a parameter tuning mechanism is developed based on studying the cut-value¹⁷ achieved on the data that only contains the selected features. We find the mechanism is often effective, but very time consuming. Therefore developing an efficient and effective technique for determine the value for the regularization parameter λ is an important work for future study.

¹⁷Given a data \mathbf{X} , the cut-value can be calculated by: first, constructing the adjacent matrix \mathbf{A} from \mathbf{X} ; then, forming the normalized Laplacian matrix \mathbb{L} using \mathbf{A} ; at the last, obtaining the cut-value by calculating the second smallest eigenvalue of \mathbb{L} .

Chapter 3

CONNECTIONS TO EXISTING ALGORITHMS

The proposed spectral feature selection framework, SPEC, is a general framework. It can systematically generate family of new feature selection algorithms, and it also unifies many existing feature selection algorithms as its special cases. In this chapter, we connect the proposed spectral feature selection framework to existing feature selection algorithms, which includes Laplacian Score [31], Fisher Score [19], ReliefF [80], Trace Ratio [62], and HSIC [85]. To achieve this, we first reformulate the relevance evaluation criteria proposed in the SPEC framework to a more general form:

$$\max_{\mathbb{F}_{sub}} \sum_{F \in \mathbb{F}_{sub}} \varphi(F) = \max_{\mathbb{F}_{sub}} \sum_{F \in \mathbb{F}_{sub}} \hat{\mathbf{f}}^\top \hat{\mathbf{S}} \hat{\mathbf{f}}, \quad \hat{\mathbf{f}} \in \mathbb{R}^n, \quad \hat{\mathbf{S}} \in \mathbb{R}^{n \times n} \quad (3.1)$$

where, \mathbb{F}_{sub} is the set of selected features, $\hat{\mathbf{f}}$ and $\hat{\mathbf{S}}$ are the normalized feature vector and normalized sample similarity matrix obtained from \mathbf{f} and \mathbf{S} according to certain normalization rules, respectively. We then show that the algorithms mentioned above can all be formulated in the form of Eq. (3.1). The only difference of these algorithms is that they use different rules to normalize \mathbf{f} and \mathbf{S} , which results in different $\hat{\mathbf{f}}$ and $\hat{\mathbf{S}}$.

It is shown in [77], that solving the following problem:

$$\max_{\mathbf{K} \succeq 0} \text{trace}(\mathbf{KS}) \quad \text{st.} \quad \text{trace}(\mathbf{K}) \leq 1, \quad (3.2)$$

will results in a kernel matrix \mathbf{K} , which well preserves the sample similarity specified in \mathbf{S} . Since $\max_{\mathbb{F}_{sub}} \sum_{F \in \mathbb{F}_{sub}} \hat{\mathbf{f}}^\top \hat{\mathbf{S}} \hat{\mathbf{f}} = \max_{\mathbb{F}_{sub}} \sum_{F \in \mathbb{F}_{sub}} \text{trace}(\hat{\mathbf{f}} \hat{\mathbf{f}}^\top \mathbf{S}) = \max_{\mathbb{F}_{sub}} \text{trace}\{(\sum_{F \in \mathbb{F}_{sub}} \hat{\mathbf{f}} \hat{\mathbf{f}}^\top) \mathbf{S}\}$. Also we have $\sum_{F \in \mathbb{F}_{sub}} \hat{\mathbf{f}} \hat{\mathbf{f}}^\top = \mathbf{X}_{\mathbb{F}_{sub}}^\top \mathbf{X}_{\mathbb{F}_{sub}}$, here $\mathbf{X}_{\mathbb{F}_{sub}}$ is the data containing only the features in \mathbb{F}_{sub} . Thus $\max_{\mathbb{F}_{sub}} \sum_{F \in \mathbb{F}_{sub}} \hat{\mathbf{f}}^\top \hat{\mathbf{S}} \hat{\mathbf{f}}$ equals to select a set of features \mathbb{F}_{sub} , such that the linear kernel on $\mathbf{X}_{\mathbb{F}_{sub}}$ can preserve the pairwise sample similarity specified in \mathbf{K} well. Therefore we can see that features maximize the value of Eq. (3.1), should have strong capability on

preserving the pairwise sample similarity specified in $\hat{\mathbf{S}}$. This can also be shown in a more intuitively way, since $\hat{\mathbf{f}}^\top \hat{\mathbf{S}} \hat{\mathbf{f}} = \sum_i \sum_j \hat{s}_{ij} \hat{f}_i \hat{f}_j$, under the assumption that features are normalized $\|\hat{\mathbf{f}}\| = 1$, to obtain a large value of Eq. (3.1), a feature must assign similar values to the samples that are similar to each other according to $\hat{\mathbf{S}}$. And this ensures the feature have strong capability on preserving the sample similarity specified in $\hat{\mathbf{S}}$. In the following sections, we will show that many existing successful feature selection algorithms can be formulated in the form of Eq. (3.1), therefore, although these algorithms are different in many ways, they are all special cases of the proposed spectral feature selection framework.

3.1 Reformulation for SPEC

We first study the case when the spectral matrix function is not applied. Given \mathbf{S} , the similarity matrix, with the following theorem, we show that the three feature ranking function $\varphi_1(\cdot)$, $\varphi_2(\cdot)$, $\varphi_3(\cdot)$ can be formulated in the common form of $\max_{\mathbb{F}_{sub}} \sum_{F \in \mathbb{F}_{sub}} \hat{\mathbf{f}}^\top \hat{\mathbf{S}} \hat{\mathbf{f}}$, with different definitions of $\hat{\mathbf{f}}$ and $\hat{\mathbf{S}}$. Here $\hat{\mathbf{f}}$ and $\hat{\mathbf{S}}$ are the normalized \mathbf{f} and \mathbf{S} .

Theorem 7 *Let $\hat{\mathbf{S}}$ be the similarity matrix, using SPEC to select k features can be achieved by maximizing the following objective function:*

$$\arg \max_{F_{i_1}, \dots, F_{i_k}} \sum_{j=1}^k \hat{\mathbf{f}}_{i_j}^\top \hat{\mathbf{S}} \hat{\mathbf{f}}_{i_j}. \quad (3.3)$$

When $\varphi_1(\cdot)$ is applied, $\hat{\mathbf{f}}$ and $\hat{\mathbf{S}}$ are defined by:

$$\hat{\mathbf{f}} = \frac{\mathbf{D}^{\frac{1}{2}} \mathbf{f}}{\|\mathbf{D}^{\frac{1}{2}} \mathbf{f}\|}, \quad \hat{\mathbf{S}} = \mathbf{D}^{-\frac{1}{2}} \mathbf{S} \mathbf{D}^{-\frac{1}{2}}. \quad (3.4)$$

When $\varphi_2(\cdot)$ is applied, let ξ_1 be the first eigenvector of \mathbb{L} , which is equal to $\|\mathbf{D}^{-\frac{1}{2}} \mathbf{1}\|_2^{-1} \mathbf{D}^{-\frac{1}{2}} \mathbf{1}$. $\hat{\mathbf{f}}$ and $\hat{\mathbf{S}}$ are defined by:

$$\tilde{\mathbf{f}} = \frac{\mathbf{D}^{\frac{1}{2}} \mathbf{f}}{\|\mathbf{D}^{\frac{1}{2}} \mathbf{f}\|}, \quad \hat{\mathbf{f}} = \frac{\tilde{\mathbf{f}} - \tilde{\mathbf{f}}^\top \xi_1 \xi_1}{\sqrt{1 - (\tilde{\mathbf{f}}^\top \xi_1)^2}}, \quad \hat{\mathbf{S}} = \mathbf{D}^{-\frac{1}{2}} \mathbf{S} \mathbf{D}^{-\frac{1}{2}}. \quad (3.5)$$

When $\varphi_3(\cdot)$ is applied, let $\mathbb{L} = \mathbf{U}\Sigma\mathbf{U}^\top$, where $\mathbf{U} = (\xi_1, \dots, \xi_n)$ and $\Sigma = \text{diag}(\lambda_1, \dots, \lambda_n)$ be the eigen decomposition of \mathbb{L} . $\hat{\mathbf{f}}$ and $\hat{\mathbf{S}}$ are defined by:

$$\hat{\mathbf{f}} = \frac{\mathbf{D}^{\frac{1}{2}}\mathbf{f}}{\|\mathbf{D}^{\frac{1}{2}}\mathbf{f}\|}, \quad \hat{\mathbf{S}} = \mathbf{U}_k(2\mathbf{I} - \Sigma_k)\mathbf{U}_k^\top. \quad \mathbf{U}_k = (\xi_2, \dots, \xi_k), \quad \Sigma_k = \text{diag}(\lambda_2, \dots, \lambda_k) \quad (3.6)$$

Proof: We start from $\varphi_1(\cdot)$. It is easy to verify that $\varphi_1(F) = \hat{\mathbf{f}}^\top (\mathbf{I} - \hat{\mathbf{S}}) \hat{\mathbf{f}}$. In SPEC, features are evaluated independently, therefore using $\varphi_1(\cdot)$ to select k features can be achieved by picking the top k features, which have the smallest $\varphi_1(\cdot)$ values. This process can be formulated as the following optimization problem,

$$\arg \min_{F_{i_1}, \dots, F_{i_k}} \sum_{j=1}^k \hat{\mathbf{f}}_{i_j}^\top (\mathbf{I} - \hat{\mathbf{S}}) \hat{\mathbf{f}}_{i_j}.$$

Note, in the above equation, features are actually evaluated independently. Also we have,

$$\arg \min_{F_{i_1}, \dots, F_{i_k}} \sum_{j=1}^k \hat{\mathbf{f}}_{i_j}^\top (\mathbf{I} - \hat{\mathbf{S}}) \hat{\mathbf{f}}_{i_j} = \arg \max_{F_{i_1}, \dots, F_{i_k}} \sum_{j=1}^k \hat{\mathbf{f}}_{i_j}^\top \hat{\mathbf{S}} \hat{\mathbf{f}}_{i_j}.$$

This is true, since features have been normalized and $\hat{\mathbf{f}}_1, \dots, \hat{\mathbf{f}}_m$ all have unit norm. We prove the case when $\varphi_1(\cdot)$ is applied in SPEC for feature selection.

When $\varphi_2(\cdot)$ is applied, first we can show that $\|\hat{\mathbf{f}}\| = 1$. Second, since the first eigenvalue of \mathbb{L} is equal to zero, $\lambda_1 = 0$, we have $\xi_1^\top \hat{\mathbf{L}} \mathbf{x} = 0$ for any $\mathbf{x} \in \mathbb{R}^{n \times 1}$. By noticing the above fact, we can prove the equivalence for the cases when $\varphi_2(\cdot)$ is applied.

Also it is easy to verify the following equation:

$$\mathbf{f}^\top \mathbf{U}_k (2\mathbf{I} - \Sigma_k) \mathbf{U}_k^\top \mathbf{f} = \sum_{j=2}^k (2 - \lambda_j) \alpha_j^2 = \varphi_3(F).$$

This allow us to prove the equivalence for the cases when $\varphi_3(\cdot)$ is used in SPEC. ■

Theorem 8 shows that when $\varphi_1(\cdot)$ and $\varphi_2(\cdot)$ are used in SPEC, it tries to preserve the sample similarity specified by $\mathbf{D}^{-\frac{1}{2}}\mathbf{S}\mathbf{D}^{-\frac{1}{2}}$, which is the normalized sample similarity. And

when $\varphi_3(\cdot)$ is used, it tries to preserve the sample similarity specified by $\mathbf{U}_k(2\mathbf{I} - \Sigma_k)\mathbf{U}_k^\top$, which is derived from \mathbb{L} by adjusting the leading eigenvalues and discarding the tail eigenpairs. It is known that the tail eigen-pairs of \mathbb{L} correspond to noise. In $\varphi_1(\cdot)$ and $\varphi_3(\cdot)$, features are normalized by $\mathbf{D}^{\frac{1}{2}}\mathbf{f}$, which form the density reweighted features [31]. And they are normalized to have unit norm. This step emphasizes positions in a feature, which correspond to samples from neighborhoods with dense sample distribution. In $\varphi_2(\cdot)$, there is a second normalization step: features are first made to be orthogonal to ξ_1 , and then normalized to have unit norm. This step removes ξ_1 from consideration. As mentioned above that by aligning closely to ξ_1 , a feature can achieve a big $\varphi_1(\cdot)$ value, however, ξ_1 only capture the density information of the data. The second normalization step ensures that we will not assign high relevance scores to features being aligned closely to ξ_1 .

Similarly, when the spectral matrix function $\gamma(\cdot)$ is used in SPEC. By the following theorem, we show that the three feature ranking function $\hat{\varphi}_1(\cdot)$, $\hat{\varphi}_2(\cdot)$, $\hat{\varphi}_3(\cdot)$ can also be formulated in the form of $\max_{\mathbb{F}_{sub}} \sum_{F \in \mathbb{F}_{sub}} \hat{\mathbf{f}}^\top \hat{\mathbf{S}} \hat{\mathbf{f}}$, with different definition of $\hat{\mathbf{f}}$ and $\hat{\mathbf{S}}$.

Theorem 8 *Let $\hat{\mathbf{S}}$ be the similarity matrix and $\gamma(\cdot)$, using SPEC to select k features can be achieved by maximizing the following objective function:*

$$\arg \max_{F_1, \dots, F_k} \sum_{j=1}^k \hat{\mathbf{f}}_{i_j}^\top \hat{\mathbf{S}} \hat{\mathbf{f}}_{i_j}. \quad (3.7)$$

Let $\mathbb{L} = \mathbf{U}\Sigma\mathbf{U}^\top$, where $\mathbf{U} = (\xi_1, \dots, \xi_n)$ and $\Sigma = \text{diag}(\lambda_1, \dots, \lambda_n)$

When $\hat{\varphi}_1(\cdot)$ is applied, $\hat{\mathbf{f}}$ and $\hat{\mathbf{S}}$ are defined by:

$$\hat{\mathbf{f}} = \frac{\mathbf{D}^{\frac{1}{2}}\mathbf{f}}{\|\mathbf{D}^{\frac{1}{2}}\mathbf{f}\|}, \quad \hat{\mathbf{S}} = \mathbf{U}(\mathbf{I} - \gamma(\Sigma))\mathbf{U}^\top. \quad (3.8)$$

When $\hat{\varphi}_2(\cdot)$ is applied, let ξ_1 be the first eigenvector of \mathbb{L} , which is equal to $\|\mathbf{D}^{-\frac{1}{2}}\mathbf{1}\|_2^{-1}\mathbf{D}^{-\frac{1}{2}}\mathbf{1}$.

$\hat{\mathbf{f}}$ and $\hat{\mathbf{S}}$ are defined by:

$$\tilde{\mathbf{f}} = \frac{\mathbf{D}^{\frac{1}{2}}\mathbf{f}}{\|\mathbf{D}^{\frac{1}{2}}\mathbf{f}\|}, \quad \hat{\mathbf{f}} = \frac{\tilde{\mathbf{f}} - \tilde{\mathbf{f}}^\top \xi_1 \xi_1}{\sqrt{1 - (\tilde{\mathbf{f}}^\top \xi_1)^2}}, \quad \hat{\mathbf{S}} = \mathbf{U}(\mathbf{I} - \gamma(\Sigma))\mathbf{U}^\top. \quad (3.9)$$

When $\hat{\phi}_3(\cdot)$ is applied, $\hat{\mathbf{f}}$ and $\hat{\mathbf{S}}$ are defined by:

$$\hat{\mathbf{f}} = \frac{\mathbf{D}^{\frac{1}{2}}\mathbf{f}}{\|\mathbf{D}^{\frac{1}{2}}\mathbf{f}\|}, \quad \hat{\mathbf{S}} = \mathbf{U}_k(\gamma(2\mathbf{I}) - \gamma(\Sigma_k)) \mathbf{U}_k^\top. \quad \mathbf{U}_k = (\xi_2, \dots, \xi_k), \quad \Sigma_k = \text{diag}(\lambda_2, \dots, \lambda_k) \quad (3.10)$$

3.2 Reformulation for Laplacian Score

Laplacian Score [31] is a unsupervised feature weighting algorithms with filter model. Given an affinity matrix \mathbf{S} , its corresponding degree matrix \mathbf{D} and laplacian matrix \mathbf{L} , the Laplacian Score of a feature \mathbf{f} is calculated via:

$$\varphi_L(\mathbf{f}) = \frac{\tilde{\mathbf{f}}^\top \mathbf{L} \tilde{\mathbf{f}}}{\tilde{\mathbf{f}}^\top \mathbf{D} \tilde{\mathbf{f}}}, \quad \text{where } \tilde{\mathbf{f}} = \mathbf{f} - \frac{\mathbf{f}^\top \mathbf{D} \mathbf{1}}{\mathbf{1}^\top \mathbf{D} \mathbf{1}} \mathbf{1}. \quad (3.11)$$

With the following theorem, we show that $\varphi_2(\mathbf{f}) = \varphi_L(\mathbf{f})$, Here $\varphi_2(\cdot)$ is the second feature ranking function defined in SPEC without using the matrix spectral function $\gamma(\cdot)$.

Theorem 9 *The unsupervised feature selection algorithm Laplacian Score [31] is a special case of SPEC, by setting $\hat{\phi}(\cdot) = \varphi_2(\cdot)$ in SPEC.*

Proof: The ranking function of Laplacian score is:

$$\varphi_L(F) = \frac{\tilde{\mathbf{f}}^\top \mathbf{L} \tilde{\mathbf{f}}}{\tilde{\mathbf{f}}^\top \mathbf{D} \tilde{\mathbf{f}}}, \quad \text{where } \tilde{\mathbf{f}} = \mathbf{f} - \frac{\mathbf{f}^\top \mathbf{D} \mathbf{1}}{\mathbf{1}^\top \mathbf{D} \mathbf{1}} \mathbf{1}.$$

Substituting $\tilde{\mathbf{f}}$ in $\varphi_L(F)$ and applying Proposition 1, we have:

$$\varphi_L(F) = \frac{\mathbf{f}^\top \mathbf{L} \mathbf{f}}{\mathbf{f}^\top \mathbf{D} \mathbf{f} - \frac{(\mathbf{f}^\top \mathbf{D} \mathbf{1})^2}{\mathbf{1}^\top \mathbf{D} \mathbf{1}}} = \frac{(\mathbf{D}^{\frac{1}{2}}\mathbf{f})^\top \mathbf{L} (\mathbf{D}^{\frac{1}{2}}\mathbf{f})}{(\mathbf{D}^{\frac{1}{2}}\mathbf{f})^\top (\mathbf{D}^{\frac{1}{2}}\mathbf{f}) - \frac{\left((\mathbf{D}^{\frac{1}{2}}\mathbf{f})^\top (\mathbf{D}^{\frac{1}{2}}\mathbf{1}) \right)^2}{(\mathbf{D}^{\frac{1}{2}}\mathbf{1})^\top (\mathbf{D}^{\frac{1}{2}}\mathbf{1})}}$$

As $\xi_1 = \frac{\mathbf{D}^{\frac{1}{2}}\mathbf{1}}{\|\mathbf{D}^{\frac{1}{2}}\mathbf{1}\|}$ and $\hat{\mathbf{f}} = \frac{\mathbf{D}^{\frac{1}{2}}\mathbf{f}}{\|\mathbf{D}^{\frac{1}{2}}\mathbf{f}\|}$, we have:

$$\varphi_L(F) = \frac{\hat{\mathbf{f}}^\top \mathbf{L} \hat{\mathbf{f}}}{1 - (\hat{\mathbf{f}}^\top \xi_1)^2} = \varphi_2(\cdot)$$

■

Theorem 9 shows that Laplacian Score is a special case of SPEC, which explains our observation in Section 2.6.1 that when $\widehat{\varphi}_2(\cdot)$ is used, and $\gamma(\mathbb{L}) = \mathbb{L}$, SPEC works exactly the same as Laplacian Score. Based on this theorem, we claim:

Theorem 10 *Let $\widehat{\mathbf{S}}$ be the similarity matrix, using Laplacian Score to select k features can be achieved by maximizing the following objective function:*

$$\arg \max_{F_{i_1}, \dots, F_{i_k}} \sum_{j=1}^k \widehat{\mathbf{f}}_{i_j}^\top \widehat{\mathbf{S}} \widehat{\mathbf{f}}_{i_j}.$$

Let ξ_1 be the first eigenvector of \mathbb{L} , $\widehat{\mathbf{f}}$ and $\widehat{\mathbf{S}}$ are defined as:

$$\tilde{\mathbf{f}} = \frac{\mathbf{D}^{\frac{1}{2}} \mathbf{f}}{\|\mathbf{D}^{\frac{1}{2}} \mathbf{f}\|}, \widehat{\mathbf{f}} = \frac{\tilde{\mathbf{f}} - \tilde{\mathbf{f}}^\top \xi_1 \xi_1}{\sqrt{1 - (\tilde{\mathbf{f}}^\top \xi_1)^2}}, \widehat{\mathbf{S}} = \mathbf{D}^{-\frac{1}{2}} \mathbf{S} \mathbf{D}^{-\frac{1}{2}}.$$

3.3 Reformulation for Fisher Score

Fisher Score [19] is a supervised feature weighting algorithms with filter model. Given the class label $\mathbf{y} = \{y_1, \dots, y_n\}$, Fisher Score select features, that assign similar values to the samples from the same class, and assign different values to samples from different classes.

