Robust feature selection for high-dimensional and small-sized gene expression data

Yang, Feng

2011

Yang, F. (2011). Robust feature selection for high-dimensional and small-sized gene expression data. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/48641>

<https://doi.org/10.32657/10356/48641>

**Robust Feature Selection for High-Dimensional and
Small-Sized Gene Expression Data**

Yang Feng

School of Electrical & Electronic Engineering

A Thesis Submitted To The Nanyang Technological University
In Fulfillment of the Requirement for the Degree Of
Doctor of Philosophy

2011

Acknowledgements

I would like to express my sincere gratitude to all those who made it possible for me to complete this thesis. First and foremost, I am deeply indebted to my supervisor Dr. Mao Kezhi, Associate Professor of School of Electrical and Electronic Engineering, Nanyang Technological University (NTU), who is a man of full wisdom and immense knowledge. His continuous support and guidance, stimulating ideas and suggestions, especially his patience and encouragement during the tough times in my PhD pursuit, have been of utmost importance for my PhD study and for the accomplishment of this thesis.

I would like to acknowledge Nanyang Technological University for the scholarship supporting my research and life in Singapore, and for the travel funds to attend overseas conference.

I thank all my fellow labmates and friends in the Computer Vision Lab, Biomedical Electronics Lab and BIRC (Bioinformatics Research Center) for their stimulating discussions concerning my research topic. Especially, I want to appreciate my senior Dr. Tang Wenying, now a research fellow at NTU, who gave me great help and many valuable advices for my study. The technical staffs in the labs, Ms. Yang Xin, Mr. Yap SingLoon, Mr. Sow PeckHeng, Mr. Yuen SienHuan and Mr. Tan SingYau,

deserve thanks for their technical assistance.

I also would like to express my thanks to Mr. Li Guoqi who is one of my best friends. I enjoy the discussion and even debate process with him on some research problems, from which I benefit a lot.

Last but not the least, my deep gratitude belongs to my parents and my elder sister who are always there when I need them.

Table of contents

Acknowledgements

Summary ix

List of Figures xii

List of Tables xviii

Notation Table xix

Abbreviation Table xx

1 Introduction 1

1.1 Motivation 1

1.2 Objectives 4

1.3 Overview of the Thesis 4

2 Backgrounds and Preliminaries 9

2.1 An Overview of Pattern Recognition Systems 9

2.2 Pattern Classification 16

2.3 Classification Techniques 19

2.3.1	Classification Techniques Based on Similarity	20
2.3.2	Classification Techniques Based on Probability Estimation . .	22
2.3.3	Classification Techniques Based on Optimization of Error Cri- teria	24
2.3.4	Classification Techniques Based on Logic	30
2.4	Classification Performance Evaluation	33
2.4.1	Error Rate and Accuracy	35
2.4.2	Sensitivity and Specificity	36
2.4.3	ROC and AUC	37
2.4.4	Validation Techniques for Performance Evaluation	39
2.5	Dimension Reduction	43
2.5.1	Feature Selection	43
2.5.2	Feature Extraction	44
2.6	Data Analyzed in Thesis	45
2.6.1	A Brief Introduction to Microarrays	45
2.6.2	Micriarray Gene Expression Data Sets	47
2.6.3	Formal Representation of Data Sets	49
3	Feature Selection and the Robustness Issue	51
3.1	Feature Selection: What, Why and How?	51
3.2	Components in Feature Selection Procedure	54
3.2.1	Subset Evaluation Criteria	54
3.2.2	Subset Search Strategies	56
3.2.3	Stopping Criteria	61

3.2.4	Result Validation	62
3.3	Feature Selection Techniques	62
3.3.1	Filter Methods	63
3.3.2	Wrapper Methods	67
3.3.3	Embedded Methods	69
3.4	Robustness Issue in Feature Selection	71
3.5	Evaluation of Robustness	76
3.5.1	Similarity	76
3.5.2	Robustness Measures Based on Similarity	78
4	Improving Robustness of Feature Ranking by Multi-Criterion Com-	
	bination with Novel Feature Importance Transformation	80
4.1	Introduction	80
4.2	Illustration of Inconsistency Problems in Feature Selection	82
4.3	Basis Criteria for Multi-Criterion Combination	86
4.3.1	Fisher's ratio	87
4.3.2	ADC	87
4.3.3	Relief	88
4.3.4	AW-SVM	89
4.4	Basic Combination Methods	89
4.4.1	Score-Based Combination	90
4.4.2	Ranking-Based Combination	91
4.5	Problems in Existing Methods	93
4.5.1	Domination Problem in Score Combination	93

4.5.2	Score Inaccuracy Problem under Small Sample Size	94
4.5.3	Score Saturation Problem in PIMP	95
4.6	Resampling and Permutation-Based Feature Importance Transforma- tion	97
4.7	Experimental Studies	99
4.7.1	Robustness Measure for Feature Ranking	99
4.7.2	Experimental Results and Discussions	100
4.8	Summary of Chapter	103
5	Robust Feature Subset Selection Based on Fusion of Multiple Cri- teria	104
5.1	Introduction	104
5.2	An Intensive Consideration of Integrating Multiple Criteria	105
5.3	Multi-Criterion Fusion-Based Recursive Feature Elimination (MCF- RFE) Algorithm	107
5.4	An Improved Measure for Evaluating Robustness of Feature Selection Algorithms	110
5.5	Experimental Studies	112
5.5.1	Experimental Setup	112
5.5.2	Comparative Study of MCF-RFE with Basis Criteria and SVM- RFE	113
5.5.3	Comparative Study of MCF-RFE with Bagging-Based Ensem- ble Technique (BBET)	115
5.6	Summary of Chapter	121