The evaluation criterion used in Fisher Score can be formulated as:

$$\varphi_F(F_i) = \frac{\sum_{j=1}^c n_j (\mu_j - \mu)^2}{\sum_{j=1}^c n_j \sigma_j^2} \quad (3.12)$$

In the equation, μ is the mean of the feature \mathbf{f}_i , n_j is the number of samples in the j th class, and μ_j and σ_j is the mean and the variance of \mathbf{f}_i on class j , respectively.

As shown in [31], when the similarity matrix \mathbf{S} is derived from the class label as:

$$\mathbf{S}_{ij}^{FIS} = \begin{cases} \frac{1}{n_l}, & y_i = y_j = l \\ 0, & \text{otherwise} \end{cases} \quad (3.13)$$

Laplacian Score and Fisher Score are equivalent, and have the following relationship:

$$\varphi_L(F_i) = \frac{1}{1 + \varphi_F(F_i)} \quad (3.14)$$

Therefore we have the following theorem:

Theorem 11 *Let $\hat{\mathbf{S}}$ be the similarity matrix defined in Eq. (3.13), using Fisher Score to select k features can be achieved by maximizing the following objective function:*

$$\arg \max_{F_{i_1}, \dots, F_{i_k}} \sum_{j=1}^k \hat{\mathbf{f}}_{i_j}^\top \hat{\mathbf{S}} \hat{\mathbf{f}}_{i_j}.$$

Let ξ_1 be the first eigenvector of \mathbb{L} , which is derived from \mathbf{S} , $\hat{\mathbf{f}}$ and $\hat{\mathbf{S}}$ are defined as:

$$\tilde{\mathbf{f}} = \frac{\mathbf{D}^{\frac{1}{2}} \mathbf{f}}{\|\mathbf{D}^{\frac{1}{2}} \mathbf{f}\|}, \hat{\mathbf{f}} = \frac{\tilde{\mathbf{f}} - \tilde{\mathbf{f}}^\top \xi_1 \xi_1}{\sqrt{1 - (\tilde{\mathbf{f}}^\top \xi_1)^2}}, \hat{\mathbf{S}} = \mathbf{D}^{-\frac{1}{2}} \mathbf{S} \mathbf{D}^{-\frac{1}{2}}.$$

3.4 Reformulation of ReliefF

Relief [41] and its multiclass extension ReliefF [42] are supervised feature weighting algorithms with filter model. Assume M instances are randomly sampled from data, the feature evaluation criterion of Relief is define as:

$$\varphi_R(F_i) = \frac{1}{2} \sum_{t=1}^M (\|x_{t,i} - NM(\mathbf{x}_t)_i\| - \|x_{t,i} - NH(\mathbf{x}_t)_i\|).$$

In the equation, $x_{t,i}$ denotes the value of instance \mathbf{x}_t on feature \mathbf{f}_i , $NH(\mathbf{x})$ and $NM(\mathbf{x})$ denote the nearest points to \mathbf{x} in the data with the same and different label respectively, and $\|\cdot\|$ is a distance measurement. To handle multiclass problems, the above evaluation metric is extended and has the following form:

$$\begin{aligned} \varphi_R(F_i) = \frac{1}{M} \cdot \sum_{t=1}^M \left\{ -\frac{1}{M_{t,CL(\mathbf{x}_t)}} \sum_{\mathbf{x}_j \in NH(\mathbf{x}_t)} \|x_{t,i} - x_{j,i}\| \right. \\ \left. + \sum_{C \neq CL(\mathbf{x}_t)} \frac{P(C)}{1 - P(CL(\mathbf{x}_t))} \times \frac{1}{M_{t,C}} \times \right. \\ \left. \sum_{\mathbf{x}_j \in NM(\mathbf{x}_t, C)} \|x_{t,i} - x_{j,i}\| \right\}. \end{aligned} \quad (3.15)$$

Here, $CL(\mathbf{x}_t)$ returns the class label of the instance \mathbf{x}_t and $P(C)$ is the probability of a instances being from the class C . $x_{t,i}$ is the value of the feature \mathbf{f}_i on instance \mathbf{x}_t ; $NH(\mathbf{x})$ or $NM(\mathbf{x}, C)$ denote a set of nearest points to \mathbf{x} with the same class of \mathbf{x} , or a different class (the class C), respectively. $M_{t,CL(\mathbf{x}_t)}$ is the size of the set $NH(\mathbf{x})$, and $M_{t,C}$ is the size of the set $NM(\mathbf{x}, C)$. Usually, the size of both $NH(\mathbf{x})$ and $NM(\mathbf{x}, C)$, $\forall C \neq CL(\mathbf{x})$, is set to a pre-specified constant h . Eq. (3.15) forms the relevance evaluation criterion of ReliefF. The relevance evaluation criteria of Relief and ReliefF tell that the two algorithms seek features contribute the separation of samples from different classes.

Assume the training data has c classes with l instances in each class, there are h instances in both $NH(\mathbf{x})$ and $NM(\mathbf{x})$ and all features have been normalized to have unit norm. As show in [108], under the specified assumptions, the feature relevance evaluation criterion of ReliefF can be formulated using the following equation:

$$\sum_{i=1}^n \left(\sum_{j=1}^k \frac{1}{k} (f_i - f_{NH_j})^2 - \sum_{C \neq y_i} \frac{\sum_{j=1}^k (f_i - f_{NM_{C,j}})^2}{(c-1)k} \right) \quad (3.16)$$

In the equation, f_i is the value of feature \mathbf{f} on the i th instance, \mathbf{x}_i ; NH_j denotes the j th nearest hit of \mathbf{x}_i ; and $NM_{C,j}$ denotes the j th nearest miss of \mathbf{x}_i in class C . Here we use the Euclidean distance to calculate the difference between two values and use all training data to train ReliefF. When the similarity matrix S is defined as:

$$\mathbf{S}_{i,j}^{REL} = \begin{cases} 1 & i = j \\ -\frac{1}{k} & x_j \in NH(\mathbf{x}_i) \\ \frac{1}{(c-1)k} & x_j \in NM(\mathbf{x}_i, CL(\mathbf{x}_i)) \end{cases}, \quad (3.17)$$

Assuming that if $x_j \in NH(\mathbf{x}_i)$, we also have $x_i \in NH(\mathbf{x}_j)$; and if $x_j \in NM(\mathbf{x}_i, CL(\mathbf{x}_i))$, we also have $x_i \in NM(\mathbf{x}_j, CL(\mathbf{x}_j))$, which ensures \mathbf{S}^{REL} is symmetric. By applying Proposition 1, it is easy to verify that $\mathbf{D} = \mathbf{I}$, and $\varphi_R(F_i) = -1 + \mathbf{f}^\top \mathbf{S}^{REL} \mathbf{f}$ is equivalent to the evaluation criterion defined in Eq. (3.16) up to a constant. By the fact that $\varphi_1(F_i) = -1 + \mathbf{f}^\top \mathbf{S}^{REL} \mathbf{f}$,

where $\varphi_1(\cdot)$ is the first feature ranking function defined in SPEC without using spectral matrix function, we can see that under the assumptions, ReliefF also forms a special case of SPEC. Based on the above observation we have the following theorem:

Theorem 12 *Let $\hat{\mathbf{S}}$ be the similarity matrix defined in Eq. (3.17), using ReliefF to select k features can be achieved by maximizing the following objective function:*

$$\arg \max_{F_{i_1}, \dots, F_{i_k}} \sum_{j=1}^k \hat{\mathbf{f}}_{i_j}^\top \hat{\mathbf{S}} \hat{\mathbf{f}}_{i_j}.$$

Here, $\hat{\mathbf{f}}$ and $\hat{\mathbf{S}}$ are defined by:

$$\hat{\mathbf{f}} = \frac{\mathbf{D}^{\frac{1}{2}} \mathbf{f}}{\|\mathbf{D}^{\frac{1}{2}} \mathbf{f}\|}, \quad \hat{\mathbf{S}} = \mathbf{D}^{-\frac{1}{2}} \mathbf{S} \mathbf{D}^{-\frac{1}{2}}. \quad (3.18)$$

3.5 Reformulation for Trace Ratio Criterion

The trace ratio criterion for subset-level feature selection is proposed in [62]. It defines two affinity matrices \mathbf{S}_w and \mathbf{S}_b . \mathbf{S}_w represents the within-class or local affinity relationship of instances, whereas \mathbf{S}_b represents the between-class or the global counterpart. Two graphs \mathbb{G}_w and \mathbb{G}_b can be constructed and their corresponding graph Laplacian matrices are \mathbf{L}_w and \mathbf{L}_b , respectively. Assume we want to select k , $\mathbf{W} = [\mathbf{w}_{i_1}, \mathbf{w}_{i_2}, \dots, \mathbf{w}_{i_k}] \in \mathbb{R}^{n \times k}$ is the selection matrix, where the column vector \mathbf{w}_{i_j} has one and only one “1” at its i_j -th element, and $\{i_1, i_2, \dots, i_k\} \in \{1, 2, \dots, n\}$. The Trace Ratio Criterion seeks the best selection matrix \mathbf{W} by maximizing the following objective function:

$$\mathbf{W}^* = \arg \max_{\mathbf{W}} \frac{\text{trace}(\mathbf{W}^\top \mathbf{X}^\top \mathbf{L}_b \mathbf{X} \mathbf{W})}{\text{trace}(\mathbf{W}^\top \mathbf{X}^\top \mathbf{L}_w \mathbf{X} \mathbf{W})}. \quad (3.19)$$

As shown in [62], the optimal solution of the problem can be obtained by iteratively solving the following two subproblems. First, when λ_i is mixed, we solve the problem (P1):

$$(P1) : \mathbf{W}_{i+1} = \arg \max_{\mathbf{W}} \text{trace} \left(\mathbf{W}^\top \mathbf{X}^\top (\mathbf{L}_b - \lambda_i \mathbf{L}_w) \mathbf{X} \mathbf{W} \right). \quad (3.20)$$

Second, when \mathbf{W}_i is fixed, we solve the problem (P2):

$$(P2): \lambda_{i+1} = \frac{\text{trace}(\mathbf{W}_{i+1}^\top \mathbf{X}^\top \mathbf{L}_b \mathbf{X} \mathbf{W}_{i+1})}{\text{trace}(\mathbf{W}_{i+1}^\top \mathbf{X}^\top \mathbf{L}_w \mathbf{X} \mathbf{W}_{i+1})}. \quad (3.21)$$

Since $\text{trace}(\mathbf{W}^\top \mathbf{X}^\top (\mathbf{L}_b - \lambda \mathbf{L}_w) \mathbf{X} \mathbf{W}) = \sum_{i \in \{i_1, i_2, \dots, i_k\}} \mathbf{f}_i^\top (\mathbf{L}_b - \lambda \mathbf{L}_w) \mathbf{f}_i$, it is easy to see that when λ is fixed, the subproblem (P1) can be solved by picking the top k features with large $\mathbf{f}_i^\top (\mathbf{L}_b - \lambda \mathbf{L}_w) \mathbf{f}_i$ values. Therefore, although the Trace Ratio Criterion is proposed for subset feature selection, in feature selection process, which corresponds to solving the subproblem (P1), features are actually evaluated independently.

Based on the above analysis, we have the following theorem to built the connection between the Trace Ratio Criterion and the SPEC framework.

Theorem 13 *Assume λ^* is the optimal λ for the problem defined in Eq. (3.19), using the Trace Ratio Criterion to select k features can be achieved by maximizing the following objective function:*

$$\arg \max_{F_{i_1}, \dots, F_{i_k}} \sum_{j=1}^k \hat{\mathbf{f}}_{i_j}^\top \hat{\mathbf{S}} \hat{\mathbf{f}}_{i_j}.$$

Here, $\hat{\mathbf{f}}$ and $\hat{\mathbf{S}}$ are defined by:

$$\hat{\mathbf{f}} = \mathbf{f}, \quad \hat{\mathbf{S}} = (\mathbf{L}_b - \lambda^* \mathbf{L}_w) \quad (3.22)$$

The theorem suggests that to maximize $\mathbf{f}^\top (\mathbf{L}_b - \lambda \mathbf{L}_w) \mathbf{f}$, a feature needs to simultaneously maximize $\mathbf{f}^\top \mathbf{L}_b \mathbf{f}$, which requires it to assign different values to samples that are from different classes, and minimize $\mathbf{f}^\top \mathbf{L}_w \mathbf{f}$, which requires it to assign similar values to samples that are from the same class. And the λ is the regularization parameter to balance the two components in the criterion. It can be seen that the Trace Ratio Criterion selects features in a similar way as the Fisher Score. Actually, it is shown in [62], that with specific definitions for \mathbf{L}_w and \mathbf{L}_b , the Trace Ratio Criterion is equivalent to Fisher Score.

3.6 Reformulation for HSIC Criterion

HSIC is first proposed in [27] for measuring the dependence between two kernels. In [85], HSIC is applied for feature selection, and the basic idea is to select a subset of features, such that the kernel constructed using the feature subset maximize the HSIC criterion when compared to a given kernel \mathbf{K} . In [85] an unbiased estimator of HSIC is given as:

$$\varphi_H(\mathbf{F}) = \frac{1}{n(n-3)} \left[\text{Trace}(\mathbf{K}_F \mathbf{K}) + \frac{\mathbf{1}^\top \mathbf{K}_F \mathbf{1} \mathbf{1}^\top \mathbf{K} \mathbf{1}}{(n-1)(n-2)} \frac{2}{n-2} \mathbf{1}^\top \mathbf{K}_F \mathbf{K} \mathbf{1} \right]. \quad (3.23)$$

In the equation, \mathbf{F} is a subset of the original features and \mathbf{K}_F is the kernel obtained from \mathbf{F} . To achieve unbiased estimation, HSIC criterion requires the diagonal elements of \mathbf{K} and \mathbf{K}_F are set to 0. Based on the HSIC criterion, features can be selected via either backward elimination or forward selection. To do feature selection with the HSIC criterion using a general kernel can be very time consuming due to the complexity of the kernel calculation step in each iteration. Therefore linear kernel is usually used in the HSIC criterion for feature selection. It is shown in [84] that when linear kernel is used for constructing \mathbf{K}_F , using the HSIC criterion to select k features can be formulated as solving the problem:

$$\arg \max_{F_{i_1}, \dots, F_{i_k}} \sum_{j=1}^k \mathbf{f}_{i_j}^\top \mathbf{S}_{HSIC} \mathbf{f}_{i_j}.$$

$$\mathbf{S}_{HSIC} = \frac{1}{n(n-3)} \left[\mathbf{K} + (\mathbf{1}\mathbf{1}^\top - \mathbf{I}) \frac{\mathbf{1}^\top \mathbf{K} \mathbf{1}}{(n-1)(n-2)} \frac{2}{n-2} (\mathbf{K} \mathbf{1}\mathbf{1}^\top - \text{diag}(\mathbf{K} \mathbf{1})) \right] \quad (3.24)$$

It is clear that in the case the HSIC criterion forms a special case of SPEC, which is formally stated in the following theorem:

Theorem 14 *When the linear kernel is applied, using the HSIC to select k features can be achieved by maximizing the following objective function:*

$$\arg \max_{F_{i_1}, \dots, F_{i_k}} \sum_{j=1}^k \hat{\mathbf{f}}_{i_j}^\top \hat{\mathbf{S}} \hat{\mathbf{f}}_{i_j}, \quad \hat{\mathbf{f}} = \mathbf{f}, \quad \hat{\mathbf{S}} = \mathbf{S}_{HSIC}$$

3.7 Discussion

In this chapter, we show that five existing successful feature selection algorithms, including Laplacian Score, Fisher Score, ReliefF, Trace Ratio, and HSIC fit into the framework formulated in Eq. (3.1). In Table 3.1, we summarized the sample similarity matrix and the corresponding feature normalization criteria used in the algorithms. It turns out that although these algorithms are originally designed to achieve different goals, they actually select features via estimating their capability on preserving sample similarity, which is defined in certain ways.. One limitation of all these algorithms is that they evaluate features independently, causing them unable to handle redundant features, which forms a common drawback of these algorithms. To address this limitation, in the next chapter, we propose a multivariate formulation for spectral feature selection. The formulation can effectively remove redundant features and can be efficiently solved. We will also empirically evaluate the performance of the above algorithms in comparison to the proposed multivariate formulation for spectral feature selection.

Table 3.1: The normalized similarity matrix and feature vector used in different algorithms.

Algorithm	Sample Similarity Matrix	Feature Normalization
SPEC - $\hat{\phi}_1(\cdot)$	$\hat{\mathbf{S}} = \mathbf{U}(\mathbf{I} - \gamma(\Sigma))\mathbf{U}^\top$	$\hat{\mathbf{f}} = \frac{\mathbf{D}^{\frac{1}{2}}\mathbf{f}}{\ \mathbf{D}^{\frac{1}{2}}\mathbf{f}\ }$
SPEC - $\hat{\phi}_2(\cdot)$	$\hat{\mathbf{S}} = \mathbf{U}(\mathbf{I} - \gamma(\Sigma))\mathbf{U}^\top$	$\tilde{\mathbf{f}} = \frac{\mathbf{D}^{\frac{1}{2}}\mathbf{f}}{\ \mathbf{D}^{\frac{1}{2}}\mathbf{f}\ }, \hat{\mathbf{f}} = \frac{\tilde{\mathbf{f}} - \tilde{\mathbf{f}}^\top \xi_1 \xi_1}{\sqrt{1 - (\tilde{\mathbf{f}}^\top \xi_1)^2}}$
SPEC - $\hat{\phi}_3(\cdot)$	$\mathbf{S} = \mathbf{U}_k(\gamma(2\mathbf{I}) - \gamma(\Sigma_k))\mathbf{U}_k^\top$	$\hat{\mathbf{f}} = \frac{\mathbf{D}^{\frac{1}{2}}\mathbf{f}}{\ \mathbf{D}^{\frac{1}{2}}\mathbf{f}\ }$
Laplacian Score	$\mathbf{D}^{-\frac{1}{2}}\mathbf{S}\mathbf{D}^{-\frac{1}{2}}$	$\tilde{\mathbf{f}} = \frac{\mathbf{D}^{\frac{1}{2}}\mathbf{f}}{\ \mathbf{D}^{\frac{1}{2}}\mathbf{f}\ }, \hat{\mathbf{f}} = \frac{\tilde{\mathbf{f}} - \tilde{\mathbf{f}}^\top \xi_1 \xi_1}{\sqrt{1 - (\tilde{\mathbf{f}}^\top \xi_1)^2}}$
Fisher Score	\mathbf{S}^{FIS}	$\tilde{\mathbf{f}} = \frac{\mathbf{D}^{\frac{1}{2}}\mathbf{f}}{\ \mathbf{D}^{\frac{1}{2}}\mathbf{f}\ }, \hat{\mathbf{f}} = \frac{\tilde{\mathbf{f}} - \tilde{\mathbf{f}}^\top \xi_1 \xi_1}{\sqrt{1 - (\tilde{\mathbf{f}}^\top \xi_1)^2}}$
ReliefF	\mathbf{S}^{REL}	$\hat{\mathbf{f}} = \frac{\mathbf{D}^{\frac{1}{2}}\mathbf{f}}{\ \mathbf{D}^{\frac{1}{2}}\mathbf{f}\ }$
Trace Ratio Criterion	$\mathbf{L}_b - \lambda^* \mathbf{L}_w$	$\hat{\mathbf{f}} = \mathbf{f}$
HSIC	\mathbf{S}_{HSIC}	$\hat{\mathbf{f}} = \mathbf{f}$

Chapter 4

A MULTIVARIATE FORMULATION FOR SPECTRAL FEATURE SELECTION

Given m features, and a similarity matrix \mathbf{S} of the samples, the idea of spectral feature selection is to select features that align well with the leading eigenvectors of \mathbf{S} . Since the leading eigenvectors of \mathbf{S} contain structure information of sample distribution and group similar samples into compact clusters [95], features aligning better to them will have stronger capability on preserving sample similarity. As shown in the previous chapter, many spectral feature selection algorithms exist, including Laplacian Score [31], Fisher Score [19], Trace Ratio [62], Relief and ReliefF [80], and HSIC [85]. These algorithms demonstrated excellent performance in both supervised and unsupervised learning. However, since they evaluate features individually, they cannot handle redundant features. Redundant features increase dimensionality unnecessarily [39], and worsen learning performance when facing shortage of data. It is also shown empirically that removing redundant features can result in significant performance improvement [29, 16, 23, 100, 2, 18]. Note that none of these redundancy removal algorithms are based on spectral analysis.

In this chapter, we address the limitation of existing spectral feature selection algorithms in handling redundant features, and propose a novel spectral feature selection algorithm of an embedded model, which evaluates the utility of a set of features jointly and can efficiently remove redundant features. The algorithm is derived from a formulation based on multi-output regression [30], and feature selection is achieved by enforcing sparsity through applying $L_{2,1}$ -norm constraint on the solutions [63, 4]. We analyze its capability on redundancy removal and study the properties of its optimal solutions, which paves the way for an efficient path-following solver. By exploiting the necessary and sufficient conditions for the optimal solutions, the solver can automatically adjust its parameters to

generate a solution path for selecting a specific number of features efficiently. We conduct extensive empirical study on the proposed algorithm in both supervised and unsupervised learning to demonstrate that it can select relevant features with low redundancy.

4.1 Spectral Feature Selection with Sparse Multi-output Regression

Let $\mathbf{X} \in \mathbb{R}^{n \times m}$ be the data matrix, where n and m are the number of samples and features, respectively. Given a sample similarity matrix \mathbf{S} specifying the similarity among samples, spectral feature selection aims to select features that preserve the sample similarity specified by \mathbf{S} . Given a feature F , as shown in Chapter 3, different spectral feature selection criteria can be formulated in a common form:

$$\varphi(F, \mathbf{S}) = \hat{\mathbf{f}}^\top \hat{\mathbf{S}} \hat{\mathbf{f}} = \sum_{i=1}^n \hat{\lambda}_i (\hat{\mathbf{f}}^\top \hat{\xi}_i)^2 = \sum_{i=1}^n \left(\hat{\lambda}_i^{\frac{1}{2}} \hat{\mathbf{f}}^\top \hat{\xi}_i \right)^2. \quad (4.1)$$

In the equation, $\hat{\mathbf{f}}$ and $\hat{\mathbf{S}}$ are the normalized feature vector \mathbf{f} and similarity matrix \mathbf{S} obtained by applying certain normalization rules. $\hat{\lambda}_i$ and $\hat{\xi}_i$ are the i -th eigenvalue and eigenvector of the $\hat{\mathbf{S}}$, respectively. Different spectral feature selection algorithms adopt different ways to define \mathbf{S} and use different rules to normalize \mathbf{f} and \mathbf{S} to achieve the certain effect, such as noise removal. Eq. (4.1) shows that existing spectral feature selection algorithms evaluate features individually. Therefore they cannot identify redundant features, which is a common drawback of these algorithms and needs to be addressed.

To identify redundant features, features must be evaluated jointly. To this end, given $\mathbf{y}_i = \lambda_i^{1/2} \xi_i$, where λ_i and ξ_i are the i -th eigenvalue and eigenvector of \mathbf{S} , instead of looking for one feature which closely aligns to \mathbf{y}_i , as formulated in Eq. (4.1), we propose to find a set of l features, such that their linear span is close to \mathbf{y}_i . The idea can be formulated as:

$$\arg \min_{\mathbb{A}, \mathbf{w}_{i,\mathbb{A}}} \|\mathbf{y}_i - \mathbf{X}_{\mathbb{A}} \mathbf{w}_{i,\mathbb{A}}\|_2^2.$$

In the equation, $\mathbb{A} = \{i_1, \dots, i_l\} \subseteq \{1, \dots, m\}$, $\mathbf{X}_{\mathbb{A}} = (\mathbf{f}_{i_1}, \dots, \mathbf{f}_{i_l})$ and $\mathbf{w}_i \in \mathbb{R}^{l \times 1}$. Note, to facilitate the subsequent formulations, in the above equation, we use L_2 norm on the

difference of two vectors to measure the closeness among vectors. When all λ_i and ξ_i are considered, their joint optimization can be formulated as:

$$\arg \min_{\mathbb{A}, \mathbf{w}_{i,\mathbb{A}}} \sum_{i=1}^n \|\mathbf{y}_i - \mathbf{X}_{\mathbb{A}} \mathbf{w}_{i,\mathbb{A}}\|_2^2 = \|\mathbf{Y} - \mathbf{X}_{\mathbb{A}} \mathbf{W}_{\mathbb{A}}\|_F^2. \quad (4.2)$$

In the equation, $\mathbf{Y}=(\mathbf{y}_1, \dots, \mathbf{y}_n)$, $\mathbf{W}_{\mathbb{A}}=(\mathbf{w}_{1,\mathbb{A}}, \dots, \mathbf{w}_{n,\mathbb{A}})$. Assume $\mathbf{S} = \mathbf{U}\Sigma\mathbf{U}^\top$ is the SVD of \mathbf{S} , we have $\mathbf{Y}=\mathbf{U}\Sigma^{1/2}$. Note, when \mathbb{A} contains only one feature, the formulation reduces to searching for features that maximize the Eq. (4.1).

Given \mathbf{Y} and $\mathbf{X}_{\mathbb{A}}$, $\mathbf{W}_{\mathbb{A}}$ can be obtained in a closed form. However, feature selection needs to find the optimal \mathbb{A} , which is a combinatorial problem of NP-hard. To efficiently solve the problem, we propose the following formulation:

$$\begin{aligned} & \arg \min_{\mathbf{W}, c} \|\mathbf{Y} - \mathbf{X}\mathbf{W}\|_F^2 & (4.3) \\ \text{s.t.} & \quad \|\mathbf{W}\|_{2,1} \leq t \\ & \quad \mathbb{A} = \{i : \|\mathbf{w}^i\|_2 > 0\}, \text{Card}(\mathbb{A}) = l \end{aligned}$$

Here \mathbf{w}^i denotes the i th row of \mathbf{W} , and $\|\mathbf{W}\|_{2,1}$ is the $L_{2,1}$ -norm which is defined as:

$$\|\mathbf{W}\|_{2,1} = \sum_{i=1}^m \|\mathbf{w}^i\|_2 \quad (4.4)$$

When applied in regression, the $L_{2,1}$ -norm constraint is equivalent to applying Laplace prior [76] on \mathbf{w}^i , which tends to force many rows in \mathbf{W} to be $\mathbf{0}^\top$, resulting sparse solution. The advantages of the formulation presented in Eq. (4.3) are three folds.

First, it can find a set of features jointly preserving the sample similarity specified by \mathbf{S} .

Theorem 15 Let $\mathbf{S} = \mathbf{U}\Sigma\mathbf{U}^\top$, $\mathbf{Y} = \mathbf{U}\Sigma^{1/2}$ and $\Omega = \mathbf{Y} - \mathbf{X}\mathbf{W}$. We have:

$$\left\| \mathbf{X}\mathbf{W}\mathbf{W}^\top\mathbf{X}^\top - \mathbf{S} \right\|_F \leq 2(\|\mathbf{Y}\|_F + \|\Omega\|_F) \|\Omega\|_F$$

Proof: Since $\mathbf{S} = \mathbf{Y}\mathbf{Y}^\top$, and $\|\mathbf{A} + \mathbf{B}\|_F \leq \|\mathbf{A}\|_F + \|\mathbf{B}\|_F$, we have:

$$\begin{aligned}
\|\mathbf{X}\mathbf{W}\mathbf{W}^\top\mathbf{X}^\top - \mathbf{Y}\mathbf{Y}^\top\|_F &= \|\mathbf{\Omega}\mathbf{Y}^\top + \mathbf{Y}\mathbf{\Omega}^\top + \mathbf{\Omega}\mathbf{\Omega}^\top\|_F \\
&\leq \|\mathbf{\Omega}\mathbf{Y}^\top\|_F + \|\mathbf{Y}\mathbf{\Omega}^\top\|_F + \|\mathbf{\Omega}\mathbf{\Omega}^\top\|_F \\
&\leq 2\|\mathbf{Y}\|_F\|\mathbf{\Omega}\|_F + \|\mathbf{\Omega}\|_F^2 \\
&= (2\|\mathbf{Y}\|_F + \|\mathbf{\Omega}\|_F)\|\mathbf{\Omega}\|_F
\end{aligned}$$

In the derivation, we use the fact: $\|\mathbf{AB}\|_F \leq \|\mathbf{A}\|_F\|\mathbf{B}\|_F$. ■

In the theorem, $\mathbf{X}\mathbf{W}$ is a new representation of samples obtained by linearly combining the selected features¹. And $\mathbf{X}\mathbf{W}\mathbf{W}^\top\mathbf{X}^\top$ returns the pairwise similarity among samples measured by their inner product under the new representation. The theorem shows that by minimizing $\|\mathbf{\Omega}\|_F$, we also minimize $\|\mathbf{X}\mathbf{W}\mathbf{W}^\top\mathbf{X}^\top - \mathbf{S}\|_F$, which ensures the selected features can jointly preserve the sample similarity specified by \mathbf{S} .

Second, by jointly evaluating a set of features, it tends to select non-redundant features. Assume two features \mathbf{f}_p and \mathbf{f}_q satisfy the following conditions: (1) they are equally correlated to \mathbf{Y} , i.e. $\mathbf{f}_p^\top\mathbf{Y} = \mathbf{f}_q^\top\mathbf{Y}$; (2) \mathbf{f}_q is highly correlated to \mathbf{f}_d , i.e. $\mathbf{f}_q^\top\mathbf{f}_d \rightarrow 1$. And \mathbf{f}_q is less correlated to \mathbf{f}_d , i.e. $\mathbf{f}_p^\top\mathbf{f}_d > \mathbf{f}_q^\top\mathbf{f}_d$. Without loss of generality, we assume both \mathbf{f}_p and \mathbf{f}_q are positively correlated to \mathbf{f}_d ; (3) they are equally correlated to other features, i.e. $\mathbf{f}_p^\top\mathbf{f}_i = \mathbf{f}_q^\top\mathbf{f}_i$, $\forall i \in \{1, \dots, m\}$, $i \neq d$. Based on the assumptions, we have the following theorem:

Theorem 16 *Given the above assumptions, assume \mathbf{f}_d is selected by an optimal solution of Eq. (4.3), then \mathbf{f}_q has higher priority than \mathbf{f}_p to be selected in the optimal solution.*

Proof: Let $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_k)$ be the $n \times k$ response matrix and \mathbf{W} be the $m \times k$ weight matrix. The i -th row and j -th column of \mathbf{W} are denoted by $\mathbf{W}_{i\cdot}$ and $\mathbf{W}_{\cdot j}$ respectively. Recall that

¹Note that although $\mathbf{W} \in \mathbb{R}^{m \times k}$, many of its rows are $\mathbf{0}^\top$. Therefore, the representation is generated by using only a small set of selected features

$\|\cdot\|_F$ is the Frobenius norm and $\|\mathbf{W}\|_{2,1} = \sum_{i=1}^m \|\mathbf{W}_i\|_2$. Let $\bar{\mathbf{W}}$ be the current solution in which two strongly correlated feature \mathbf{f}_d and \mathbf{f}_p are selected, i.e. $\|\bar{\mathbf{W}}_{d:}\|_F > 0, \|\bar{\mathbf{W}}_{p:}\|_F > 0$. By using the technique developed in [70], it can be shown that in the optimal solution of Eq. (4.3), $\langle \bar{\mathbf{W}}_{d:}, \bar{\mathbf{W}}_{p:} \rangle > 0$ when $\mathbf{f}_d^\top \mathbf{f}_p \rightarrow 1$. Assume the three conditions specified before the theorem hold. Below, we show that as long as \mathbf{f}_q has a sufficiently small $\rho_{dq} = \mathbf{f}_d^\top \mathbf{f}_q$, selecting \mathbf{f}_q rather than \mathbf{f}_p can always decrease the objective function. To this end, we define another weight matrix $\tilde{\mathbf{W}}$ as: $\tilde{\mathbf{W}}_i = \bar{\mathbf{W}}_i, \forall i \neq p, q$ and $\tilde{\mathbf{W}}_q = \bar{\mathbf{W}}_{p:}, \tilde{\mathbf{W}}_p = \mathbf{0}$. Note that (1) $\|\tilde{\mathbf{W}}\|_{2,1} = \|\bar{\mathbf{W}}\|_{2,1}$; and (2) $\tilde{\mathbf{W}}$ and $\bar{\mathbf{W}}$ have no difference except the p -th and q -th rows. Since $\|\mathbf{Y} - \mathbf{XW}\|_F^2 = \sum_{j=1}^l \|\mathbf{y}_j - \mathbf{XW}_{:j}\|_2^2$, we can show that:

$$\|\mathbf{Y} - \mathbf{X}\bar{\mathbf{W}}\|_F^2 - \|\mathbf{Y} - \mathbf{X}\tilde{\mathbf{W}}\|_F^2 = 2\bar{\mathbf{W}}_{p:} \mathbf{Y}^\top (\mathbf{f}_q - \mathbf{f}_p) + 2\langle \bar{\mathbf{W}}_{d:}, \bar{\mathbf{W}}_{p:} \rangle (\rho_{dp} - \rho_{dq}),$$

where, $\rho_{ij} = \mathbf{f}_i^\top \mathbf{f}_j$. In the derivation, we use the fact that:

$$\sum_{j=1}^l \bar{\mathbf{W}}_{pj} \mathbf{y}_j^\top = \bar{\mathbf{W}}_{p:} \mathbf{Y}^\top, \sum_{j=1}^l \bar{\mathbf{W}}_{dj} \bar{\mathbf{W}}_{pj} = \langle \bar{\mathbf{W}}_{d:}, \bar{\mathbf{W}}_{p:} \rangle.$$

Based on the equation, we have the following inequality:

$$\|\mathbf{Y} - \mathbf{X}\bar{\mathbf{W}}\|_F^2 - \|\mathbf{Y} - \mathbf{X}\tilde{\mathbf{W}}\|_F^2 > 0 \Leftrightarrow (\rho_{dp} - \rho_{dq}) > (\langle \bar{\mathbf{W}}_{d:}, \bar{\mathbf{W}}_{p:} \rangle)^{-1} \bar{\mathbf{W}}_{p:} \mathbf{Y}^\top (\mathbf{f}_p - \mathbf{f}_q).$$

Since $\|\mathbf{Y}^\top \mathbf{f}_p - \mathbf{Y}^\top \mathbf{f}_q\|_2 = 0$, according to the assumption. We have that as far as $\rho_{dq} < \rho_{dp}$, selecting \mathbf{f}_q rather than \mathbf{f}_p can always decrease the objective function. ■

The theorem shows that the formulation in Eq. (4.3) tends to select features that are less correlated, which ensures the selection of non-redundant features.

Third, it is tractable. Given a value for t , the problem:

$$\arg \min_{\mathbf{W}, c} \|\mathbf{Y} - \mathbf{XW}\|_F^2 \quad s.t. \quad \|\mathbf{W}\|_{2,1} \leq t \quad (4.5)$$

can be solved by applying a general solver [63, 4, 53]. And given l , a proper c value, which results in the selection of about l features, can be found by applying either a grid search or a binary search based on the observation that, a smaller t value usually results in selecting fewer features. However, for a given l , this approach may require to run a solver many times for searching the proper t value, which is computationally inefficient.

4.2 MRSE, An Efficient Solver

We propose an efficient path-following solver for solving the problem specified in Eq. (4.3). It can automatically detect the points when new features enter its “active set”, and update its parameters accordingly. It can efficiently generate a solution path to select the specified number of features. We start by deriving the necessary and sufficient conditions for a feature to be selected in an optimal solution of Eq. (4.5).

The Lagrangian for Eq. (4.5) has the following form:

$$\mathcal{L}(\mathbf{W}, \lambda) = \|\mathbf{Y} - \mathbf{XW}\|_F^2 - \lambda \left(t - \|\mathbf{W}\|_{2,1} \right). \quad (4.6)$$

$\mathcal{L}(\mathbf{W}, \lambda)$ is convex. According to the convex optimization theorem [7], \mathbf{W}_* minimizes $\mathcal{L}(\mathbf{W}, \lambda)$ if and only if $\mathbf{0} \in \partial_{\mathbf{w}^i} \mathcal{L}(\mathbf{W}, \lambda)|_{\mathbf{W}=\mathbf{W}_*}$, $i = 1, \dots, m$. Here, $\partial_{\mathbf{w}^i} \mathcal{L}(\mathbf{W}, \lambda)$ is the subdifferential of $\mathcal{L}(\mathbf{W}, \lambda)$ corresponding to \mathbf{w}^i , and has the following form:

$$\begin{aligned} \partial_{\mathbf{w}^i} \mathcal{L}(\mathbf{W}, \lambda) &= \mathbf{f}_i^\top (\mathbf{Y} - \mathbf{XW}) + \lambda \mathbf{v}_i \\ \mathbf{v}_i &= \frac{\mathbf{w}^i}{\|\mathbf{w}^i\|}, \text{ if } \mathbf{w}^i \neq \mathbf{0} \\ \mathbf{v}_i &\in \left\{ \mathbf{u} \mid \mathbf{u} \in \mathbb{R}^{1 \times k}, \|\mathbf{u}\|_2 \leq 1 \right\}, \text{ if } \mathbf{w}^i = \mathbf{0} \end{aligned} \quad (4.7)$$

Therefore, \mathbf{W}_* is an optimal solution if and only if:

$$-\lambda \mathbf{v}_i = \mathbf{f}_i^\top (\mathbf{Y} - \mathbf{XW})|_{\mathbf{W}=\mathbf{W}_*}, \forall i \in \{1, \dots, m\} \quad (4.8)$$

Base on this observation, we give the necessary conditions, and the necessary and sufficient conditions for \mathbf{W} to be optimal with the following two propositions.

Proposition 2 Assume \mathbf{w}^i is the i -th row of \mathbf{W} , the necessary conditions for \mathbf{W} to be optimal are: $\forall i \in \{1, \dots, m\}$:

$$\begin{aligned}\mathbf{w}^i \neq \mathbf{0} &\Rightarrow \|\mathbf{f}^\top (\mathbf{Y} - \mathbf{XW})\|_2 = \lambda \\ \mathbf{w}^i = \mathbf{0} &\Rightarrow \|\mathbf{f}^\top (\mathbf{Y} - \mathbf{XW})\|_2 \leq \lambda\end{aligned}\quad (4.9)$$

Proposition 3 Assume \mathbf{w}^i is the i -th row of \mathbf{W} , the necessary and sufficient conditions for \mathbf{W} to be optimal are: $\forall i$

$$\begin{aligned}\mathbf{w}^i \neq \mathbf{0} &\Rightarrow \mathbf{f}^\top (\mathbf{Y} - \mathbf{XW}) = -\lambda \frac{\mathbf{w}^i}{\|\mathbf{w}^i\|_2} \\ \mathbf{w}^i = \mathbf{0} &\Rightarrow \|\mathbf{f}^\top (\mathbf{Y} - \mathbf{XW})\|_2 \leq \lambda\end{aligned}\quad (4.10)$$

Based on the two propositions, we propose an efficient solver for Eq. (4.3), and its pseudo code can be found in Algorithm 2. In the algorithm, \mathbb{A}_i is the ‘‘active set’’ in the i -th run, which contains the features selected in that run. Algorithm 2 contains two major steps. (1) Lines 4-10, the algorithm determines the direction for updating $\mathbf{W}^{[i]}$ (Line 4), and the step size (Lines 5-8), by which, it updates the active set and the λ (Line 10). (2) Lines 11-18, the algorithm finds an optimal solution corresponding to the λ obtained in step 1. Given λ , it first solves an $L_{2,1}$ -norm regularized regression problem using a general solver based on the Nesterov’s method [53] (Line 11). Note, this problem is of small scale, since it is only based on the features in the current active set, but not the whole set. $\hat{\mathbf{W}}$ is used as a starting point to ensure fast convergence of the Nesterov solver. It then checks whether the obtained solution is also optimal on the whole data (Line 13). If it is true, the algorithm records the current optimal solution and goes to Line 4 to start next run (Line 14). Otherwise, it adjusts the active set and goes back to Line 11 (Line 17).

Theorem 17 (1) Given $\mathbf{W}^{[i-1]}$ is the current optimal solution, the $\hat{\mathbf{W}}$ generated in step 1 (Line 9) satisfies the necessary condition for an optimal solution specified in Proposition 2. (2) And the $\tilde{\mathbf{W}}$ in Line 14 of step 2 is an optimal solution corresponding to the current λ .