6 Regularizing Linear Discriminant Analysis with Emphasis on Minority Class for Feature Subset Selection	128
6.1 Introduction	128
6.2 Regularized LDA-Based Techniques for Feature Subset Evaluation . .	130
6.2.1 From Fisher's Ratio to LDA for Feature Subset Evaluation . .	130
6.2.2 Conventional Forms of Regularization to LDA	133
6.2.3 A New Regularization Giving More Emphasis to Minority Class	135
6.3 Incremental Implementation of LDA-Based Feature Subset Selection .	137
6.4 Experimental Studies	142
6.4.1 Investigation of the Influence of Parameter γ in Class Emphasis	142
6.4.2 Determination of the Parameter ρ for Diagonal Emphasis . . .	147
6.4.3 Comparative Studies of Different Forms of Regularization . . .	151
6.5 Summary of Chapter	157
7 A Framework of Reducing Feature Redundancy in Ranking-Based Feature Selection	161
7.1 Feature Redundancy: What Is Its Role?	161
7.2 Review of Related Work	164
7.3 Impact of Feature Redundancy: A Case Study	166
7.4 A Feature Redundancy Reduction Framework Based on Orthogonal Decomposition and Sequential Forward Search	170
7.4.1 Feature Evaluation Based on Orthogonal Decomposition . . .	171
7.4.2 The Procedure to Reduce Feature Redundancy through Or- thogonal Decomposition and Forward Search	173

7.4.3	Saturation Problems in the OD-Based Feature Redundancy	
	Reduction Framework	174
7.5	Relaxed Orthogonal Decomposition-Based Framework	177
7.6	Experimental Studies	182
7.7	Summary of Chapter	184
8	Conclusions and Future Work	189
8.1	Thesis Conclusions	189
8.2	Future Work Recommendations	192
	Author's Publications	194
	Bibliography	196

Summary

One important issue in constructing a pattern recognition system is feature selection. The goal of feature selection, including feature ranking and feature subset selection, is to identify target-relevant features. When applied to high-dimensional and small-sized (HDSS) data, e.g. microarray gene expression data, the commonly used feature selection algorithms encounter problems such as over-sensitivity to variations in training data, i.e. robustness issue. The aim of this thesis is to address the robustness issue of feature selection for the HDSS data.

Firstly, a novel criteria normalization algorithm has been proposed for multi-criterion combination to improve robustness of feature ranking. Traditional feature ranking criteria are apt to produce inconsistent ranking results even with slight perturbations in training samples when applied to HDSS data. A widely used strategy for solving the inconsistency problems is multi-criterion combination. But one crucial problem in multi-criterion combination is how to normalize feature scores from different criteria. In the thesis, a new feature importance transformation algorithm based on resampling and permutation is proposed for score normalization. Experimental studies on four popular gene expression data sets show that the multi-criterion combination based on the proposed score normalization produces

gene rankings with improved robustness.

Secondly, a multi-criterion fusion-based recursive feature elimination (MCF-RFE) algorithm has been developed with the goal of improving both classification performance and robustness of feature subset selection. Feature subset selection often aims to select a compact subset of features to build a pattern classifier with reduced complexity. From the perspective of pattern analysis, producing stable or robust solution is also a desired property of a feature subset selection algorithm. In the thesis, we analyze the robustness issue existing in feature subset selection for HDSS gene expression data, and propose the MCF-RFE algorithm. Experimental studies on five gene expression data sets show that the MCF-RFE algorithm outperforms the commonly used benchmark feature selection algorithm SVM-RFE.

Thirdly, a new regularized linear discriminant analysis (LDA) based algorithm has been proposed for robust feature selection of HDSS data. When applied to gene expression data which usually have high dimensionality, small sample size and class imbalance (i.e. great discrepancy in the number of samples between classes), LDA-based feature selection encounters problems such as singularity of scatter matrix, overfitting, overwhelming and prohibitive computational complexity. In the thesis, we propose a new regularization technique giving more emphasis to minority class, with the expectation of improving overall performance by alleviating overwhelming of majority class to minority class as well as overfitting in minority class. In addition, an incremental implementation of LDA-based feature selection has been developed to reduce computational overhead. Comparative studies on five gene microarray problems show that LDA with the new regularization can produce gene subsets with excellent performance in both classification and robustness.

Fourthly, a framework of reducing feature redundancy has been proposed. In order to improve compactness of feature subsets, the usual practice is to utmostly remove redundancy. But this strategy does not necessarily help in some problems such as HDSS gene expression data analysis. We argue that a moderate degree of feature redundancy should be retained in feature subset for improving both classification and robustness. The proposed framework based on this argument is flexible and can be applied to any feature ranking algorithms. We have implemented the framework with the popular feature ranking algorithm—Fisher’s ratio, and conducted experiments on five gene expression data sets. The experimental studies show that the proposed framework produces feature subsets with improved classification performance and good performance in robustness.

List of Figures

2.1	Three paradigms of pattern recognition	12
2.2	Flow chart of a typical pattern recognition system	15
2.3	Detailed block diagram of a pattern classification system	18
2.4	An example of decision tree	31
2.5	Illustration of a binary-class classification problem and the ROC curve	38
3.1	Four key steps of feature selection	53
4.1	Average number of common genes vs. number of samples removed when a subset of 10 genes is selected	84
4.2	Score-based multi-criterion combination	90
4.3	Ranking-based multi-criterion combination	92
4.4	Scores generated by each basis criterion on Colon data: (a) Original scores and (b) Scores after normalization.	93
4.5	Scores of top 10 genes on Colon data: (a) By AW-SVM and (b) By ADC.	95
4.6	PIMP scores from each basis criterion	96
4.7	Score distribution	98
4.8	RPI scores from each basis criterion	99

4.9	Experimental results on Colon data: (a) Robustness and (b) Classification error rate.	101
4.10	Experimental results on Lymphoma data: (a) Robustness and (b) Classification error rate.	101
4.11	Experimental results on Leukemia data: (a) Robustness and (b) Classification error rate.	102
4.12	Experimental results on Prostate data: (a) Robustness and (b) Classification error rate.	102
5.1	The procedure of MCF-RFE	109
5.2	Score & ranking-based multi-criterion fusion	109
5.3	Computation of sum of correlation	111
5.4	Robustness Index when 10 samples are removed in gene selection . . .	112
5.5	Performance comparisons on Colon data: (a) Classification error. (b) Standard deviation of error estimation. (c) AUC and (d) Robustness.	116
5.6	Performance comparisons on Leukemia data: (a) Classification error. (b) Standard deviation of error estimation. (c) AUC and (d) Robustness.	117
5.7	Performance comparisons on Prostate data: (a) Classification error. (b) Standard deviation of error estimation. (c) AUC and (d) Robustness.	118
5.8	Performance comparisons on CNS data: (a) Classification error. (b) Standard deviation of error estimation. (c) AUC and (d) Robustness.	119

5.9	Performance comparisons on DLBCL data: (a) Classification error. (b) Standard deviation of error estimation. (c) AUC and (d) Robustness.	120
5.10	MCF-RFE and BBET comparisons on Colon data: (a) Classification error. (b) Standard deviation of error estimation. (c) AUC and (d) Robustness.	122
5.11	MCF-RFE and BBET comparisons on Leukemia data: (a) Classification error. (b) Standard deviation of error estimation. (c) AUC and (d) Robustness.	123
5.12	MCF-RFE and BBET comparisons on Prostate data: (a) Classification error. (b) Standard deviation of error estimation. (c) AUC and (d) Robustness.	124
5.13	MCF-RFE and BBET comparisons on CNS data: (a) Classification error. (b) Standard deviation of error estimation. (c) AUC and (d) Robustness.	125
5.14	MCF-RFE and BBET comparisons on DLBCL data: (a) Classification error. (b) Standard deviation of error estimation. (c) AUC and (d) Robustness.	126
6.1	Classification and robustness performance under different values of γ for class emphasis on DLBCL data: (a) Classification and (b) Robustness.	143

6.2	Classification and robustness performance under different values of γ for class emphasis on Prostate data: (a) Classification and (b) Robustness.	144
6.3	Classification and robustness performance under different values of γ for class emphasis on Colon data: (a) Classification and (b) Robustness.	144
6.4	Classification and robustness performance under different values of γ for class emphasis on Leukemia data: (a) Classification and (b) Robustness.	145
6.5	Classification and robustness performance under different values of γ for class emphasis on CNS data: (a) Classification and (b) Robustness.	145
6.6	Classification and robustness performance under different values of ρ for diagonal emphasis on DLBCL data: (a) Classification and (b) Robustness.	148
6.7	Classification and robustness performance under different values of ρ for diagonal emphasis on Prostate data: (a) Classification and (b) Robustness.	149
6.8	Classification and robustness performance under different values of ρ for diagonal emphasis on Colon data: (a) Classification and (b) Robustness.	149
6.9	Classification and robustness performance under different values of ρ for diagonal emphasis on Leukemia data: (a) Classification and (b) Robustness.	150