Proof: To prove the first point, it is sufficient to show that $\mathbf{f}_i^\top \mathbf{X}_{\mathbb{A}_i} \left(\mathbf{X}_{\mathbb{A}_i}^\top \mathbf{X}_{\mathbb{A}_i} \right)^{-1} \mathbf{X}_{\mathbb{A}_i}^\top = \mathbf{f}_i^\top$, $\forall i \in \mathbb{A}_i$. And the second point of the theory can be simply verified by applying the necessary and sufficient conditions for the optimal solutions developed in Proposition 3. ■

Algorithm 2: MRSF

Input: \mathbf{X} , \mathbf{Y} , k
Output: \mathbf{W}

- 1 $\mathbf{W}^{[0]} = \mathbf{0}$, $i = 1$ and $\mathbf{R}^{[0]} = \mathbf{Y}$;
- 2 Compute the initial “active set” $\mathbb{A}_1 = \arg \max_j \|\mathbf{f}_j^\top \mathbf{R}^{[0]}\|_2^2$;
- 3 **while** $i \leq k$ **do**
- 4 Compute the walking direction $\gamma_{\mathbb{A}_i}$: $\gamma_{\mathbb{A}_i} = \left(\mathbf{X}_{\mathbb{A}_i}^\top \mathbf{X}_{\mathbb{A}_i} \right)^{-1} \mathbf{X}_{\mathbb{A}_i}^\top \mathbf{R}^{[i-1]}$;
- 5 **for each** $j \notin \mathbb{A}_i$ **and an arbitrary** $t \in \mathbb{A}_i$ **do**
- 6 Compute the step size α_j in direction $\gamma_{\mathbb{A}_i}$ for \mathbf{f}_j to enter \mathbb{A}_i .
 $\|\mathbf{f}_j^\top \left(\mathbf{R}^{[i-1]} - \alpha_j \mathbf{X}_{\mathbb{A}_i} \gamma_{\mathbb{A}_i} \right)\|_2 = (1 - \alpha_j) \|\mathbf{f}_t^\top \mathbf{R}^{[i-1]}\|_2$;
- 7 **end**
- 8 $j^* = \arg \min_{j \notin \mathbb{A}_i} \alpha_j$;
- 9 $\hat{\mathbf{W}} = \left(\left(\mathbf{W}^{[i-1]} + \alpha_{j^*} \gamma_{\mathbb{A}_i} \right)^\top, \mathbf{0} \right)^\top$;
- 10 $\hat{\mathbb{A}} = \mathbb{A}_i \cup \{j^*\}$, $\lambda = (1 - \alpha_{j^*}) \|\mathbf{f}_t^\top \mathbf{R}^{[i-1]}\|_2$;
- 11 Solve the smaller optimization problem $\min_{\tilde{\mathbf{W}}} \|\mathbf{Y} - \mathbf{X}_{\hat{\mathbb{A}}} \tilde{\mathbf{W}}\|_F^2 + \lambda \|\tilde{\mathbf{W}}\|_{2,1}$ using Nesterov’s method, with $\hat{\mathbf{W}}$ as the starting point;
- 12 $\tilde{\mathbf{R}} = \mathbf{Y} - \mathbf{X}_{\hat{\mathbb{A}}} \tilde{\mathbf{W}}$;
- 13 **if** $\forall i \notin \hat{\mathbb{A}}, \|\mathbf{f}_i^\top \tilde{\mathbf{R}}\|_2 \leq \lambda$ **then**
- 14 $i = i + 1$, $\mathbb{A}_i = \hat{\mathbb{A}}$, $\mathbf{W}^{[i-1]} = \tilde{\mathbf{W}}$, $\mathbf{R}^{[i-1]} = \tilde{\mathbf{R}}$;
- 15 **else**
- 16 $\hat{\mathbb{A}} = \{i : \|\tilde{\mathbf{w}}^i\| \neq 0\} \cup \{\arg \max_j \|\mathbf{f}_j^\top \tilde{\mathbf{R}}\|_2\}$;
- 17 Remove $\tilde{\mathbf{w}}^i$ from $\tilde{\mathbf{W}}$, if $\|\tilde{\mathbf{w}}^i\| = 0$, $\hat{\mathbf{W}} = (\tilde{\mathbf{W}}^\top, \mathbf{0})^\top$, **Goto** line 11;
- 18 **end**
- 19 **end**
- 20 Extend $\mathbf{W}^{[k]}$ to \mathbf{W} by adding empty rows to $\mathbf{W}^{[k]}$;
- 21 **return** $\mathbf{W}^{[k]}$;

Algorithm 2 is very efficient. In each run, in step 1 it increases the size of its active set and decreases the λ accordingly. At the same time, it generates a tentative solution, which satisfies the necessary conditions for the optimal solution. And in step 2, it generates an optimal solution corresponding to λ for the whole data by working on features in the active set only. Since the tentative solution is usually very close to a true optimal, step 2 often converges in just a few iterations. It can be shown that when n features are selected, Algorithm 2 has a time complexity of $O(n^3k + mn^2)$.

4.3 Experimental Study

We now empirically evaluate the performance of the proposed multivariate formulation for spectral feature selection in both supervised and unsupervised learning context. We name it MRSF, since it is proposed to Minimize the feature Redundancy for Spectral Feature selection. In the experiments, we choose eight representative feature selection algorithms for comparison. For supervised learning, six feature selection algorithms are chosen as baselines: ReliefF, Fisher score, Trace Ratio Criterion, HSIC, mRMR [16] and AROM-SVM [96]. The first four are existing spectral feature selection algorithms. And the last two are the-state-of-the-art feature selection algorithms for removing redundant features. For unsupervised learning, four algorithms are used for comparison: Laplacian Score, SPEC, Trace Ratio Criterion, and HSIC. They are all spectral feature selection algorithms. For MRSF, in supervised learning, \mathbf{S} is calculated by Eq. (3.13); and in unsupervised learning, \mathbf{S} is calculated by the Gaussian RBF kernel function. Six high dimensional data sets are used in the experiment. They are four image data: AR10P², PIE10P³, PIX10P⁴, and ORL10P⁵. Two Microarray data: TOX and CLL-SUB from the Gene Expression Omnibus

²http://rv11.ecn.purdue.edu/~leix/aleix_face.DB.html. Images are subsampled down to size of 60×40.

³<http://peipa.essex.ac.uk/ipa/pix/faces/manchester/>. Images are subsampled down to the size of 60×40.

⁴http://www.ri.cmu.edu/projects/project_418.html. Images are subsampled down to the size of 100×100.

⁵<http://www.uk.research.att.com/facedatabase.html>. Images are subsampled down to size of 100×100.

Table 4.1: Summary of the benchmark data sets

Data Set	# Features	# Instances	# Classes
AR10P	2400	130	10
PIE10P	2400	210	10
PIX10P	10000	100	10
ORL10P	10000	100	10
TOX	5748	171	4
CLL-SUB	11340	111	3

(GEO) gene expression repository⁶ with retrieval IDs: GDS1454 and GDS968. Detailed information of the benchmark data sets is listed in Table 5.3.

Assume \mathbf{F} is the set of selected features, and $\mathbf{X}_{\mathbf{F}}$ only containing features in \mathbf{F} . In the supervised learning context, algorithms are compared on (1) classification accuracy and (2) redundancy rate, the redundancy rate is measured by:

$$\text{RED}(\mathbf{F}) = \frac{1}{m(m-1)} \sum_{f_i, f_j \in \mathbf{F}, i > j} \rho_{i,j},$$

where, $\rho_{i,j}$ returns the correlation between the i -th and the j -th features. A large value of $\text{RED}(\mathbf{F})$ indicates that many selected features are strongly correlated and thus redundancy is expected to exist in \mathbf{F} . For unsupervised case, two measurements are used: (1) the redundancy rate as defined above; and (2) the Jaccard score computed by:

$$\text{JAC}(\mathbf{S}_{\mathbf{F}}, \mathbf{S}, k) = \frac{1}{n} \sum_{i=1}^n \frac{NB(i, k, \mathbf{S}_{\mathbf{F}}) \cap NB(i, k, \mathbf{S})}{NB(i, k, \mathbf{S}_{\mathbf{F}}) \cup NB(i, k, \mathbf{S})},$$

where, $\mathbf{S}_{\mathbf{F}} = \mathbf{X}_{\mathbf{F}} \mathbf{X}_{\mathbf{F}}^{\top}$ is the similarity matrix computed from the selected features using inner product; and $NB(i, k, \mathbf{S})$ returns the k nearest neighbors of the i -th sample according to \mathbf{S} . The Jaccard score measures the averaged overlapping of the neighborhoods specified by $\mathbf{S}_{\mathbf{F}}$ and \mathbf{S} . A high Jaccard score indicates that sample similarity are well preserved.

For each data set, we randomly sample 50% samples as the training data and the remaining are used for test. The process is repeated for 20 times and results in 20 different

⁶<http://www.ncbi.nlm.nih.gov/geo/>

partitions. Different algorithms are evaluated on each partition. The results are recorded and averaged to generate the final results. Linear SVM is used for classification. The parameters in feature selection algorithms and SVM are tuned via cross-validation if necessary. Student's t-test is used to evaluate the statistical significance with $p\text{Val} < 0.05$.

4.3.1 Study of Supervised Cases

Accuracy: the classification accuracy results are shown in Figure 5.5 and Table 5.4. Figure 5.5 contains the plots of the accuracy achieved by the SVM classifier when uses the top 10, 20, ..., 200 features selected by each algorithm. Table 5.4 shows the “aggregated accuracy” of different algorithms on each data set. The aggregated accuracy is obtained by averaging the averaged accuracy achieved by SVM using the top 10, 20, ..., 200 features selected by each algorithm. The value in the parentheses is the p -Val. In Figure 5.5 and Table 5.4, we can observe that MRSF produces superior classification performance comparing to the baseline algorithms. The averaged value for aggregated accuracy achieved by the baseline algorithms is 0.78, which is 11% lower than that achieved by MRSF.

Redundancy rate: Table 4.3 presents the averaged redundancy rates of the top n features selected by different algorithms, where n is the number of samples. We choose n , since when the number of selected features is larger than n , any feature can be expressed by a linear combination of the remaining ones, which will introduces unnecessary redundancy in evaluation. In the table, the boldfaced values are the lowest redundancy rates or the ones without significant difference to the lowest. The results show that MRSF attains very low redundancy, which suggests that the redundancy removal mechanism in MRSF is effective.

4.3.2 Study of Unsupervised Cases

Jaccard Score: Tables 4.4 present the averaged Jaccard score achieved by different algorithms. Results show that MRSF achieves significant better results on all data sets compar-

ing to the baseline algorithms, which demonstrates its strong capability on selecting good features for preserving sample similarity specified in the given similarity matrix.

Redundancy rate: Table 4.5 shows the averaged redundancy rates achieved with the top n features selected by different algorithms. The results show that the features selected by MRSF contains much less redundancy comparing with the baseline algorithms. This is expected, since the latter cannot remove redundant features in feature selection.

4.3.3 Study of Efficiency

Figure 4.1 presents the running time of MRSF and a solver for Eq. (4.5) proposed in [53]. Since that solver is based the Nesterov's method, we call it Nesterov in this paper. As shown in [53], Nesterov is one of the fastest solvers for solving Eq. (4.5). The running time is obtained in the following way. We first run MRSF to select 100 features on each data set and record the obtained $t = \|W\|_{2,1}$ and the time it used. We then run Nesterov on each data using the c we obtained from MRSF and record its running time. The precision of the Nesterov is set to 10^{-6} . Note that Nesterov is also used in Line 11 of MRSF, where its precision is set to 10^{-6} , too. The results show that on the six data sets, MRSF achieved an average running time of 23.33s, which compares to the 37.46s of Nesterov. Note that if the grid search or binary search is applied to determine c , Nesterov may have a running time which is much longer, even with the warm start strategy [53]. The results demonstrate the high efficiency of the proposed MRSF algorithm.

The experiment results from both supervised and unsupervised learning cases show consistently that MRSF can very efficiently select features containing less redundancy and producing excellent learning performance.

4.4 discussion

In this chapter, we propose a novel spectral feature selection algorithm based on sparse multi-output regression with $L_{2,1}$ -norm constraint. We study the properties of its solutions,

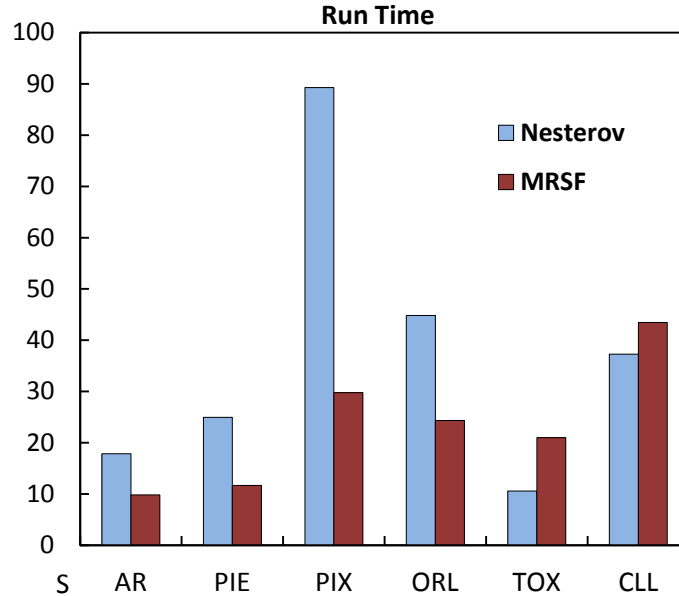


Figure 4.1: The Running time of MRSF and the Nesterov method on each data set.

and design an efficient solver following our formulation. The algorithm improves existing spectral feature selection algorithms by overcoming a common drawback in handling feature redundancy. As illustrated by extensive experimental study, the proposed algorithm can effectively remove redundant features and achieve superior performance in both supervised and unsupervised learning. In our study, we find that our formulation for spectral feature selection can be linked to a wide range of learning models, such as principle component analysis (PCA), support vector machine (SVM), and linear discriminant analysis (LDA) through their least square formulations [91, 89].

Given a covariance matrix, principle component analysis (PCA) seeks factors to explain a maximum amount of variance in the given data. Below we show the connection between MRSF and PCA. Assume $m \geq n$ and $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ is the SVD of \mathbf{X} , and $\mathbf{U} = \{\mathbf{u}_1, \dots, \mathbf{u}_n\}$, $\mathbf{V} = \{\mathbf{v}_1, \dots, \mathbf{v}_n\}$, and $\mathbf{\Sigma} = \text{diag}\{\lambda_1, \dots, \lambda_n\}$. In [116], it is shown that the principle compo-

nents generated by PCA can be equally obtained by solving the following problem:

$$\begin{aligned}\mathbf{w}_i &= \arg \max_{\mathbf{w}_i} \|\mathbf{y}_i - \mathbf{X}\mathbf{w}_i\|_2 \\ \mathbf{y}_i &= \lambda_i \mathbf{v}_i.\end{aligned}\quad (4.11)$$

Based on this observation, in [116], the authors proposed to obtain sparse solutions for PCA to achieve sparse PCA (SPCA) by solving the following 1-norm regularization problem.

$$\begin{aligned}\mathbf{w}_i &= \arg \max_{\mathbf{w}_i} \|\mathbf{y}_i - \mathbf{X}\mathbf{w}_i\|_2 \\ \|\mathbf{w}_i\|_1 &\leq t, \\ \mathbf{y}_i &= \lambda_i \mathbf{v}_i.\end{aligned}\quad (4.12)$$

The objective of Eq. 4.12 can be written in the form of:

$$\begin{aligned}\mathbf{W} &= \arg \max_{\mathbf{W}} \|\mathbf{Y} - \mathbf{X}\mathbf{W}\|_F \\ \mathbf{W} &= \{\mathbf{w}_1, \dots, \mathbf{w}_k\}, \\ \|\mathbf{w}_i\|_1 &\leq t, \\ \mathbf{Y} &= \mathbf{V}_k \Sigma_k,\end{aligned}\quad (4.13)$$

where $\mathbf{V}_k = \{\mathbf{v}_1, \dots, \mathbf{v}_k\}$, and $\Sigma_k = \text{diag}\{\lambda_1, \dots, \lambda_k\}$. As we can see that $\mathbf{Y}\mathbf{Y}^\top = \mathbf{V}_k \Sigma_k^2 \mathbf{V}_k^\top \simeq \mathbf{X}\mathbf{X}^\top$. Here, \simeq is used to denote an approximation, and we have:

$$\|\mathbf{X}\mathbf{X}^\top - \mathbf{Y}\mathbf{Y}^\top\|_F = \sum_{i=k+1}^n \lambda_i.$$

When k is large enough $\mathbf{X}\mathbf{X}^\top$ can be very close to $\mathbf{Y}\mathbf{Y}^\top$, and only a small amount of variance is lost in the approximation. Comparing Eq. (4.11) and Eq. (4.13) to Eq. (4.3), it is clear that PCA eccentrically projects samples to a lower dimensionality space, where the similarity among samples measured by their inner product in the original space are best preserved. Therefore SPCA formulation specified in Eq. (4.12) can be regarded as a

special case of MRSF when linear kernel is used to define the similarity among samples. As compared with SPCA, MRSF is derived from the idea of similarity preserving but not variance preserving. This eccentric difference enables MRSF to work with sample similarities defined by various metric, besides linear kernel. Also, the usage of $L_{2,1}$ norm in MRSF make it more suitable for feature selection. The L_1 norm used in Eq. 4.12 ensures sparse pattern on each coordinate. However, for generating different coordinates, different features are used, when considered all coordinates together, still, many features will be used in the obtained \mathbf{W} . In comparing to the L_1 norm, the $L_{2,1}$ norm is able to ensure that only a small set of the features are used to define all the coordinates.

The least square Linear Discriminant Analysis (LSLDA) and the least square Support Vector Machine (LSSVM) are proposed in [99] and [91], respectively. The two learning models can be formulated in a common way:

$$\min_{\mathbf{W}} \|\mathbf{XW} - \mathbf{Y}\|_F^2, \quad \mathbf{W} \in \mathbb{R}^{m \times k}. \quad (4.14)$$

In Eq. (4.14), $\mathbf{Y} \in \mathbb{R}^{n \times k}$ is the label matrix. Each row of \mathbf{Y} corresponds to an instance and each column corresponds to a class. For LSLDA, the \mathbf{Y} is defined as:

$$\mathbf{Y}_{i,j}^{LDA} = \begin{cases} \sqrt{\frac{n}{n_j}} - \sqrt{\frac{n_j}{n}} & y_i = j \\ -\sqrt{\frac{n_j}{n}} & \text{otherwise.} \end{cases} \quad (4.15)$$

For LSSVM, the \mathbf{Y} is defined as:

$$\mathbf{Y}_{i,j}^{SVM} = \begin{cases} 1 & y_i = j \\ -1 & \text{otherwise.} \end{cases} \quad (4.16)$$

Comparing Eq. (4.14) to Eq. (4.3), it shows that if we plug the \mathbf{Y} defined in Eq. (4.15) and (4.16) in Eq. (4.3), MRSF will select features using the criteria specified in LSLDA and LSSVM, and generate sparse solution for the two least square models. Also we can show that let \mathbf{S} be the matrix defined as in Eq. (3.13), \mathbf{K}_{LSLDA} can also be formulated as:

$$\mathbf{K}_{LSLDA} = (\mathbf{Y}^{LDA})(\mathbf{Y}^{LDA})^\top = n\mathbf{K}^{FIS} - \mathbf{1}\mathbf{1}^\top. \quad (4.17)$$

Since the matrix $\mathbf{1}\mathbf{1}^\top$ and n are constant, we can see that Fisher score and LSLDA essentially specify the same simple similarity. Unlike Fisher score, LSLDA is a supervised feature extraction algorithm, which generates new features by combining original features.

The above analysis connects MRSF to some well known learning models, including principle component analysis (PCA), linear discriminant analysis (LDA) and support vector machine (SVM) through their least square formulations. The connections provide insights on these models as well as MRSF, and allow us to use MRSF to generate sparse (or more sparser) solutions for them. Note, MRSF is for feature selection, while PCA and LDA are for feature extraction, therefore the analysis essentially build a bridge that connects feature selection and feature extraction. This forms an interesting contribution, which may be studied in greater detail as one line the future research work.