6.10	Classification and robustness performance under different values of ρ for diagonal emphasis on CNS data: (a) Classification and (b) Ro- bustness.	150
6.11	Performance comparisons of different forms of regularization on Colon data: (a) Classification error rate. (b) AUC and (c) Robustness. . . .	152
6.12	Performance comparisons of different forms of regularization on Leukemia data: (a) Classification error rate. (b) AUC and (c) Robustness. . . .	153
6.13	Performance comparisons of different forms of regularization on DL- BCL data: (a) Classification error rate. (b) AUC and (c) Robustness.	154
6.14	Performance comparisons of different forms of regularization on CNS data: (a) Classification error rate. (b) AUC and (c) Robustness. . . .	155
6.15	Performance comparisons of different forms of regularization on Prostate data: (a) Classification error rate. (b) AUC and (c) Robustness. . . .	156
6.16	Classification accuracy of single class on Colon data: (a) True positive rate (TPR) and (b) True negative rate (TNR).	157
6.17	Classification accuracy of single class on Leukemia data: (a) True positive rate (TPR) and (b) True negative rate (TNR).	158
6.18	Classification accuracy of single class on DLBCL data: (a) True pos- itive rate (TPR) and (b) True negative rate (TNR).	158
6.19	Classification accuracy of single class on CNS data: (a) True positive rate (TPR) and (b) True negative rate (TNR).	159
6.20	Classification accuracy of single class on Prostate data: (a) True pos- itive rate (TPR) and (b) True negative rate (TNR).	159

7.1	Performance of different degrees of redundancy reduction: (a) Classification error rate and (b) Robustness.	169
7.2	Illustration of the process when evaluating a candidate feature in the OD-based framework: (a) Evaluating the second feature. (b) Evaluating the third feature and (c) The space relationship of the first three selected features.	175
7.3	The accumulated Fisher's ratio scores from the OD-based framework.	177
7.4	Illustration of the relationships when evaluating the first three selected features X'_1, X'_2, X'_3 in the ROD-based framework.	179
7.5	A comparison of accumulated Fisher's ratio scores from the OD-based framework and ROD-based framework.	180
7.6	(a) Comparison of the correlation level and (b) Comparison of robustness	181
7.7	Performance comparisons on Colon data: (a) Classification error. (b) AUC and (c) Robustness.	184
7.8	Performance comparisons on Leukemia data: (a) Classification error. (b) AUC and (c) Robustness.	185
7.9	Performance comparisons on DLBCL data: (a) Classification error. (b) AUC and (c) Robustness.	186
7.10	Performance comparisons on CNS data: (a) Classification error. (b) AUC and (c) Robustness.	187
7.11	Performance comparisons on Prostate data: (a) Classification error. (b) AUC and (c) Robustness.	188

List of Tables

- 1.1 Top 10 genes selected by Fisher’s ratio on three different batches of
training data generated by sampling with replacement on full CNS
data set 3

- 2.1 Contingency table 35
- 2.2 Data sets description 49

- 4.1 Top 10 genes selected by five different feature selection criteria on
CNS data with all 60 samples as training data 85

- 6.1 The forms of regularization 151

Notation Table

Notation	Description
$\mathcal{X}$	The sample space
$\mathcal{Y}$	The label space
$\mathcal{G}$	A mapping function, $\mathcal{G} : \mathcal{X} \rightarrow \mathcal{Y}$
$\mathbf{x}$	A sample from the sample space, $\mathbf{x} \in \mathcal{X}$
y	The label of a sample, $y \in \mathcal{Y}$
$\mathbf{D}$	A data set represented by a $n \times p$ matrix
$\mathbf{y}$	The $1 \times n$ label vector of the data set $\mathbf{D}$
n	The total number of samples in data set $\mathbf{D}$
p	The total number of features in data set $\mathbf{D}$
C	The total number of classes in data set $\mathbf{D}$
n_c	The number of samples in class c , and $\sum_{c=1}^C n_c = n$
$\mathbf{Tr}$	A training data set
$\mathbf{Te}$	A test data set
$\mathbf{x}(i)$	The i -th sample in a data set
$y(i)$	The label of the i -th sample
X_j	A feature
$\mathbf{x}_i$	The realization of feature X_j in a data set
$\mathcal{F}$	a feature set space with each element a feature set
$\mathbf{X}$	A feature set
$\mathbf{X}(j)$	The j -th element in the feature set $\mathbf{X}$
$ \mathbf{X} $	The cardinality of the feature set $\mathbf{X}$
$\mathbf{F}^{(k)}$	A feature set containing k features
$\mathbf{F}_l^{(k)}$	A feature set containing k features selected with the l -th training data
$\mathbf{F}_0^{(k)}$	A feature set containing k features selected with full data

Abbreviation Table

Abbreviation	Full Name	Location
ADC	Asymmetric Dependency Coefficient	4.3.2
AUC	Area Under the ROC Curve	2.4.3
AW-SVM	Absolute Weight of SVM	4.3.4
BBET	Bagging-Based Ensemble Technique	5.5.3
ERR	Error Rate	2.4.1
FNR	False Negative Rate	2.4.2
FPR	False Positive Rate	2.4.2
GSBS	Generalized Sequential Backward Selection	3.2.2
GSFS	Generalized Sequential Forward Selection	3.2.2
HDSS	High-Dimensional and Small-Sized	1.1
k NN	k -Nearest-Neighbor	2.3.1
LDA	Linear Discriminant Analysis	2.3.3
MCF-RFE	Multi-Criterion Fusion-Based Recursive Feature Elimination	5.3
NBC	Naive Bayes Classifier	2.3.2
OD	Orthogonal Decomposition	7.4
QDA	Quadratic Discriminant Analysis	2.3.3
ROC	Receiver Operation Characteristic	2.4.3
ROD	Relaxed Orthogonal Decomposition	7.5
RPI	Resampling and Permutation-based Feature Importance	4.6
SBFS	Sequential Backward Floating Selection	3.2.2
SBS	Sequential Backward Selection	3.2.2
SDR	Sample-to-Dimension Ratio	2.6.2
SFFS	Sequential Forward Floating Selection	3.2.2
SFS	Sequential Forward Selection	3.2.2
SVM	Support Vector Machine	2.3.3
SVM-RFE	SVM Recursive Feature Elimination	3.3.3
TNR	True Negative Rate	2.4.2
TPR	True Positive Rate	2.4.2

Chapter 1

Introduction

1.1 Motivation

In pattern recognition, the *a priori* knowledge related to patterns (i.e. samples or instances) is usually unknown and the knowledge needs to be learned or extracted from a number of observed samples. Suppose that there are n samples and each sample contains p variables (features). In traditional low dimensional problems, the number of samples is much greater than the number of variables (i.e. $n \gg p$), which makes the data set contain enough information to estimate the distribution of the patterns. Many existing pattern recognition algorithms have been designed to dispose this kind of low dimensional data and can usually obtain satisfactory results.

With more extensive applications of pattern recognition, high-dimensional and small-sized (HDSS) problems have attracted considerable attention. The prominent characteristic of HDSS data is that the number of variables is much greater than the number of samples, that is, $p \gg n$. One example of the HDSS data is the microarray gene expression data (see Section 2.6 in Chapter 2 for an introduction of microarray data) where usually fewer than one hundred samples (patients) are available altogether for training and testing, but the number of genes (i.e. variables or features) ranges from several thousands to tens of thousands. This problem is

often referred to as the “curse of dimensionality”.

The expression “curse of dimensionality” is due to Bellman (1961) and in pattern classification it means that we need an “enormous” amount of observed samples to get a “good” estimate of the decision function. When $n \ll p$, the estimated decision function may perform very well on the training data but it may perform rather badly on future unseen testing data. To tackle this problem, one practical and effective way is to reduce dimensionality of data by *feature selection*. Feature selection is a process to select the most salient features by removing irrelevant and redundant features. There are three types of feature selection methods including *filter*, *wrapper* and *embedded* methods. When applied to HDSS data, existing feature selection methods may encounter problems such as intractable computational complexity in wrapper methods (Kohavi and John, 1997), ties problem in feature selection with counting-based evaluation criteria (Zhou and Mao, 2006), singularity problem in linear discriminant analysis based feature selection (Hastie et al., 2009), etc.

Another very important yet relatively overlooked problem is the robustness (stability) of feature selection for HDSS data. Robustness, in the context of feature selection, is taken as the ability of selecting the same set of features irrespective of variations in the partitioning of the training data. Robustness actually measures the sensitivity of a feature selection algorithm to the variations in the input training data. The root cause incurring instability in feature selection in HDSS data is the relatively very small number of samples in high dimensional space, which may lead to inaccurate even random evaluation values of feature importance. To illustrate the robustness issue, the top 10 genes selected by Fisher’s ratio (see Section 4.3 about this algorithm) on three batches of training data generated by sampling with

replacement on the full CNS data set (see for details in Section 2.6) are shown in Table (1.1). Note that each gene is represented by its serial number in the data set. It can be observed that the number of common genes (see the bold numbers) of *Batch 1 vs. Batch 2*, *Batch 1 vs. Batch 3* and *Batch 2 vs. Batch 3* are 0, 1 and 1, respectively, which indicates the lack of robustness of Fisher’s ratio algorithm on CNS data. More experimental results illustrating the robustness issue can be found in Section 4.2 in Chapter 4.

Training data	Top 10 genes									
	1	2	3	4	5	6	7	8	9	10
Batch 1	237	3849	2196	3350	2426	1352	5806	348	821	6582
Batch 2	6810	5812	4174	1320	4224	6680	327	6805	844	42
Batch 3	348	677	6533	2474	977	1363	2854	1320	4362	3946

Table 1.1: Top 10 genes selected by Fisher’s ratio on three different batches of training data generated by sampling with replacement on full CNS data set

With the application of feature selection spreading from a preprocess for pattern recognition to a means of knowledge discovery, the robustness of feature selection has become increasingly important in many practical domains that are characterized by a large number of features, but a small number of samples. Some typical examples of these domains include text mining, computational chemistry, bioinformatics and biomedical field. When knowledge discovery is a prime concern, the lack of robustness under slight variations in training data can prove problematic. Take for instance the gene selection in the analysis of microarray gene expression data, the distinct gene subsets selected by a feature selection algorithm (see for example the experimental results in Table (1.1)) could cause great confusions to biological researchers and thus result in their loss of enthusiasm and confidence in applying machine learning techniques to solve biological problems. To avoid these

unfavorable effects, feature selection algorithms should be robust enough to training data variations. A robust feature selection result not only provides domain experts confidence, but also saves their effort and cost for subsequent research, e.g. gene validation through biological experiments.