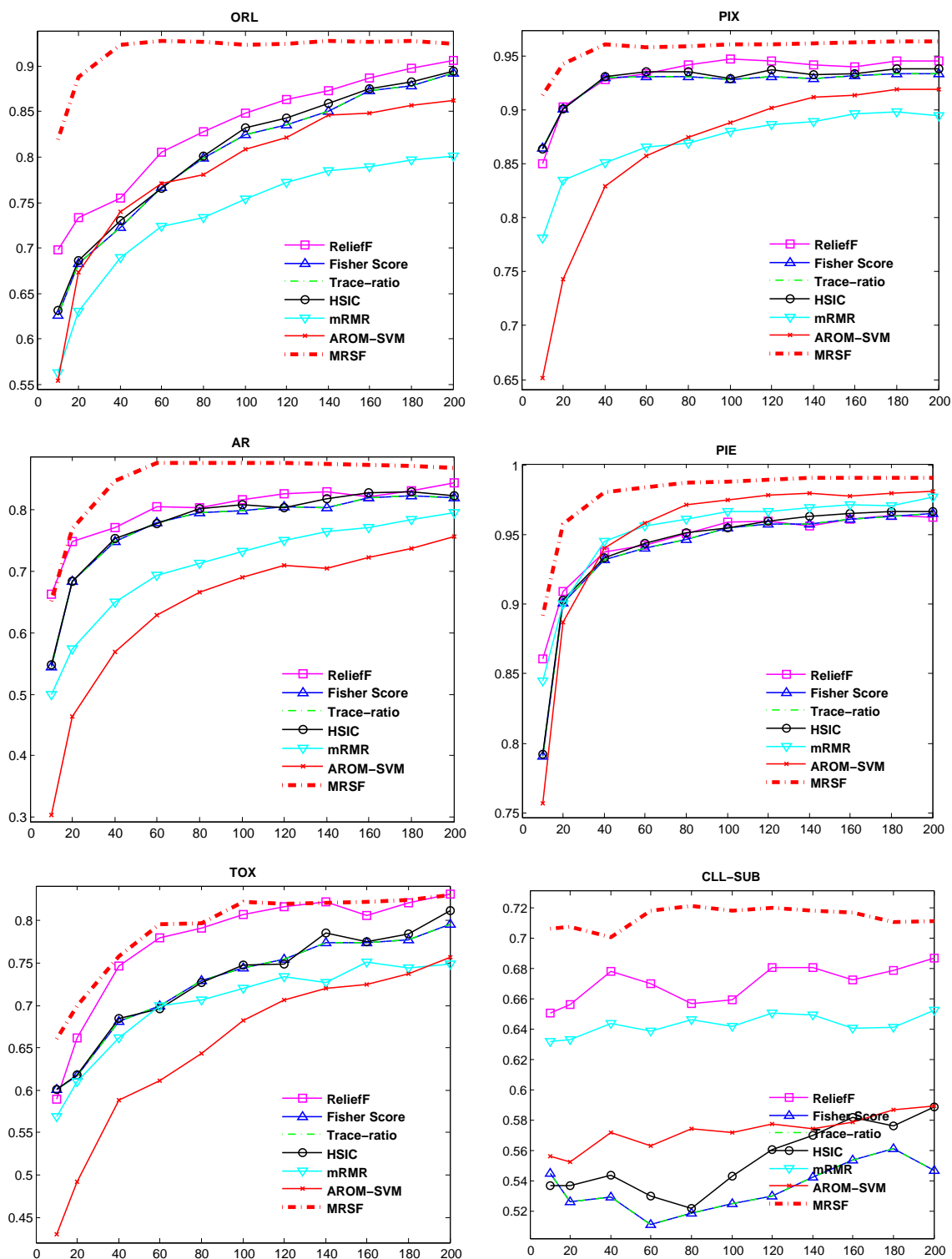


Figure 4.2: Study of supervised cases, plots for accuracy (y axis) vs. different numbers of selected features (x axis) on the six data sets. The higher the accuracy the better.

Table 4.2: Study of supervised cases: aggregated accuracy, the higher the better. The number in the parentheses is the p -Val obtained from t-test.

Algorithm	ORL	PIX	AR	PIE	TOX	CLL-SUB	AVE
ReliefF	0.83 (0.00)	0.93 (0.00)	0.80 (0.00)	0.94 (0.00)	0.77 (0.03)	0.67 (0.00)	0.82
Fisher Score	0.80 (0.00)	0.92 (0.00)	0.77 (0.00)	0.93 (0.00)	0.72 (0.00)	0.54 (0.00)	0.78
Trace-ratio	0.80 (0.00)	0.92 (0.00)	0.77 (0.00)	0.93 (0.00)	0.72 (0.00)	0.54 (0.00)	0.78
HSIC	0.80 (0.00)	0.93 (0.00)	0.77 (0.00)	0.94 (0.00)	0.73 (0.00)	0.55 (0.00)	0.79
mRMR	0.73 (0.00)	0.87 (0.00)	0.70 (0.00)	0.95 (0.00)	0.70 (0.00)	0.64 (0.00)	0.76
AROM-SVM	0.78 (0.00)	0.86 (0.00)	0.63 (0.00)	0.94 (0.02)	0.64 (0.00)	0.57 (0.00)	0.74
MRSF	0.91 (1.00)	0.96 (1.00)	0.84 (1.00)	0.98 (1.00)	0.79 (1.00)	0.71 (1.00)	0.86

Table 4.3: Study of supervised cases: averaged redundancy rate, the lower the better. The number in the parentheses is the p -Val obtained from t-test.

Algorithm	ORL	PIX	AR	PIE	TOX	CLL-SUB	AVE
ReliefF	0.92 (0.00)	0.79 (0.00)	0.77 (0.00)	0.36 (0.00)	0.34 (0.00)	0.59 (0.00)	0.63
Fisher Score	0.79 (0.00)	0.83 (0.00)	0.67 (0.00)	0.37 (0.00)	0.56 (0.00)	0.76 (0.00)	0.66
Trace-ratio	0.79 (0.00)	0.83 (0.00)	0.67 (0.00)	0.37 (0.00)	0.56 (0.00)	0.76 (0.00)	0.66
HSIC	0.79 (0.00)	0.83 (0.00)	0.67 (0.00)	0.37 (0.00)	0.56 (0.00)	0.76 (0.00)	0.66
mRMR	0.25 (0.29)	0.33 (0.00)	0.26 (0.00)	0.29 (0.00)	0.26 (0.00)	0.26 (0.00)	0.27
AROM-SVM	0.25 (0.44)	0.26 (1.00)	0.25 (0.00)	0.32 (0.00)	0.15 (1.00)	0.59 (0.00)	0.31
MRSF	0.25 (1.00)	0.35 (0.17)	0.21 (1.00)	0.24 (1.00)	0.16 (0.40)	0.21 (1.00)	0.24

Table 4.4: Study of unsupervised cases: averaged Jaccard score, the higher the better.. The number in the parentheses is the p -Val obtained from t-test.

Algorithm	ORL	PIX	AR	PIE	TOX	CLL-SUB	AVE
NB = 1							
Laplacian Score	0.07 (0.00)	0.05 (0.00)	0.07 (0.00)	0.04 (0.00)	0.10 (0.00)	0.06 (0.00)	0.07
SPEC	0.15 (0.00)	0.05 (0.00)	0.09 (0.00)	0.05 (0.00)	0.12 (0.00)	0.05 (0.00)	0.09
Trace-Ratio	0.06 (0.00)	0.05 (0.00)	0.08 (0.00)	0.03 (0.00)	0.12 (0.00)	0.08 (0.00)	0.07
HSIC	0.08 (0.00)	0.05 (0.00)	0.07 (0.00)	0.04 (0.00)	0.12 (0.00)	0.10 (0.00)	0.08
MRSF	0.56 (1.00)	0.53 (1.00)	0.41 (1.00)	0.41 (1.00)	0.31 (1.00)	0.17 (1.00)	0.40
NB = 5							
Laplacian Score	0.16 (0.00)	0.11 (0.00)	0.13 (0.00)	0.08 (0.00)	0.17 (0.00)	0.16 (0.00)	0.13
SPEC	0.28 (0.00)	0.11 (0.00)	0.16 (0.00)	0.11 (0.00)	0.19 (0.00)	0.14 (0.00)	0.17
Trace-Ratio	0.15 (0.00)	0.11 (0.00)	0.14 (0.00)	0.08 (0.00)	0.18 (0.00)	0.17 (0.00)	0.14
HSIC	0.16 (0.00)	0.13 (0.00)	0.14 (0.00)	0.10 (0.00)	0.18 (0.00)	0.16 (0.00)	0.14
MRSF	0.57 (1.00)	0.63 (1.00)	0.41 (1.00)	0.38 (1.00)	0.34 (1.00)	0.24 (1.00)	0.43

Table 4.5: Study of unsupervised cases: averaged redundancy rate, the lower the better. The number in the parentheses is the p -Val obtained from t-test.

Algorithm	ORL	PIX	AR	PIE	TOX	CLL-SUB	AVE
Laplacian Score	0.88 (0.00)	0.97 (0.00)	0.82 (0.00)	0.84 (0.00)	0.57 (0.00)	0.65 (0.00)	0.68
SPEC	0.72 (0.00)	0.97 (0.00)	0.75 (0.00)	0.77 (0.00)	0.47 (0.00)	0.59 (0.00)	0.61
Trace-Ratio	0.88 (0.00)	0.97 (0.00)	0.81 (0.00)	0.87 (0.00)	0.57 (0.00)	0.67 (0.00)	0.68
HSIC	0.88 (0.00)	0.97 (0.00)	0.80 (0.00)	0.82 (0.00)	0.57 (0.00)	0.64 (0.00)	0.67
MRSF	0.35 (1.00)	0.32 (1.00)	0.32 (1.00)	0.29 (1.00)	0.27 (1.00)	0.37 (1.00)	0.27

Chapter 5

INTEGRATING MULTIPLE KNOWLEDGE SOURCES IN FEATURE SELECTION

Much progress has been made over the last decade in developing effective feature selection algorithms [49, 28]. Many feature selection algorithms have been developed and proven to be effective in handling data of single source. One area in which feature selection is intensively used is gene selection based cDNA microarray where high-throughput microarray techniques generate gene expression data with 30K-50K features (genes or oligonucleotide probes), but only tens of samples [47, 5, 103, 9]. Given a cDNA microarray data, most existing feature selection algorithms try to identify genes that are differentially expressed over the samples. Discriminative genes help classifiers or clustering algorithms to achieve high accuracy [48, 20, 40]. However, does the better accuracy necessarily indicate higher biological relevance of genes? We applied a supervised feature selection algorithm, Fisher Score and an unsupervised algorithm, Laplacian Score on the expression profiling of bone marrow from 18 pediatric patients with acute lymphoblastic leukemia (ALL) [65] to select genes that may provide insight into the pathogenesis of pediatric ALL. The top 20 genes selected by the two algorithms are examined by our biologist collaborators. Table 5.1 contains a list of the biologically relevant genes identified by the biologists, and the accuracy achieved by the *knn* classifier on the selected genes. The result shows that a gene list of higher accuracy does not necessarily contain more relevant genes. Hence, selecting genes to achieve high accuracy should not be the sole goal of biological discovery.

There could be two sensible explanations. First, a cDNA Microarray data usually contains more than several thousand genes but only fewer than 100 samples. A data set of this kind usually leads to the small sample problem [69]. With so few samples, many genes, which are not biologically relevant, can easily gain their statistical relevance due

Table 5.1: Biologically relevant genes identified by two algorithms for childhood ALL.

Unsupervised (ACC: 0.61, REL: 7)			
SFRS5	TM9SF1	WTAP	GPSM3
STAC3	POMP	SLC25A6	
Supervised (ACC: 0.97, REL: 4)			
USP33	IL2RG	SIGIRR	CHCHD2

to randomness [81]. Second, even genes that are related, may have different importance. For instance, to understand a specific biological process, the genes acting as the “triggers” are much more important than the genes acting as the “fire”. Therefore, sometimes, the genes that act as the “fire” are not considered as relevant in biologists’ study. Addressing these problems goes beyond what the cDNA Microarray data can offer, and necessitates the need for additional information to conduct effective gene selection. Recent developments in bioinformatics have made various knowledge sources available, including the KEEG pathway repository [37], the Gene Ontology database [8] and the NCI Gene-Cancer database [75], etc. Recent work has also revealed the existence of a class of small non-coding RNA species known as microRNAs, which are surprisingly informative for identifying cancerous tissues [54]. The availability of these various knowledge sources presents unprecedented opportunities to advance research solving previously unsolvable problems. In this chapter, we study the novel problem of multi-source feature selection based on the proposed spectral feature selection framework to integrate multiple knowledge sources in the process of feature selection for improving the reliability of relevance estimation. The major challenge here is how to address the heterogeneity in different knowledge sources.

In gene selection research, researchers have tried to use various types of knowledge to assist gene selection. For instance, the authors in [1] propose to use different types of knowledge about genes to calculate gene similarity, which is then used to identify genes that are closest to the given example genes. In [88] the authors focus on using gene sets,

which are groups of genes that share common biological functions, chromosomal locations, or regulations to interpret the gene selection outputs. In [64], gene annotations are used for choosing gene ranking criteria. In [3], protein interaction, gene-disease association and gene function annotations are used for choosing cancer-related genes. Gene selection approaches using gene regulatory networks and gene ontology are also studied in [46] and [66, 86], respectively. Since most existing work is designed for specific research purposes, they can only handle one or limited types of knowledge of the same category. For instance, the models proposed in [88, 1, 3] can only handle knowledge about genes (features), but not knowledge about samples. To address this limitation, we propose an integrative approach to systematically incorporate different types of knowledge in feature selection. To achieve this, given multiple knowledge sources, we first extract a local sample distribution pattern from each source, which can be then combined to generate a global sample distribution pattern to reflect the intrinsic relationships among samples [44]. And the obtained global pattern can be used in spectral feature selection to achieve multi-source feature selection. Figure 5.6 presents the major steps in the approach.

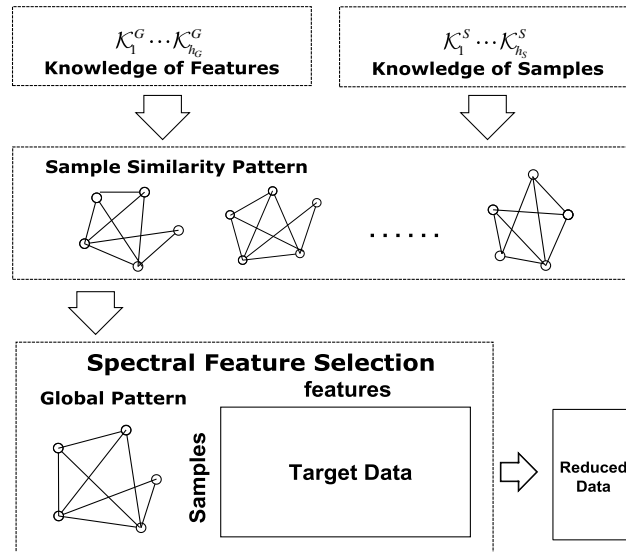


Figure 5.1: The framework of multi-source spectral feature selection.

As shown in Figure 5.6, the proposed spectral feature selection approach contains three steps: (1) Knowledge Conversion - knowledge understandable for human beings may not be directly applicable in a learning model. Therefore, the first step is to use different types of knowledge to extract sample similarity. Assume we have L different knowledge sources $\mathcal{K}_1, \dots, \mathcal{K}_L$. For the i -th knowledge source, we can apply a conversion operator $c_i(\cdot)$ to extract a local specification of the sample similarity matrix \mathbf{S}_i , and this allows us to formalize knowledge conversion with the following equation:

$$\mathbf{S}_i = c_i(\mathcal{K}_i), i = 1, \dots, L \quad (5.1)$$

(2) Knowledge Integration - Given multiple local sample similarity matrices, a global similarity matrix can be obtained by linearly combining local similarity matrices [114].

$$\mathbf{S}_{global} = \sum_{i=1}^L \alpha_i \mathbf{S}_i \quad (5.2)$$

In the equation, and α_i is the combination coefficient, which can be assigned by domain experts according to their domain knowledge [114], or, if the label information is available, learned automatically via convex optimization based on a set of kernel matrices carrying the information about the local geometric patterns. We refer readers to literature for comprehensive study on the research issues of kernel combination [44, 99].

(3) Feature Selection - after obtained the \mathbf{S}_{global} it can be used in the proposed spectral feature selection framework for feature selection.

Next, we will study how to categorize the external knowledge sources, and how to define the converting operators $c(\cdot)$ to extract local sample similarity from different types of knowledge with heterogeneous representations.

5.1 Categorization of Different Types of Knowledge

Different types of knowledge need to be handled properly in multi-source feature selection. We now study how to categorize different types of knowledge that can be used in

feature selection. To assist understanding, gene selection is used as an application context to illustrate the ideas, in which genes are corresponding to features. We also study how to efficiently and effectively extract sample similarity information from different types of knowledge, which are of heterogeneous representations.

Various types of knowledge sources can be used in feature selection. We categorize them into two groups: the knowledge about features, and the knowledge about samples. The knowledge about features usually contains information about the properties of features or their relationships. Figure 5.2 presents three different types of knowledge about genes (features) that can be used in gene selection: (a) metabolic pathway, which depicts a series of biochemical reactions occurring in cells and reflects how genes interact with each other to accomplish a specific function; (b) gene ontology (GO) annotation [8], which uses a controlled vocabulary to describe the characteristics of genes; and (c) gene sequence, which describes the order of the nucleotide bases of genes. The figure shows that the three types of knowledge have heterogeneous representations. The nature of the knowledge determines how it can be used in feature selection. According to the way knowledge is used in

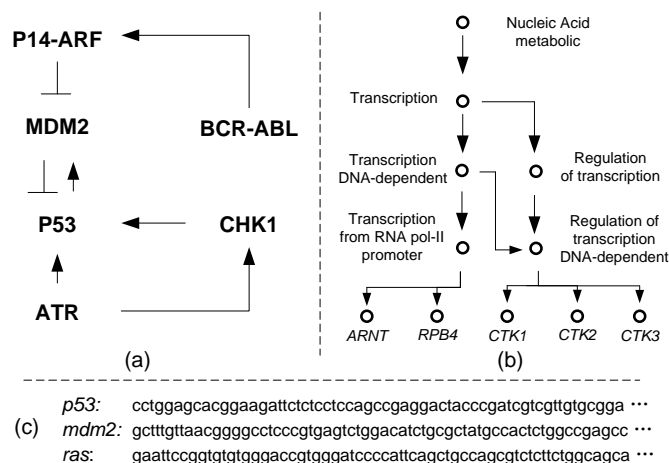


Figure 5.2: An example of three different types of knowledge about genes (features): (a) Metabolic Pathway, (b) Gene Ontology Annotation, and (c) Gene Sequence.

feature selection, we further divide different types of knowledge into three categories: (1) knowledge about feature similarity, \mathcal{K}_{SIM}^F , for example, with gene sequence information, gene similarities can be obtained by applying a sequence alignment algorithm. (2) Knowledge of feature functions, \mathcal{K}_{FUN}^F , for instance, in a metabolic pathway, a set of genes act together to accomplish particular biological functions; and in gene ontology annotation, the functions of genes are also provided. (3) Knowledge of feature interaction, \mathcal{K}_{INT}^F , for example, in the BioGRID [87], over 198k genetic interactions related to different types of biological processes are recorded. The knowledge of features is accumulated and cross-examined by human researchers in their research by generalizing evidences from multiple experiments, therefore, is more reliable, and independent of any specific experiment.

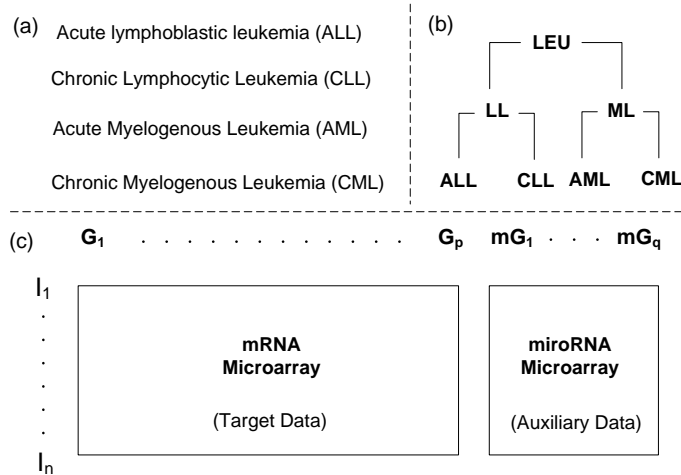


Figure 5.3: Different types of knowledge about samples, (a) the class label information, (b) sample hierarchy, and (c) an example of the auxiliary data.