1.2 Objectives

The thesis is focused mainly on the robustness issue of feature selection for the high-dimensional and small-sized problems. The main goals are threefold:

1. To explore the root causes of instability in feature selection of HDSS data.
2. To investigate the robustness of existing feature selection algorithms under high dimensionality and small sample size, and to carry out studies on how to improve the robustness of the existing feature selection algorithms.
3. Most importantly, to devise new feature selection algorithms with improved robustness performance and with better or no sacrificed performance in classification.

1.3 Overview of the Thesis

This thesis consists of eight chapters and each of those chapters is described below:

Chapter 1

This chapter presents the motivation and objectives of the thesis. An overview of the thesis is also provided.

Chapter 2

Chapter 2 summarizes the background and preliminary knowledge. It first gives an introduction to the skeleton and some basic concepts of pattern recognition and classification systems. Then some popular classification techniques as well as some techniques for evaluating the performance of a classification system are reviewed. A brief introduction to dimension reduction including feature selection and feature extraction is also presented. At last, this chapter describes the microarray data which will be used in the experimental studies in the thesis.

Chapter 3

Chapter 3 surveys feature selection and the robustness issue. Some basic concepts and the typical structure of a feature selection algorithm are first discussed. It then reviews the existing feature selection techniques including filter, wrapper and embedded methods. After that, this chapter focuses on the robustness issue in feature selection. Some concepts and definitions related to robust feature selection are provided, and a detailed literature review related to robustness of feature selection is also given. Moreover, techniques for evaluating the robustness of feature selection algorithms independent of classification are discussed.

Chapter 4

This chapter focuses on improving robustness of feature ranking by multi-criterion combination. We investigate the problems in multi-criterion combination and propose a novel criteria normalization algorithm to combine multiple feature ranking criteria. Two types of inconsistency problems are illustrated, including the incon-

sistency of single feature ranking criterion under perturbations of training data and the inconsistency between different feature ranking criteria. Multi-criterion combination is a reasonable and usually effective strategy for alleviating or solving the inconsistency problems, but one crucial problem in multi-criterion combination is the score normalization between criteria. In this chapter, a new feature importance transformation algorithm based on resampling and permutation is proposed for score normalization. Experimental studies on four popular gene expression data sets show that the multi-criterion combination based on the proposed score normalization produces gene rankings with improved robustness.

Chapter 5

In this chapter, we propose a feature subset selection algorithm named multi-criterion fusion-based recursive feature elimination (MCF-RFE) for the purpose of improving both robustness and classification performance. This algorithm is based on the combination of multiple feature ranking criteria as well as the feature subset search strategy recursive feature elimination (RFE). An improved robustness measure based on Jaccard index is also proposed. Comparative studies between the proposed MCF-RFE, the four basis feature ranking criteria, and the benchmark feature selection algorithm SVM-RFE, are carried out on five gene expression data sets. The experimental results show the superiority of our MCF-RFE algorithm.

Chapter 6

The study in this chapter is an improvement of the existing linear discriminant analysis (LDA) based feature selection algorithm. When applied to gene expression data

which usually have high dimensionality, small sample size and class imbalance (i.e. great discrepancy in the number of samples between classes), LDA-based feature selection encounters problems including singularity of scatter matrix, overfitting, overwhelming and prohibitively computational complexity. In this chapter, we propose a new regularization technique giving more emphasis on minority class, with the expectation of improving overall performance by alleviating overwhelming of majority class to minority class as well as overfitting in minority class. In addition, an incremental implementation of LDA-based feature selection has been developed to reduce computational overhead. Comparative studies on five gene microarray problems show that LDA with the new regularization can produce gene subsets with excellent performance in both classification and robustness.

Chapter 7

Chapter 7 begins with a discussion and examination of the role of feature redundancy in feature subset selection, and a framework of reducing feature redundancy is then proposed. The usual practice for treating redundancy is to utmostly remove redundant features. But this strategy does not necessarily help in some problems such as gene expression data analysis. We argue that a moderate degree of redundancy should be retained in feature subset for improving both classification and robustness. The proposed framework based on this argument is flexible and can be applied to any feature ranking algorithms. We have implemented the framework with the popular feature ranking algorithm—Fisher’s ratio, and conducted experiments on five gene expression data sets. The experimental studies show that the proposed framework can select features with improved classification and acceptable

robustness performance.

Chapter 8

The last chapter concludes the work in the thesis and suggests some possible directions and topics for future research.

Chapter 2

Backgrounds and Preliminaries

This chapter introduces some preliminary knowledge related to pattern recognition.

The overview of pattern recognition systems is given in the first section. Then in the next three sections, a review of a pattern classification system is given, including the structure of a pattern classification system, classification techniques, performance evaluation methods as well as dimension reduction. A brief introduction to microarrays and the utilized data sets with their formal representation are given in the last section.

2.1 An Overview of Pattern Recognition Systems

Pattern recognition is an activity to recognize (such as describe, group and classify) an object. It is in our daily activities and each human being is always performing as a natural pattern recognition system (although he/she may probably never realize this) with innate or acquired abilities to recognize targets. For instance, infants will accept sweet food and will reject bitter one, which may be an inherent pattern recognition ability. With their growth, they can learn to recognize their parents by their appearances or voices, or distinguish an apple from a durian by their shapes and colors, or even their odors.

Pattern recognition has been studied in many areas in past a few decades, including statistics, engineering, artificial intelligence, computer science, cognitive science, psychology, physiology, psychiatry and ethology, among others (Theodoridis and Koutroumbas, 2006). In this thesis, pattern recognition will be discussed mainly from the perspective of machine learning in computer science. Pattern recognition is often an important integral component in a machine intelligence system which is usually utilized for both data preprocessing and decision making. The study of pattern recognition has a very long history, but before sixties of last century, pattern recognition mainly existed in the theoretical research of statistics. The advent and advancement of computers, as well as the industrial demands for information retrieval and utilization, have drawn pattern cognition from theoretical research to practical applications, which in turn sets new demands to pattern recognition and boosts pattern recognition in its theoretical research. The mutual and continuous promotion between the research in theory and practical applications has pushed pattern recognition into its high level in today's engineering applications and research. The research of pattern recognition was often entered in two ways (Bishop, 2006). Some researchers got into the field with solving a real world problem, ranging from the classical ones such as automatic character recognition in machine vision and computer-aided medical diagnosis, to the more recent ones in data mining such as consumer sales analysis and users' web-browsing behavior analysis. Other researchers started the study of pattern recognition with the motivation of developing intelligent machines with "brain-like" performance, and to some extent that could emulate human performance. Intelligent robotics are results of such kind of research.

In machine learning, pattern recognition is a process to assign some sort of

output value to a given input pattern (sometimes it is also termed an instance or a sample), according to the tasks of recognition with a specific recognition algorithm. A *pattern* is usually represented by a collection of features (variables, attributes), each of which is realized with a numerical or categorial (or nominal which will be of one of a set of predefined items such as a blood type of “A”, “B”, “AB” or “O”) measurement, forming a vector (or matrix) and defining a point in an appropriate multidimensional space to together constitute a description of known characteristics of the pattern (Duda et al., 2001). Depending on the application, all the measurements in a pattern may be a description of an object (e.g. some fruit), a representation of an image, a vector of a sequence of genes expression levels, a signal waveform of a piece of speech, the state of a system, and so forth. Take for example the fruit recognition, where a fruit can be described by a group of its characters (features) such as its color, weight, shape and odor. The color can be measured as “red”, “yellow”, “white”, $\dots$; the weight can be represented by a numerical value with a unit such as “100 gram”; the shape may be roughly described as a form what we have already known such as “round” and “ellipsoidal”; the odor can simply be “light smell” or “strong smell”. An apple described as a pattern with the four features may be $\{\text{“red”}, \text{“198 gram”}, \text{“round”}, \text{“light smell”}\}$, while a durian may be $\{\text{“green and brown”}, \text{“1500 gram”}, \text{“ellipsoidal”}, \text{“strong smell”}\}$. Whether a feature is numerical or categorial depends on the way to measure it. When measured in the above way, the weight is a numerical feature while the other three are categorial features.

Depending on information availability and type of learning procedure to generate the output value for an input pattern, pattern recognition is generally divided

into three paradigms: supervised pattern recognition, unsupervised pattern recognition and semi-supervised pattern recognition (Chapelle et al., 2006; Bishop, 2006; Theodoridis and Koutroumbas, 2006). Figure 2.1 depicts the three paradigms of pattern recognition and the details of each kind of pattern recognition are described in the following.

Figure 2.1: Three paradigms of pattern recognition

- *Supervised pattern recognition*

In supervised pattern recognition, it is assumed that a training data set (i.e. a set of patterns with *a priori* knowledge of their corresponding output values) is available. Another typical assumption in supervised pattern recognition is that all the known points (each of which consists of a pattern and its corresponding output value) are drawn i.i.d. (independently and identically distributed) (Theodoridis and Koutroumbas, 2006). The task of supervised pattern recognition is to construct a model that maps from the pattern space to the output value space through a learning procedure on the training data with two (maybe conflicting) aims. One is to let the model match the training data as well as possible (which generally needs a model to be quite complex) and the other is to make the model generalize as well as possible

(which usually requires the model to be as simple as possible) to the unseen patterns from the pattern space. One kind of supervised pattern recognition is *classification*, whose training data have output values belonging to a finite set of predefined class labels and whose goal is to assign each input pattern a class label from the predefined class label set. For instance, to determine whether a given fruit is “a durian” or “not a durian”, or to distinguish a leukemia cancer from “ALL (Acute lymphoblastic leukemia)” to “AML (Acute myelogenous leukemia)”. When the output values of the training data are continuous either in one or multiple dimensions, there comes another kind of supervised pattern recognition—*regression*. With the real-valued output in training data, regression tries to explore the relationship between the output (or dependent variable(s)) and the input features (or independent variables) and then to assign a real value as the output to an input pattern, for example to predict the temperature in weather forecast or the stock index in stock prediction.

- *Unsupervised pattern recognition*

In contrast to supervised pattern recognition, unsupervised pattern recognition will only have a set of patterns without any output information, and the goal of unsupervised pattern recognition is trying to find the inherent structure of the data which can then be used to determine the correct output value for a new input pattern. Again, a standard assumption in unsupervised pattern recognition is that all the input patterns are sampled i.i.d. from the pattern space. A typical example of unsupervised pattern recognition is *clustering*, which is the unsupervised equivalence of classification in supervised pattern recognition with the same type of output. Clustering attempts to explore the input patterns based on some inherent similarity measure (e.g. some kind of distance between patterns, as defined in

multi-dimensional pattern space), and then group them into a number of clusters which may be defined as classes hereafter.