The knowledge of samples usually is about sample categories, \mathcal{K}_{CAT}^S , or samples' similarity relationship, \mathcal{K}_{SIM}^S . Samples can be categorized with either a flat structure (as shown in Figure 5.3-(a), which forms the standard class label) or a hierarchical structure, as shown in Figure 5.3-(b). The similarity among samples, depicted by the pairwise sample similarity matrix, can be derived from a given auxiliary data. Auxiliary data refers to the data

containing additional information of the same set of samples in the target data. The target and the auxiliary data depict the same set of samples, while using different measurements. Auxiliary data may help us get a better understanding of the geometric pattern of the samples. For example, as shown in Figure 5.3-(c), for gene selection, the microRNA Microarray can serve as auxiliary data, which measures the microRNA expression of samples. cDNA Microarray and microRNA Microarray are collected from the same set of samples. Compared to cDNA Microarray, microRNA Microarray contains the expression of only several hundreds of microRNA and are found to be surprisingly informative in separating tissues of cancer and noncancer, as well as tissues of different types of cancers [32]. Using microRNA Microarray as auxiliary data helps improve our understanding about how cancerous samples cluster together. Comparing with knowledge about features, knowledge about samples links to individual experiment, therefore is more specific.

Table 5.2: The categories and examples of different types of knowledge.

Knowledge	Samples	\mathcal{H}_{CAT} - Category	Class Label, Sample Hierarchy
		\mathcal{H}_{SIM} - Similarity	miRNA Expression Profile, mRNA Expression Profile
	Genes	\mathcal{H}_{SIM} - Similarity	Gene Sequence, Gene Ontology Annotation, Gene Lineage, Gene Locus
		\mathcal{H}_{FUN} - Function	Gene Ontology Annotation, Metabolic Pathway, Gene-Disease Association
		\mathcal{H}_{INT} - Interaction	Metabolic Pathway, Protein-Protein Interaction

Table 5.2 summarizes different categories of knowledge that can be used in feature selection. We noticed that some types of knowledge fall into more than one categories. For instance, gene ontology annotation can be used for obtaining the knowledge of both gene similarities, e.g. by comparing shared annotation terms among genes, and gene functions, e.g. by finding out the annotation terms related to specific functions of interest. Different types of knowledge have heterogenous representations and describe feature or samples

from different perspectives. The categorization of different types of knowledge helps us identify the common character of the knowledge from the same category, so that a common approach can be applied on the knowledge in that category for knowledge conversion.

5.2 Knowledge Conversion

We study how to extract sample similarity from different knowledge sources. The conversions of $\mathcal{H}_{CAT}^S \rightarrow \mathcal{H}_{SIM}^S$ is straightforward. For example, given \mathcal{H}_{CAT}^S , the category information of samples, Eq. (3.13) can be used in to construct sample similarity. When applied in the spectral feature selection framework SPEC, this leads to the Fisher Score feature selection algorithm. Below, we show how to perform conversions for: $\mathcal{H}_{SIM}^F \rightarrow \mathcal{H}_{SIM}^S$, $\mathcal{H}_{FUN}^F \rightarrow \mathcal{H}_{SIM}^S$, and $\mathcal{H}_{CON}^F \rightarrow \mathcal{H}_{SIM}^S$. Figure 5.4 illustrates how to convert \mathcal{H}_{SIM}^F and \mathcal{H}_{FUN}^F to \mathcal{H}_{SIM}^S . The idea is to involve the three types of knowledge in the calculation of the pairwise similarity among samples.

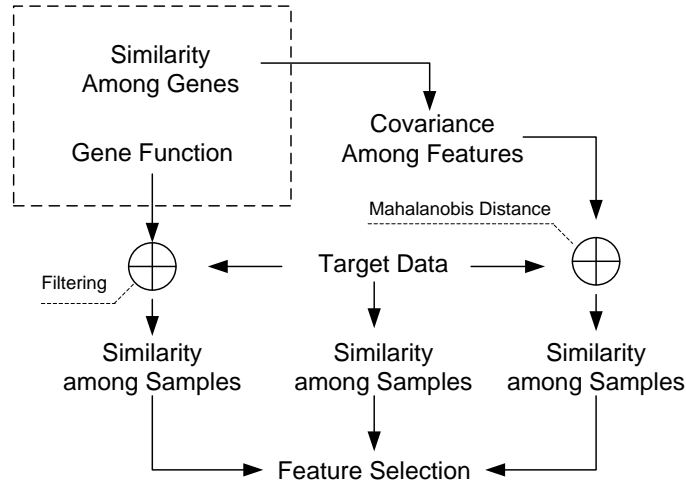


Figure 5.4: Calculating sample similarity using different types of knowledge of genes.

5.2.1 $\mathcal{H}_{SIM}^F \rightarrow \mathcal{H}_{SIM}^S$

Given similarities among features, feature covariance can be constructed and used in calculating the pairwise sample similarity via Mahalanobis distance [58], which is defined

as:

$$\|\mathbf{x} - \mathbf{y}\|_M^2 = (\mathbf{x} - \mathbf{y})^T \mathbf{C}^{-1} (\mathbf{x} - \mathbf{y}). \quad (5.3)$$

In the equation, $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m$ are two samples with m features F_1, \dots, F_m , and $\mathbf{C} \in \mathbb{R}^{m \times m}$ is the covariance matrix. In comparison to the standard Euclidian distance, Mahalanobis distance provides a better way to determine the similarities among samples by considering the probability distribution of the underlying model, and the ellipsoid best representing the probability distribution can be estimated from \mathbf{C} [30]. In real applications, \mathbf{C} is usually estimated by the following equation:

$$\mathbf{C} = \frac{1}{n-1} \sum_{k=1}^n (\mathbf{x}_k - \bar{\mathbf{x}})(\mathbf{x}_k - \bar{\mathbf{x}})^T, \quad (5.4)$$

where $\mathbf{x}_1, \dots, \mathbf{x}_n$ are the n samples of the data, with $\bar{\mathbf{x}}$ being their mean. Although Equation (5.4) specifies an unbiased estimator of the covariance matrix, when sample size is small, it may return a poor estimation. Instead of using the data, the covariance matrix can also be obtained from our knowledge of feature similarities, which may provide another (more stable and reliable) way for estimating \mathbf{C} . By the following theorem, we show how to construct the covariance matrix from \mathcal{H}_{SIM}^F , the knowledge of feature similarity.

Proposition 4 *Given $\mathbf{S}^F \in \mathbb{R}^{m \times m}$ with s_{ij} specifying the similarity between features F_i and F_j . Let \mathbf{D}^F be a diagonal matrix with $d_{ii} = \sum_k s_{ik}$, then $\mathbf{K} = (\mathbf{D} - \mathbf{W})^+$ specifies a kernel. Using its embedding, the covariance matrix can be obtained by:*

$$\mathbf{C} = \mathbf{K} \left(\mathbf{I} - \frac{1}{l} \mathbf{U} \mathbf{1} \mathbf{1}^T \mathbf{U}^T \right) \mathbf{K}. \quad (5.4)$$

In the proposition, l is the number of involved features, $\mathbf{1}$ is the vector with 1 as its only elements. $(\cdot)^+$ denotes the pseudo-inverse and $\mathbf{K} = \mathbf{U} \mathbf{\Sigma} \mathbf{U}^T$ is the SVD [25] of \mathbf{K} .

5.2.2 $\mathcal{H}_{FUN}^F, \mathcal{H}_{CON}^F \rightarrow \mathcal{H}_{SIM}^S$

In applications, some particular functions or certain types of feature interactions may be of special interests according to the research purpose. For example, in gene selection for

cancer study, genes with certain types of functions or participate in certain types of biological processes (genetic interactions) are especially interesting to biologists. Given \mathcal{H}_{FUN}^F or \mathcal{H}_{CON}^F , and \mathbb{F} , a set of feature functions of interests, or \mathbb{I} , a set of feature interactions of interests, data can be filtered by the features associated with \mathcal{F} , or \mathbb{I} :

$$\mathbf{X}_{\mathbb{F}} = \Pi_{\mathbf{F}_{\mathbb{F}}}(\mathbf{X}), \quad \mathbf{X}_{\mathbb{I}} = \Pi_{\mathbf{F}_{\mathbb{I}}}(\mathbf{X}). \quad (5.5)$$

Here $\mathbf{F}_{\mathbb{F}}$ and $\mathbf{F}_{\mathbb{I}}$ are the features related to \mathbb{F} and \mathbb{I} , respectively. And $\Pi(\cdot)$ is the projection operator. Using the filtered data $X_{\mathbb{F}}$ or $X_{\mathbb{I}}$, a pairwise sample similarity matrix \mathbf{S} can be obtained through any similarity measure. Since all features in $\mathbf{F}_{\mathbb{F}}$ (or $\mathbf{F}_{\mathbb{I}}$) are related to the feature functions (or feature interactions) of interest, sample similarity matrix \mathbf{S} should reflect the distribution under the influence of the functions (or the interactions). In case the functions (or the interactions) are closely related to the target concept under study, the distribution will give us an insight of the target concept, and help us to select relevant features. Using features which are known to have a particular function (or participate in a particular interaction) as the seeds can also help us select features that perform the same function (or participate in the same interaction) but are still unknown.

5.3 MSFS - The Framework

The above technical discussion has paved way for us to propose a framework for *Multi-Source Fene Selection: MSFS*. The detail of the framework can be found in Algorithm 1. It consists of three major steps: (1) obtaining local sample similarity from each data source (Lines 3-5); (2) combining the local sample similarity to construct a global sample similarity (Line 6); and (3) using the global sample similarity in spectral feature selection to select features (Lines 7). Below we give analysis on time complexity for *MSFS*.

Since the representation of the L data sources are heterogenous, the time complexity of extracting local sample similarity from knowledge sources can vary greatly. Assuming for knowledge sources $\mathcal{H}_1^F \dots \mathcal{H}_{L_F}^F, \mathcal{H}_1^S \dots \mathcal{H}_{L_S}^S$, each knowledge sources provides us an

Algorithm 3: MSFS: Multi-Source Feature Selection

<p>Input: $\mathcal{K}_1 \dots \mathcal{K}_L$ and X</p> <p>Output: $List_F$ - the selected feature list</p> <ol style="list-style-type: none"> 1 forall the $\mathcal{K}_i \in (\mathcal{K}_1 \dots \mathcal{K}_L)$ do 2 construct \mathbf{S}_i, the local sample similarity; 3 end 4 obtain global sample similarity \mathbf{S} from \mathbf{S}_is; 5 feed \mathbf{S} in to SPEC to select feature and form $List_F$; 6 return $List_F$;

affinity matrix depicting feature similarity and involving l features, then constructing \mathbf{C}_F using Equation (4) and calculating its (pseudo) inverse requires $O(l^3)$ operations. Computing Mahalanobis distance among n^2 pairs of instances and forming a RBF kernel require $O(l^2n^2)$ operations. Crossing the L_F knowledge source, the total cost is $O((l^2n^2 + l^3)L_F)$. Assuming for $\mathcal{K}_1^S \dots \mathcal{K}_{L_S}^S$, each knowledge source has m features depicting the same set of n samples, and we use RBF kernel to represent the local sample similarity pattern. The time complexity of forming a RBF kernel on each knowledge source is $O(mn^2)$. And crossing L_S knowledge sources, the cost is $O(L_S n^2 m)$. Therefore, the total cost of the first step is $O((l^2n^2 + l^3)L_F + n^2mL_S)$. Assuming we linearly combine \mathbf{S}_i s with a set of prespecified combination coefficients, the cost is $O((L_F + L_S)n^2)$. Using SPEC- $\phi_2(\cdot)$ to select features¹, the cost is $O(mn^2)$. Therefore the overall time complexity of the proposed MSFS framework for multi-source feature selection is $O((l^2n^2 + l^3)L_F + n^2mL_S)$.

5.4 Experimental Study on Human Cancer Data

We empirically evaluate the performance of *MSFS* for multi-source feature selection selection to identify cancer related genes in cancer study based on cDNA Mairoarray. To

¹We do not use MRSF in this application. This is because in gene selection, the research purpose is to identify genes that are related to the biological process under study. Since the sample size is small removing redundant features, may increase the risk of missing biologically relevant genes.

evaluated the quality of selected genes, we choose *accuracy*² and *hit ratio*³ as performance measures. In the experiment, label information is only used after selecting genes to calculate accuracy during the testing phase to measure how good selected genes are. In addition, we also measure *robustness* of selected genes by varying the number of classes, if class information is used in gene selection.

Human Cancer Data. It consists of five heterogeneous data sources. Two sets of gene expression profiles from a mixture of 88 normal and cancerous tissue samples⁴: a miRNA expression profile for 151 human miRNAs and a mRNA expression profile for 16,063 human mRNAs [54, 32]. miRNA profile provides relationships among samples and it is observed in [54] that comparing with mRNA, miRNA expression profiles is of more power in terms of discriminating cancer from noncancer tissues as well as cancer of different types of tissues. Three gene information profiles are provided: *Gene Function Annotation*, *Biological Pathway* and *Gene Ontology Annotation*. The three profiles are generated as follows. (1) *Gene Function Annotation*: we used the name of involved tissues (e.g., colon, lung, ...) as keywords to search in IPA system [34] for cancer related processes, for each matching process, we add all genes involved in the process to \mathbf{F}_F , which is the set of genes with relevant function. This resulted in a set containing 535 genes. \mathbf{F}_F is used to filter the data as described in Section 5.2.2. (2) *Biological Pathway*: with the mRNA expression profile, we used the IPA system to infer the relevant pathways, which results in about 40 networks of molecules, connected these networks and obtained a graph involving 571 genes. The genes is added to \mathbf{F}_I , which is the set of genes with relevant interactions (corresponding to certain biological processes). And \mathbf{F}_I is used to filter the data as described in Section 5.2.2. (3) *Gene Ontology Annotation*: 68 Gene Ontology annotation [8] terms,

²For accuracy, we first filter the data with selected genes and build a classifier on the filtered data, then obtain its accuracy as performance measure.

³For hit ratio, given a list of genes, we check how many of the genes are cancer related by using the gene function annotation information provided by Ingenuity Pathways Analysis (IPA) system [34].

⁴Colon, Pancreas, Kidney, Bladder, Prostate, Ovary, Uterus, Lung, Mesothelioma, Melanoma and Breast.

related to cell cycle, cell growth, differentiation, apoptosis, cancer development and so on, were provided by domain experts. According to whether a term is assigned to a gene, we obtained a matrix with a size of 16063×68 , which can be used to compute a covariance matrix among genes. The three profiles provide relationships information among genes. Using the five profiles, we obtained two sets of data: 2C-DATA and 4C-DATA. 2C-DATA contains all 88 tissue samples of the original data and class label is assigned to a sample according to whether the sample is a normal or a tumor tissue. 4C-DATA contains 33 tissue samples from 4 types cancerous tissues, *Mesothelioma*, *Uterus*, *Colon* and *Pancreas*, each has at least 7 samples. A summary of the two data sets is given in Table 5.3.

Table 5.3: A summary of the Human Cancer Data.

Type	Data Sources	Genes	Samples
2C DATA			
\mathcal{D}	mRNA Expression Profile (the target data)	16063	88
\mathcal{H}_1^S	miRNA Expression Profile	151	88
\mathcal{H}_1^F	Gene Function Annotation	535	-
\mathcal{H}_2^F	Biological Pathway	571	-
\mathcal{H}_3^F	Gene Ontology Annotation	4385	-
4C DATA			
\mathcal{D}	mRNA Expression Profile (the target data)	16063	33
\mathcal{H}_1^S	miRNA Expression Profile	151	33
\mathcal{H}_1^F	Gene Function Annotation	535	-
\mathcal{H}_2^F	Biological Pathway	571	-
\mathcal{H}_3^F	Gene Ontology Annotation	4385	-

In the experiment, we compare feature selection using single (target) data source with using multiple data sources. We chose 3 feature selection algorithms for gene selection using single data sources. They are two unsupervised algorithms, Laplacian Score [31] and PathSPCA [14], and one supervised algorithms, ReliefF [80]. Experiments are performed in the Matlab environment. We use *RBF* kernel to extract the local sample similarity. A domain expert determined two sets of combination coefficients for linear combination of

the local geometric patterns: COMB-1, using the combination coefficients (0.0, 0.5, 0.5, 0.0, 0.0) for 2C-DATA and (0, 0.3, 0.4, 0.3, 0) for 4C-DATA; and COMB-2, using (0.1, 0.3, 0.3, 0.2, 0.1) for 2C-DATA and (0.1, 0.2, 0.4, 0.2, 0.1) for 4C-DATA. The usefulness of each data source can also be determined by checking the quality of the genes selected by *MSFS* using the sample similarity extracted from the data source. We also tried to use class label to learn combination coefficients via a kernel learning approach proposed in [99]. We apply 1NN classifier with selected genes, and use its accuracy to measure the quality of the gene set. Reported results are based on averaging the results from 10 trials of experiments.

Experimental results are organized in terms of two sets of multi-source data. When label information is available, we study how to incorporate it in *MSFS* and its effects. We examine the biological relevance of the genes selected by *MSFS*.

5.4.1 Results on 2C-DATA

Figure 5.5-(a) compares unsupervised baseline gene selection algorithms using mRNA profiles with *MSFS* using five individual profiles as well as the combinations of them. The results show that by using miRNA profiles, COMB-1 and COMB-2, *MSFS* selects genes that provide the best accuracy which are significantly better than those achieved by genes selected using baseline algorithms. Table 5.4 shows the detailed accuracy results as well as hit ratio. According to hit ratio, averagely, using COMB-2, the gene list provided by *MSFS* containing the most known cancer relevant genes (7), which is following by using COMB-1 (6.6), miRNA (6.6) and Function Annotation (5). According to hit ratio, Biology Pathway is also helpful (3.6). The averaged number of selected known cancer related genes is 5.8 for SPCA and 3.2 for Laplacian score. We notice that the first two genes selected by *MSFS* with miRNA profile, Function Annotation, COMB-1 and COMB-2 are all known to be cancer relevant. We will provide biological relevance analysis for selected genes later. The observations suggest that (1) the sample similarity obtained from miRNA profile in-

deed possesses better discriminative power, which is consistent with the findings in [54], and (2) given a sample similarity with higher quality, *MSFS* can select better genes. This support the use of multiple data sources in gene selection, and show that *MSFS* is effective. (3) By combining multiple heterogeneous data sources we are able to achieve better performance than using any individual data source. This is consistent with the observations in [44]. Figure 5.5-(c) plots the performance of *MSFS-VAR* when different combination coefficients are used to combine local sample similarity obtained from miRNA profile and Function Annotation. We observe that the highest accuracy is achieved by using the two data sources together. This indicates the existence of complimentary information, which helps to improve the estimation of the sample similarity of the underlying model.

5.4.2 Results on 4C-DATA

Figure 5.5-(b) and Table 5.4 contain the results on 4C-DATA. We obtained largely similar observations as those from using 2C-DATA. Since the number of samples becomes smaller but the number of classes becomes larger, we observe that the unsupervised baseline algorithms perform worse comparing with on 2C-DATA. However, the performance of *MSFS* using miRNA profile, Function Annotation, Biological Pathway, COMB-1 and COMB-2 are consistently good in terms of accuracy and hit ratio. This indicates that *MSFS* can effectively select relevant genes given high quality sample similarity.

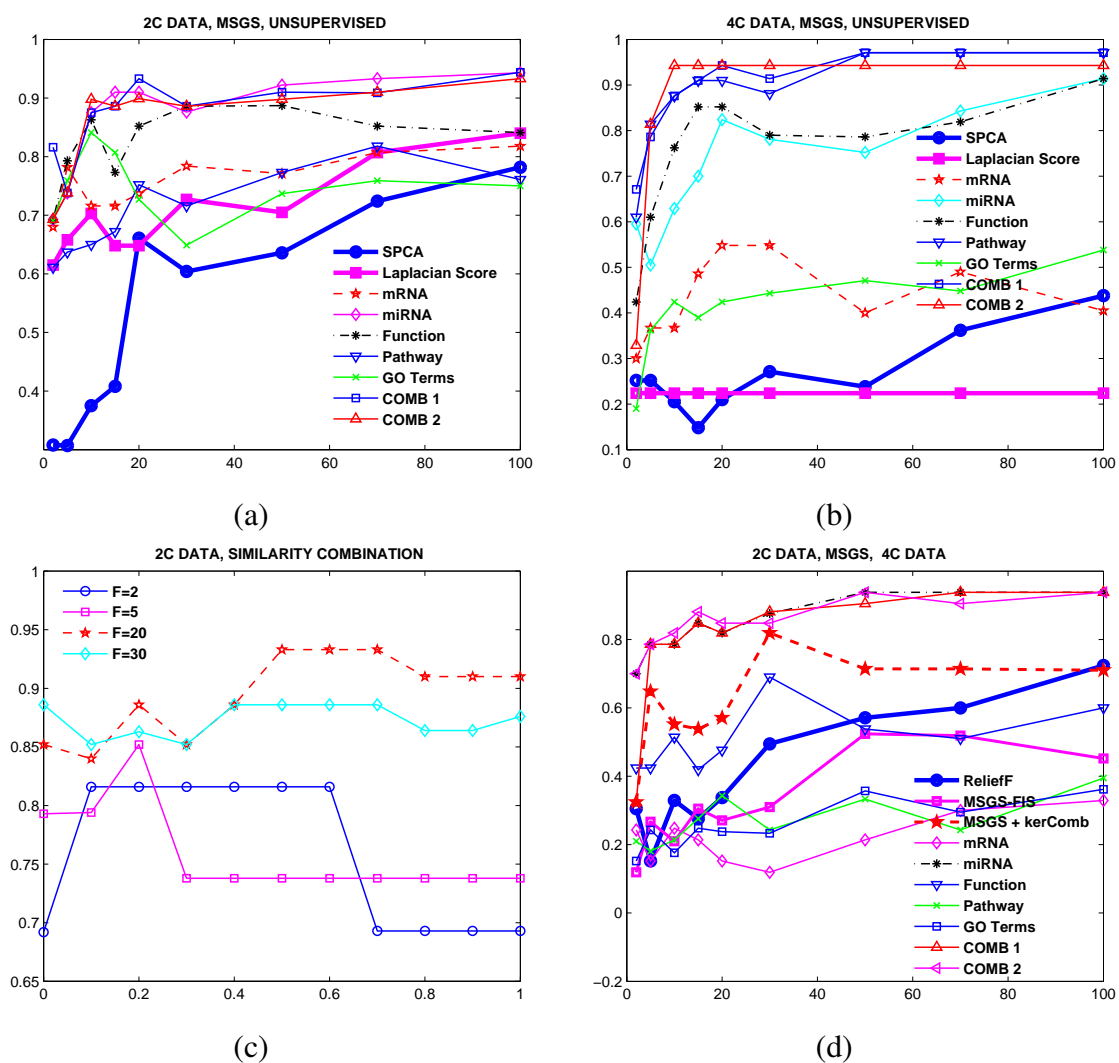


Figure 5.5: Charts (a,b,d): accuracy (y axis) vs. different numbers of genes (x axis). Chart (c): accuracy (y axis) vs. different combination coefficient (x axis).

Table 5.4: Results of accuracy and hit ratio. The numbers with bold typeface indicate the highest accuracy or hit ratio.

ALGORITHMS	2	5	10	20	30	Ave	2	5	10	20	30	Ave
	UNSUPERVISED, 2C-DATA						UNSUPERVISED, 4C-DATA					
SPCA	0.31	0.31	0.38	0.66	0.60	0.45	0.25	0.25	0.21	0.21	0.27	0.24
	1	4	5	8	11	5.8	2	2	2	3	6	3
Laplacian Score	0.62	0.66	0.70	0.65	0.73	0.67	0.22	0.22	0.22	0.22	0.22	0.22
	0	0	2	6	8	3.2	1	2	3	8	10	4.8
	MSFS, 2C-DATA						MSFS, 4C-DATA					
MSFS	0.68	0.78	0.72	0.74	0.78	0.74	0.15	0.43	0.43	0.37	0.24	0.32
mRNA	0	0	2	3	6	2.2	0	0	0	1	2	0.6
MSFS	0.69	0.74	0.88	0.91	0.88	0.82	0.60	0.51	0.63	0.82	0.78	0.67
miRNA	2	4	6	8	13	6.6	1	2	5	10	13	6.2
MSFS	0.69	0.79	0.86	0.85	0.89	0.82	0.42	0.61	0.76	0.85	0.79	0.69
Function	2	4	5	6	8	5	2	3	7	13	16	8.2
MSFS	0.61	0.64	0.65	0.75	0.72	0.67	0.61	0.81	0.88	0.91	0.88	0.82
Pathway	0	1	3	6	8	3.6	2	3	5	12	17	7.8
MSFS	0.69	0.76	0.84	0.73	0.65	0.73	0.33	0.33	0.40	0.50	0.38	0.39
GO Terms	0	0	2	2	6	2	1	1	3	4	6	3
MSFS	0.82	0.74	0.88	0.93	0.89	0.85	0.67	0.79	0.88	0.94	0.91	0.84
COMB-1	2	4	6	8	13	6.6	2	5	8	11	17	8.6
MSFS	0.69	0.74	0.90	0.90	0.89	0.82	0.33	0.81	0.94	0.94	0.94	0.79
COMB-2	2	4	7	9	13	7	2	5	6	11	15	7.8

5.4.3 Incorporating Label Information

When label information is available, we can incorporate it in *MSFS* in two ways: (1) using the label information to construct sample similarity as formulated in Eq. (3.13), and input the obtained sample similarity into *MSFS*, which is denoted as *MSFS-FIS*. It is equivalent to Fisher Score. (2) Learning the combination coefficient automatically with a supervised kernel learning algorithm and input the combined pattern into *MSFS*. In this work we implement *MSFS-kerCB*: we use the approach proposed in [99] to learn the combination coefficients⁵. For comparison we also include a supervised selection algorithm ReliefF [80] in the experiment. The upper part of Tabel 5.5 shows the performance of supervised gene selection algorithms on 2C-DATA. For accuracy, *MSFS-FIS* performs the best (0.92). For average hit ratio, *MSFS-KerCB* performs the best (5.6), while we observed that the averaged hit ratio of *MSFS* with COMB-2 is 7.

Furthermore, we experimented how robust supervised gene selection is by applying the genes selected on 2C-DATA by supervised algorithms and *MSFS* to 4C-DATA to check their discriminative power. The results are shown in the lower part of Table 5.5 and Figure 5.5-(d). We observe that the genes selected by *MSFS* using miRNA profile, COMB-1 and COMB-2 can discriminate cancer from different types of tissues, however, those selected by supervised algorithms cannot. This finding suggests that (1) results of supervised feature selection hinge upon the target concept and the change of class information can results in the selection of different genes; and (2) the geometric pattern corresponding to miRNA profiles as well as COMB-1 and COMB-2 is relatively stable - genes selected by *MSFS* can discriminate data of different numbers of classes, indicating that it may be consistent with intrinsic structure of the underlying model and more robust. *MSFS* is an unsupervised algorithm and obtains intrinsic patterns that do not vary with class definitions

⁵The learnt coefficients on 2C-DATA is: (0,0.11,0.89,0,0)

from multiple data sources. This is not so for supervised algorithms: with different class definitions, different sets of genes will be selected, which is clearly shown in Table 5.5.

Table 5.5: Upper: accuracy and hit ratio by supervised algorithms. *MSFS* with COMB-2 is unsupervised and listed for comparison. Lower: accuracy on 4C-DATA with genes selected by from 2C-DATA.

ALGORITHMS	2	5	10	20	30	Ave
ON 2C-DATA						
RELIEFF	0.83	0.90	0.92	0.95	0.96	0.91
	0	1	3	9	12	5
MSFS	0.93	0.90	0.92	0.92	0.93	0.92
FIS	0	1	3	5	8	3.4
MSFS	0.82	0.81	0.79	0.81	0.84	0.81
KerCB	2	3	5	8	10	5.6
MSFS	0.69	0.74	0.90	0.90	0.89	0.82
COMB-2	2	4	7	9	13	7
ON 4C-DATA						
RELIEFF	0.31	0.15	0.33	0.34	0.50	0.32
MSFS-FIS	0.12	0.27	0.21	0.27	0.31	0.24
MSFS-kerCB	0.32	0.65	0.55	0.57	0.82	0.58
LAPLACIAN	0.21	0.39	0.40	0.34	0.39	0.34
SPCA	0.22	0.25	0.22	0.34	0.33	0.27
mRNA	0.24	0.15	0.25	0.15	0.12	0.18
miRNA	0.70	0.79	0.79	0.82	0.88	0.79
Function	0.42	0.42	0.51	0.48	0.69	0.51
Pathway	0.21	0.18	0.21	0.34	0.24	0.24
GO Terms	0.15	0.24	0.18	0.24	0.23	0.21
COMB-1	0.32	0.79	0.79	0.82	0.88	0.72
COMB-2	0.70	0.79	0.82	0.85	0.85	0.80

5.4.4 A further Study of Gene Biological Relevance

In the experiments above, we used hit ratio to measure biological relevance of selected genes. In order to closely examine biological relevance of selected genes, we performed a further study in which our biologist collaborators examined the top 20 genes selected

by *MSFS* with COMB-2 on 2C-DATA.⁶ It turned out that 17 of the top 20 genes were experimentally confirmed to be cancer related, except for *SMNT*, *LAD1* and *LMOD1*. Detailed information of the selected genes can be found in Table 5.6. Among them, nine were already annotated to be related to cancer or tumor in the IPA system. The other genes are found to be differentially expressed in unique or several cancer cell lines and are supported by literature. Enzymes involved in metabolism, such as *FABP1* and *GPX2* (well-known oxidoreductase responsive to oxidative stress) were selected. *FABP1* plays an important role in lipid metabolism and investigations have demonstrated altered systemic lipid metabolism in cancer patients [24]. *GPX2* is one of the major cellular antioxidants as biomarkers for reactive oxygen species (ROS) producing in animal, and plants. Hydrogen peroxides, one of ROS, has been shown to act as tumor promoters [82]. Several tumor suppressor genes were also identified, including *FHL1*, *SPARCL1* and *SYNPO2*. Interestingly, most of these genes encode transmembrane proteins, implicating their role in signal transduction during cancer development. Thus, our analyses show that multi-source feature selection can select genes bearing both statistical significance and biological relevance.

The obtained results on Human Cancer Data confirm the efficacy of multi-source gene selection algorithm *MSFS* as well as the potential of multi-source gene selection.

⁶We chose this list because that the list results in high accuracy on both 2C-DATA and 4C-DATA and its hit ratio is also high.

Table 5.6: The top 20 genes selected by MSFS with COMB-2 on 2C-DATA. Genes with boldface names are known to be cancer related. Genes are ordered based on their relevance from highest to lowest.

Gene name	Gene Description	Functions and Biological Process	Disease
LGALS4	lectin, galactoside-binding soluble, 4	sugar binding, cell adhesion	colon cancer
CNN1	calponin 1	calmodulin binding,actin filament binding	bone cancer
MYH11	myosin heavy chain 11	calmodulin binding,actin binding	lung cancer
FUT6	fucosyltransferase 6	integral to membrane,L-fucose catabolism	colon cancer
LTF	lactotransferrin	ubiquitin ligase complex	prostate cancer
OLFM4	olfactomedin 4	latroxin receptor activity	gastric Cancer
FABP1	fatty acid binding protein 1	fatty acid metabolism,GABA-A receptor	prostate cancer
SYNPO2	synaptopodin 2	actin binding	prostate cancer
MYLK2	myosin, light chain kinase	protein serine/threonine kinase activity	colorectal cancer
GPX2	glutathione peroxidase 2	oxidoreductase,response to oxidative stress	breast cancer
SPARCL1	SPARC-like 1	calcium ion binding	brain cancer
TM4SF3	tetraspanin 8	signal transducer activity	colon, prostate cancer
TACSTD1	tumor-assoc calcium signal tran 1	plasma membrane	ovarian cancer
KRT15	keratin 15	structural constituent of cytoskeleton	breast cancer
FHL1	Four and a half LIM domains 1	extracellular space,complement activation	brain cancer
LMOD1	leiomodin 1	tropomyosin binding	-
NFIB	nuclear factor I/B	transcription factor activity	breast,brain cancer
LADI	ladinin 1	anchoring filament	-
CDH17	cadherin 17	transporter activity, calcium ion binding	colon cancer
SMTN	smoothelin	actin binding,muscle development	-

5.5 Discussion

In this chapter, we investigated a novel problem arising from the need to select features on one data source given multiple sources. We extend the proposed spectral feature selection framework SPEC to achieve multi-source feature selection based on similarity combination. We designed and conducted extensive experiments to objectively and systematically evaluate the proposed extension, MSFS, in comparison with existing representative single source feature selection methods. The affirmative results demonstrate that using multiple knowledge sources can help improve feature selection of the target data. As multi-source data become more common, learning and feature selection using multi-source data will be in high demand in many real applications.

One limitation of MSFS is that it relies on combining sample similarity, which restricts the model flexibility. To address this limitation, in [112], we propose to develop a general approach for systematically integrating different types of knowledge to achieve Knowledge-Oriented multi-source gene selection, which is named KOGS.

Figure 5.6 presents the major steps in the approach: (1) Knowledge Conversion - knowledge understandable for human beings may not be directly applicable in a learning model. Therefore, the first step is to convert different types of human or external knowledge to certain types of internal knowledge that can be used by gene selection algorithms. Assume we have L different external knowledge sources $\mathcal{K}_1^{ext}, \dots, \mathcal{K}_L^{ext}$. For the i th external knowledge, we can apply a conversion operator $c_i(\cdot)$ to convert the external knowledge \mathcal{K}_i^{ext} to the corresponding internal knowledge \mathcal{K}_i^{int} , and this allows us to formalize knowledge conversion with the following equation:

$$\mathcal{K}_i^{int} = c_i(\mathcal{K}_i^{ext}), i = 1, \dots, L \quad (5.6)$$

(2) Feature Ranking - assume K sets of internal knowledge $KNOW_1, \dots, KNOW_K$ is used to

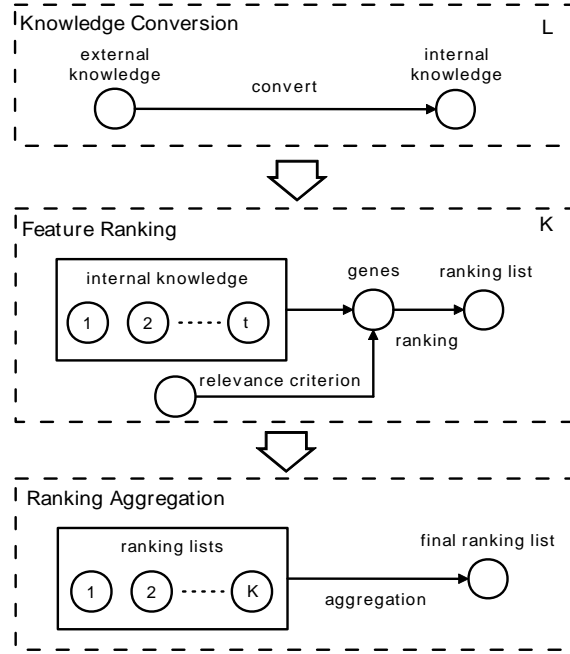


Figure 5.6: KOGS, a framework for knowledge-oriented multi-source gene selection.

rank genes, where $KNOW_i$ is defined as: $KNOW_i = \{\mathcal{H}_{i_1}^{int} \dots \mathcal{H}_{i_t}^{int}\}$. Let \mathcal{C}_i be a relevance criterion, $\mathbf{G} = \{g_1, \dots, g_M\}$ be a set of M genes, and $\mathcal{R}_i(\cdot)$ be a gene ranking function, the task of feature ranking is to use the internal knowledge with the given criterion to rank the relevance of genes in \mathbf{G} , and can be formulated as:

$$R_i^{rank} = \mathcal{R}(KNOW_i, \mathcal{C}_i, \mathbf{G}) \quad (5.7)$$

(3) Rank Aggregation - after obtained the K ranking lists, they need to be integrated to obtain a final ranking to estimate the relevance of the genes. Let $\mathcal{A}(\cdot)$ be an aggregating operator for ranking lists and \mathcal{C} be an aggregation criterion, we use $\mathcal{A}(\cdot)$ to aggregate the K ranking lists, which can be formulated as:

$$R_F^{rank} = \mathcal{A}(R_1^{rank}, \dots, R_K^{rank}, \mathcal{C}) \quad (5.8)$$

The final gene ranking list can be obtained by considering the ranking lists from all internal knowledge sets in either a supervised or an unsupervised fashion, depending upon how \mathcal{C}

is specified. The two systems are different in that (1) KOGS explicitly defines the concepts of external and internal knowledge, and organizes different types of knowledge into well defined categories, while no knowledge related concept is proposed in MSFS; (2) In the current work, the coefficient combination can be automatically learned, while this problem is not addressed in MSFS; and (3) KOGS is based on combining ranking lists, while the one in MSFS relies on combining sample similarity, which restricts the model flexibility. Our experimental results demonstrated the methods derived from KOGS is able to provide superior performance and select biologically relevant genes, which further approves the benefit of incorporating multiple knowledge source in the process of feature selection to effectively improve algorithms' performance of relevance detection.

Chapter 6

CONCLUSION

Feature selection aims to choose a subset of original features by removing irrelevant and redundant features. It is an important dimension reduction technique that is widely used in data mining. In my dissertation, I study the novel research problem of spectral feature selection, which select features from the perspective of sample similarity preserving. Spectral feature selection provides a powerful platform, which allows the unification of many existing supervised and unsupervised feature selection algorithm under a general framework to facilitate their joint study. It also provides a template, based on which novel feature selection algorithms with high effectiveness can be developed for solving difficult research problems. In this work, I developed three general spectral feature selection frameworks, including SPEC, MRSF, and MSFS. I analyzed their properties and showed their extrinsic nature of similarity preserving. I studied, through theoretical analysis, the connections among the proposed SPEC framework to many existing feature selection algorithms, such as Laplacian Score, Fisher Score, ReliefF, Trace Ratio Criterion, and HSIC. Therefore these algorithms are eccentrically similarity preserving based feature selection algorithms. I also conducted robustness analysis for SPEC, and studied how to derive new feature selection algorithms based on the framework. The proposed MSFS framework improves the existing feature selection algorithms based on sample similarity preserving by overcoming their common drawback in handling redundancy feature, which is harmful in both supervised and unsupervised learning. I also theoretically analyzed its connections to certain supervised and unsupervised learning models, including PCA, LDA, and SVM. And this provides further insights for all these learning models. I demonstrated that the proposed spectral feature selection frameworks can be conveniently extended to solve novel research

problems such as semi-supervised feature selection and multi-source feature selection. As illustrated by the extensive experimental studies, the proposed spectral feature selection frameworks achieved superior performance in various learning context, which concretely demonstrates their efficacy in solving difficult real problems.

I envision that the current development in scientific research will lead to the prevalence of extremely high dimensional data sets generated from the emerging high-throughput techniques and the availability of many useful knowledge sources resulting from collective work of state-of-the-art research. Hence my immediate future research work will aim to invent novel solutions and develop computational theories that will help scientists to keep up with rapid advance of new technologies. In my current research, I also notice that there is an obvious chasm between symbolic learning and numerical learning that prevents scientists from taking advantage of data and knowledge in a seamless way. Symbolic learning works well with knowledge and numerical learning works with data. I propose to study explanation-based feature selection that promises to bridge the current gap. The impact of this novel research direction will enable us to use existing knowledge to help narrow down the search space and explain the learning results by providing reasons why certain features are relevant. I elaborate below how I will build upon my research strengths in the area of data mining and bioinformatics to strengthen and expand my current interdisciplinary research program by investigating these intertwined research topics.

Extremely high dimensional learning: learning in extremely high ($>1M$) dimensional space. As high throughput techniques keep developing, many contemporary researches in scientific discovery generate data with extremely high dimensionality. For instance, the next-generation sequencing techniques on genetics can generate data with several giga features. Computation inherent in existing methods makes them hard to directly handle extremely high dimensional data, which raises the simultaneous challenges of computational power, statistical accuracy, and algorithmic stability. To address this challenge, I

will leverage my research experience in feature selection and feature extraction to develop efficient approaches for fast relevance identification and dimension reduction. Prior knowledge can also play important roles in this study, for instance, by providing effective ways to partition original feature space to small subspaces, which leads to significant reduction on search space and highly efficient parallel algorithms. I will apply the developed technique to assist genetic analysis based on the next-generation sequencing techniques.

Knowledge oriented sparse learning: fitting sparse learning models via utilizing multiple types of know-ledge. This direction extends my research work on multi-source feature selection. Sparse learning allows joint model fitting and features selection. Given multiple types of knowledge, I plan to study the novel problem of utilizing them to guide inference for improving learning performance. For instance, in microarray analysis, given gene regulatory network and gene ontology annotation, instead of converting them to a common representation, which may cause information loss, it is interesting to study how to simultaneously infer with both types of knowledge, for instance, via network dynamic analysis or function concordance analysis, to build accurate prediction models based on a compact gene set. One direct effect of utilizing existing knowledge in inference is that it can significantly increase the reliability of the output. Another effect of the technique is that it can reduce the experimental cost by requiring fewer samples. I will extend the developed technique to achieve low cost genetic analysis with high reliability.

Explanation-based feature selection (EBFS): feature selection via explaining training samples using concepts generalized from existing features and knowledge. In many real-world applications, the same phenomenon might be caused by disparate reasons. For example, in a cancer study, a certain phenotype may be related to mutations of either genes A or gene B in the same functional module M. And both gene A and gene B can cause the defect of M. Existing feature selection algorithm based on checking feature/class correlation may not work in this situation, due to the inconsistent (variable) expression pattern

of gene A and gene B across the cancerous samples¹. The generalization step in EBFS can effectively screen this variation by forming high-level concepts via using the ontology information obtained from annotation databases, such as GO. Another advantage of EBFS is that it can generate sensible explanations to show why the selected features are related. EBFS is related to the research of explanation-based learning (EBL) and relational learning. I will utilize the developed technique to develop robust genetic analysis approaches giving comprehensible outcomes. Besides genetic analysis, the developed technique can also be applied to policy making in health care, economics, or ecology.

¹For a cancerous sample, either gene A or gene B has abnormal expression, but not both.

BIBLIOGRAPHY

- [1] Stein Aerts, Diether Lambrechts, Sunit Maity, Peter Van Loo, Bert Coessens, Fredrik De Smet, Leon-Charles Tranchevent, Bart De Moor, Peter Marynen, Bassem Hassan, Peter Carmeliet, and Yves Moreau. Gene prioritization through genomic data fusion. *Nature Biotechnology*, 24:537–545, 2006.
- [2] Annalisa Appice, Michelangelo Ceci, Simon Rawles, and Peter Flach. Redundant feature elimination for multi-class problems. In *Proceedings of the twenty-first international conference on Machine learning (ICML)*, 2004.
- [3] Ramon Aragues, Chris Sander, and Baldo Oliva. Predicting cancer involvement of genes from heterogeneous data. *BMC Bioinformatics*, 9:172, 2008.
- [4] A. Argyriou, T. Evgeniou, and Massimiliano Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.
- [5] K. Bae and B. K. Mallick. Gene selection using a two-level hierarchical bayesian model. *BIOINFORMATICS*, 20:3423–3430, 2004.
- [6] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [7] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [8] Evelyn Camon, Michele Magrane, Daniel Barrell, Vivian Lee, Emily Dimmer, John Maslen, David Binns, Nicola Harte, Rodrigo Lopez, and Rolf Apweiler. The gene ontology annotation (goa) database: sharing knowledge in uniprot with gene ontology. *Nucleic Acids Research*, 32:262–266, 2004.
- [9] G. C. Cawley and N. L. C. Talbot. Gene selection in cancer classification using sparse logistic regression with bayesian regularization. *BIOINFORMATICS*, 22:2348–2355, 2006.
- [10] G. C. Cawley, N. L. C. Talbot, and M. Girolami. Sparse multinomial logistic regression via bayesian l1 regularisation. In *In Advances in Neural Information Processing Systems*, 2007.

- [11] O. Chapelle, B. Scholkopf, and A. Zien. *Semi-Supervised Learning*. MIT Press, 2006.
- [12] F. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.
- [13] M. Dash, K. Choi, P. Scheuermann, and H. Liu. Feature selection for clustering – a filter solution. In *Proceedings of International Conference on Data Mining (ICDM)*, 2002.
- [14] A. d’Aspremont, F. Bach, and L. El Ghaoui. Optimal solutions for sparse principal component analysis. Technical report, Princeton University, 2007.
- [15] James W. Demmel. *Applied Numerical Linear Algebra*. SIAM, 1997.
- [16] C. Ding and H. Peng. Minimum redundancy feature selection from microarray gene expression data. In *Proceedings of the Computational Systems Bioinformatics conference (CSB’03)*, pages 523–529, 2003.
- [17] D. Donoho. Formost large underdetermined systems of linear equations, the minimal l_1 -norm solution is also the sparsest solution. *Comm. Pure Appl. Math.*, 59(7):907–934, 2006.
- [18] R. Duangsoithong. Relevant and redundant feature analysis with ensemble classification. In *Proceedings of the seventh International Conference on Advances in Pattern Recognition (ICAPR ’09)*., 2009.
- [19] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley & Sons, New York, 2 edition, 2001.
- [20] Jennifer G. Dy and Carla E. Brodley. Feature selection for unsupervised learning. *J. Mach. Learn. Res.*, 5:845–889, 2004.
- [21] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–49, 2004.
- [22] George Forman. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3:1289–1305, 2003.

- [23] Evgeniy Gabrilovich and Shaul Markovitch. Text categorization with many redundant features: using aggressive feature selection to make svms competitive with c4.5. In *Proceedings of the twenty-first international conference on Machine learning (ICML'04)*, 2004.
- [24] Cicek Gerçel-Taylor, David L. Doering, Fredric B Kraemer, and Douglas D. Taylor. Aberrations in normal systemic lipid metabolism in ovarian cancer patients. *Gynecologic Oncology*, 60:35–41, 1996.
- [25] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, third edition, 1996.
- [26] R. Gonzalez and R. Woods. *Digital Image Processing*. Addison-Wesley, 2nd edition, 1993.
- [27] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *Proc. Intl. Conf. on Algorithmic Learning Theory*, pages 63–78, 2005.
- [28] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.
- [29] M.A. Hall. *Correlation Based Feature Selection for Machine Learning*. PhD thesis, University of Waikato, Dept. of Computer Science, 1999.
- [30] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001.
- [31] X. He, D. Cai, and P. Niyogi. Laplacian score for feature selection. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, Cambridge, MA, 2005. MIT Press.
- [32] J. C. Huang, T. Babak, T. W. Corson, G. Chua, S. Khan, B. L. Gallie, T. R. Hughes, B. J. Blencowe, B. J. Frey, and Q. D. Morris. Using expression profiling data to identify human microrna targets. *NATURE METHODS*, 4:1045–1049, 2007.
- [33] Kaizhu Huang, Irwin King, and Michael R. Lyu. Direct zero-norm optimization for feature selection. In *Proceeding of The Eighth IEEE International Conference on Data Mining*, 2008.

- [34] Ingenuity-Systems. Ingenuity pathways analysis. <http://www.ingenuity.com>.
- [35] Inaki Inza, Pedro Larranaga, Rosa Blanco, and Antonio J. Cerrolaza. Filter versus wrapper gene selection approaches in dna microarray domains. *Artificial Intelligence in Medicine*, 31:91–103, 2004.
- [36] A. Jain and D. Zongker. Feature selection: Evaluation, application, and small sample performance. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(2):153–158, 1997.
- [37] M. Kanehisa and S. Goto. Kegg: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Res*, 28:27–30, 2000.
- [38] Fumiaki Katagiri and Jane Glazebrook. Overview of mrna expression profiling using dna microarrays. *Current Protocols in Molecular Biology*, 22.4:s85, 2009.
- [39] M.J. Kearns and U.V. Vazirani. *An Introduction to Computational Learning Theory*. The MIT Press, 1994.
- [40] Y. B. Kim and J. Gao. Unsupervised gene selection for high dimensional data. In *Proc. Sixth IEEE Symposium on BioInformatics and BioEngineering BIBE 2006*, pages 227–234, 16–18 Oct. 2006.
- [41] K. Kira and L.A. Rendell. A practical approach to feature selection. In Sleeman and P. Edwards, editors, *Proceedings of the Ninth International Conference on Machine Learning (ICML-92)*, pages 249–256. Morgan Kaufmann, 1992.
- [42] I. Kononenko. Estimating attributes : Analysis and extension of RELIEF. In F. Bergadano and L. De Raedt, editors, *Proceedings of the European Conference on Machine Learning, April 6-8*, pages 171–182, Catania, Italy, 1994. Berlin: Springer-Verlag.
- [43] Carmen Lai, Marcel J T Reinders, Laura J van't Veer, and Lodewyk F A Wessels. A comparison of univariate and multivariate gene selection techniques for classification of cancer datasets. *BMC Bioinformatics*, 7:235, 2006.
- [44] Gert R. G. Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I. Jordan. Learning the kernel matrix with semidefinite programming. *J. Mach. Learn. Res.*, 5:27–72, 2004.

- [45] W. Lee, S. J. Stolfo, and K. W. Mok. Adaptive intrusion detection: A data mining approach. *AI Review*, 14(6):533 – 567, 2000.
- [46] Caiyan Li and Hongzhe Li. Network-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics*, 24(9):1175–1182, May 2008.
- [47] T. Li, C. Zhang, and M. Ogihara. A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *BIOINFORMATICS*, 20:2429–2437, 2004.
- [48] J.G. Liao and Khew-Voon Chin. Logistic regression for disease classification using microarray data: model selection in a large p and small n case. *BIOINFORMATICS*, 23:1945–1951, 2007.
- [49] H. Liu and H. Motoda, editors. *Computational Methods of Feature Selection*. Chapman and Hall/CRC Press, 2007.
- [50] H. Liu and L. Yu. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, 17:491–502, 2005.
- [51] Huan Liu and Zheng Zhao. *Encyclopedia of Complexity and Systems Science*, chapter Manipulating Data and Dimension Reduction Methods, pages 5348–5359. Springer, 2009.
- [52] Huiqing Liu, Jinyan Li, and Limsoon Wong. A comparative study on feature selection and classification methods using gene expression profiles and proteomic patterns. *Genome Inform*, 13:51–60, 2002.
- [53] Jun Liu, Shuiwang Ji, and Jieping Ye. Multi-task feature learning via efficient $l_{2,1}$ -norm minimization. In *The Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI 2009)*, 2009.
- [54] J. Lu, G. Getz, E. A. Miska, E. Alvarez-Saavedra, J. Lamb, D. Peck, A. Sweet-Cordero, B. L. Ebert, R. H. Mak, A. Ferrando, J. R. Downing, T. Jacks, H. R. Horvitz, and T. R. Golub. MicroRNA expression profiles classify human cancers. *Nature*, 435:834–838, 2005.

- [55] Bin Luo, Richard C. Wilson, and Edwin R. Hancock. Spectral embedding of graphs. *Pattern Recognition*, 36:2213–2230, 2003.
- [56] Shuangge Ma. Empirical study of supervised gene screening. *BMC Bioinformatics*, 7:537, 2006.
- [57] Shuangge Ma and Jian Huang. Penalized feature selection and classification in bioinformatics. *Brief Bioinform*, 9(5):392–403, Sep 2008.
- [58] P.C. Mahalanobis. On the generalized distance in statistics. *Proceedings of the National Institute of Science of India*, 12:49–55, 1936.
- [59] Carl Murie, Owen Woody, Anna Lee, and Robert Nadon. Comparison of small n statistical tests of differential expression applied to microarrays. *BMC Bioinformatics*, 10(1):45, Feb 2009.
- [60] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *The 14th Advances in Neural Information Processing Systems (NIPS)*, 2001.
- [61] A. Y. Ng. Feature selection, l_1 vs. l_2 regularization, and rotational invariance. In *ICML '04: Twenty-first international conference on Machine learning*. ACM Press, 2004.
- [62] Feiping Nie, Feiping Nie, Shiming Xiang, Yangqing Jia, Changshui Zhang, and Shuicheng Yan. Trace ratio criterion for feature selection. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2008.
- [63] G. Obozinski, M. J. Wainwright, and M. I. Jordan. Highdimensional union support recovery in multivariate regression. In *Neural Information Processing Systems*, 2008.
- [64] John H. Phan, Qiqin Yi Goen, Andrew N. Young, and May D. Wang. Improving the efficiency of biomarker identification using biological knowledge. In *Pacific Symposium on Biocomputing*, pages 427–38, 2009.
- [65] C. D. Pitta, L. Tombolan, M. Campo Dell’Orto, and B. Accordi. A leukemia-enriched cDNA microarray platform identifies new transcripts with relevance to the biology of pediatric acute lymphoblastic leukemia. *Haematologica*, 90:890–898, 2005.

- [66] Jianlong Qi and Jian Tang. Gene ontology driven feature selection from microarray gene expression data. In *Computational Intelligence and Bioinformatics and Computational Biology*, 2006.
- [67] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
- [68] Richard J. Radke. A matlab implementation of the implicitly restarted arnoldi method for solving large-scale eigenvalue problems. Master's thesis, Department of Computational and Applied Mathematics, Rice University, 1996.
- [69] Sarunas J. Raudys and Anil K. Jain. Small sample size effects in statistical pattern recognition: Recommendations for practitioners. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13:252–264, 1991.
- [70] V. Roth and B. Fischer. The group-lasso for generalized linear models: Uniqueness of solutions and efficient algorithms. In *International conference on Machine learning*, page 848855, 2008.
- [71] Yvan Saeys, Thomas Abeel, and Yves Van de Peer. Robust feature selection using ensemble feature selection techniques. In *Proceeding of European Conference on Machine Learning (ECML)*, 2008.
- [72] Yvan Saeys, Iaki Inza, and Pedro Larraaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, Oct 2007.
- [73] L.K. Saul, K. Q. Weinberger, F. Sha, J. Ham, and D. D. Lee. *Spectral Methods for Dimensionality Reduction*, chapter 16, pages 279–293. The MIT Press, 2006.
- [74] B. Scholköpfung and A. J. Smola. *Learning with Kernels*. The MIT Press, 2002.
- [75] Christine M.E. Schueller, Andreas Fritz, Eduardo Torres Schumann, Karsten Wenger, Kaj Albermann, George A. Komatsoulis, Peter A. Covitz, Lawrence W. Wright, and Frank Hartel. Towards a comprehensive catalog of gene-disease and gene-drug relationships in cancer. Technical report, National Cancer Institute, 2005.
- [76] M.W. Seeger. Bayesian inference and optimal design for the sparse linear model. *Journal of Machine Learning Research*, 9:759–813, 2008.

- [77] Blake Shaw and Tony Jebara. Structure preserving embedding. In *Proceeding of the 26th International Conference on Machine Learning (ICML 2009)*, 2009.
- [78] J. Shi and J. Malik. Normalized cuts and image segmentation. In *IEEE Conf. Computer Vision and Pattern Recognition(CVPR)*, 1997.
- [79] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.
- [80] M. R. Sikonja and I. Kononenko. Theoretical and empirical analysis of Relief and ReliefF. *Machine Learning*, 53:23–69, 2003.
- [81] C. Sima and E. R. Dougherty. What should be expected from feature selection in small-sample settings. *Bioinformatics*, 22:2430–2436, 2006.
- [82] T. J. Slaga, A. J. Klein-Szanto, L. L. Triplett, L. P. Yotti, and J. E. Trosko. Skin tumor-promoting activity of benzoyl peroxide, a widely used free radical-generating compound. *Science*, 213:1023–1025, 1981.
- [83] A.J. Smola and I.R. Kondor. Kernels and regularization on graphs. In *Proceedings of the Annual Conference on Computational Learning Theory (COLT)*, 2003.
- [84] L. Song, A. Smola, A. Gretton, J. Bedo, and K. Borgwardt. Feature selection via dependence maximization. *Journal of Machine Learning Research*, 2007.
- [85] L. Song, A. Smola, A. Gretton, K. Borgwardt, and J. Bedo. Supervised feature selection via dependence estimation. In *International Conference on Machine Learning*, 2007.
- [86] Shireesh Srivastava, Linxia Zhang, Rong Jin, and Christina Chan. A novel method incorporating gene ontology information for unsupervised clustering and feature selection. *PLoS ONE*, 3(12):e3860, 2008.
- [87] Chris Stark, Bobby-Joe Breitkreutz, Teresa Reguly, Lorrie Boucher, Ashton Breitkreutz, and Mike Tyers. Biogrid: A general repository for interaction datasets. *Nucleic Acids Res*, 34:535–539, 2006.
- [88] Aravind Subramanian, Pablo Tamayo, Vamsi K. Mootha, Sayan Mukherjee, Benjamin L. Ebert, Michael A. Gillette, Amanda Paulovich, Scott L. Pomeroy, Todd R.

- Golub, Eric S. Lander, and Jill P. Mesirov. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *PNAS*, 102:15545–15550, 2005.
- [89] Liang Sun, Shuiwang Ji, and Jieping Ye. A least squares formulation for a class of generalized eigenvalue problems in machine learning. In *Proceedings of the 26 th International Conference on Machine Learning*, 2009.
- [90] Y. Sun, C. F. Babbs, and E. J. Delp. A comparison of feature selection methods for the detection of breast cancers in mammograms: adaptive sequential floating search vs. genetic algorithm. *Conf Proc IEEE Eng Med Biol Soc*, 6:6532–6535, 2005.
- [91] J.A.K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3):1370–4621, 1999.
- [92] Michael D Swartz, Robert K Yu, and Sanjay Shete. Finding factors influencing risk: Comparing bayesian stochastic search and standard variable selection methods applied to logistic regression models of cases and controls. *Stat Med*, 27(29):6158–6174, Dec 2008.
- [93] D. L. Swets and J. J. Weng. Efficient content-based image retrieval using automatic feature selection. In *IEEE International Symposium On Computer Vision*, pages 85–90, 1995.
- [94] J. Tenenbaum, V. de Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.
- [95] U. von Luxburg. A tutorial on spectral clustering. Technical report, Max Planck Institute for Biological Cybernetics, 2007.
- [96] J. Weston, A. Elisseeff, B. Schoelkopf, and M. Tipping. Use of the zero norm with linear models and kernel methods. *Journal of Machine Learning Research*, 3:1439–1461, 2003.
- [97] L. Wolf and A. Shashua. Feature selection for unsupervised and supervised inference: the emergence of sparsity in a weighted-based approach. *Journal of Machine Learning Research*, 6:1855–1887, 2005.

- [98] Zenglin Xu, Rong Jin, Jieping Ye, Michael R. Lyu, and Irwin King. Discriminative semi-supervised feature selection via manifold regularization. In *IJCAI' 09: Proceedings of the 21th International Joint Conference on Artificial Intelligence*, 2009.
- [99] Jieping Ye, Jianhui Chen, Ravi Janardan, and Sudhir Kumar. Developmental stage annotation of drosophila gene expression pattern images via an entire solution path for lda. *ACM Transactions on Knowledge Discovery from Data, special issue on Bioinformatics*, To appear, 2007.
- [100] L. Yu and H. Liu. Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research (JMLR)*, 5(Oct):1205–1224, 2004.
- [101] Lei Yu, Chris Ding, and Steven Loscalzo. Stable feature selection via dense feature groups. In *Proceedings of The 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-08)*, 2008.
- [102] Lei Yu and Huan Liu. Efficient feature selection via analysis of relevance and redundancy. *J. Mach. Learn. Res.*, 5:1205–1224, 2004.
- [103] H. Zhang, J. Ahn, X. Lin, and C. Park. Gene selection using support vector machines with non-convex penalty. *BIOINFORMATICS*, 22:88–95, 2005.
- [104] Tong Zhang and Rie Ando. Analysis of spectral kernel design based semi-supervised learning. In *Advances in Neural Information Processing Systems 18*, pages 1601–1608, 2006.
- [105] Yi Zhang, Chris Ding, and Tao Li. Gene selection algorithm by combining relief and mrmr. *BMC Genomics*, 9:S27, 2008.
- [106] Z. Zhao and H. Liu. Semi-supervised feature selection via spectral analysis. Technical Report TR-06-022, Computer Science and Engineering Department, 2006.
- [107] Z. Zhao and H. Liu. Semi-supervised feature selection via spectral analysis. In *SIAM International Conference on Data Mining*, 2007.
- [108] Z. Zhao and H. Liu. Spectral feature selection for supervised and unsupervised learning. In *International Conference on Machine Learning (ICML)*, 2007.

- [109] Z. Zhao, J. Wang, H. Liu, J. Ye, and Y. Chang. Identifying biologically relevant genes via multiple heterogeneous data sources. In *The Fourteenth ACM SIGKDD International Conference On Knowledge Discovery and Data Mining (SIGKDD 2008)*, 2008.
- [110] Zheng Zhao and Huan Liu. Semi-supervised feature selection via spectral analysis. In *Proceedings of SIAM International Conference on Data Mining (SDM)*, 2007.
- [111] Zheng Zhao and Huan Liu. Multi-source feature selection via geometry-dependent covariance analysis. In *Journal of Machine Learning Research, Workshop and Conference Proceedings Volume 4: New challenges for feature selection in data mining and knowledge discovery*, volume 4, pages 36–47, 2008.
- [112] Zheng Zhao, Jiangxin Wang, Shashvata Sharma, Nitin Agarwal, Huan Liu, and Yung Chang. An integrative approach to identifying biologically relevant genes. In *Proceedings of SIAM International Conference on Data Mining (SDM)*, 2010.
- [113] Zheng Zhao, Lei Wang, and Huan Liu. Efficient spectral feature selection with minimum redundancy. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*, 2010.
- [114] D. Zhou and C. Burges. Spectral clustering and transductive learning with multiple views. In *in Proceedings of the 24th International Conference on Machine Learning*, 2007.
- [115] Ji Zhu, Saharon Rosset, Trevor Hastie, and Rob Tibshirani. 1-norm support vector machines. In *Advances in Neural Information Processing Systems 16*, 2003.
- [116] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. Technical report, Statistics Department, Stanford University, 2004.