- *Semi-supervised pattern recognition*

Semi-supervised pattern recognition is halfway between supervised and unsupervised pattern recognition (Chapelle et al., 2006). In addition to a set of training data (with known output values) as what is available in supervised pattern recognition, another group of patterns without any output information is also provided. This is the “standard” setting of semi-supervised pattern recognition. In many cases, the patterns without output information will dominate in all patterns (i.e. only a small portion of patterns have output values), which is in line with the practical applications where the acquisition of the output values often requires skilled human agent or complicated equipments and procedures. The cost involved thus may render an infeasible task to obtain a full training data, whereas acquisition of data without output information is relatively inexpensive. In such situations, semi-supervised pattern recognition can be of great practical value. Other settings of semi-supervised pattern recognition also exist. For example, there may be constraints such as “these patterns have similar or more generally, constrained output values”. These different settings will result in distinct views of semi-supervised pattern recognition. In the “standard” setting, semi-supervised pattern recognition is commonly viewed as supervised pattern recognition with additional information on the distribution of the patterns; while in the latter setting where output values have constraints, it is often seen as unsupervised pattern recognition guided by constraints (Chapelle et al., 2006).

The flow chart of a typical pattern recognition system is shown in Figure 2.2.

As shown in the figure, the design of a typical pattern recognition system usually consists of five stages including sensors, feature generation, feature selection and/or extraction, recognizer design and system evaluation (Theodoridis and Koutroumbas, 2006). They are depicted as follows:

Figure 2.2: Flow chart of a typical pattern recognition system

- ◇ Sensors perceive the outside world and gather the input raw data from which we want to mine useful information.
- ◇ After the raw data are gathered, a feature generation stage is sometimes followed to generate a group of features, all or part of which could be useful in the recognition. How those features are generated is however problem dependent. In practice, a larger than necessary number of feature candidates are usually generated.
- ◇ Some of the features generated in the previous stage may not be necessary for the pattern recognition, so in this stage, the feature selection and/or extraction mechanism is adopted to attain an “appropriate” set of features.
- ◇ Having attained the “appropriate”, for a specific recognition task, set of features, how can a recognizer be designed? This is also problem dependent. In

practice, there are different kinds of recognizers depending on the recognition tasks (e.g. classification, regression or clustering) and different methods to design a recognizer for each recognition task.

- ◇ Finally, once the recognizer is designed, how can one assess the performance of the designed pattern recognizer? This is the task of the system evaluation stage. The evaluation to the system is sometimes used to feed back to previous stages for the improvement and enhancement.

As observed from Figure 2.2, the feedback/interaction arrows show that the above stages are not independent. Actually, they are interrelated, and depending on the results, one may return to the earlier stages in order to improve the overall pattern recognition performance. For example, the important features from the feature selection stage may help to discard the unnecessary sensors thus to reduce the complexity as well as the cost of constructing the pattern recognition system.

Among the above three types of pattern recognition, the pattern classification which belongs to supervised pattern recognition will be the scope of my thesis. More details about pattern classification are given in the next section.

2.2 Pattern Classification

Pattern classification may be the most prevalent recognition task in pattern recognition, compared to its counterpart clustering in unsupervised pattern recognition as well as other recognition tasks such as regression and pattern matching. As a sub-branch of supervised pattern recognition, pattern classification is to classify any new pattern (sample) whose class label is unknown into one of a group of predefined

classes, which forms an integral part in most machine intelligence systems built for decision making (Duda et al., 2001).

More formally, a pattern is represented by a p -dimensional vector $\mathbf{x} = (x_1, x_2, \dots, x_p)$, each value in $\mathbf{x}$ is the realization of a corresponding feature that describes one attribute of the pattern. The unknown nature of the pattern is called a *class* or a *category*, which is usually denoted by y and takes the values in a finite set $\mathcal{Y} = \{\pi_1, \pi_2, \dots, \pi_C\}$, where C is the total number of possible classes. For a binary-class problem, the class label set is generally denoted by $\mathcal{Y} = \{+1, -1\}$. Suppose the sample space is $\mathcal{X}$, a classifier is then a mapping function $\mathcal{G}$ maps from $\mathcal{X}$ to $\mathcal{Y}$:

$$\mathcal{G} : \mathcal{X} \rightarrow \mathcal{Y} \quad (2.1)$$

On the other hand, if a pattern vector with p dimensions $\mathbf{x} \in \mathcal{X}$ is regarded as a point lying in a p -dimensional space, then a classifier can also be viewed as a decision boundary (either linear or non-linear) in this space to separate each class from others.

The classification scheme is usually learned based on the availability of a set of samples whose class labels are already known. This set of samples is termed the *training set* $\mathbf{Tr}$ containing n pairs of sample-label $\langle \mathbf{x}(i), y(i) \rangle$, $i = 1, 2, \dots, n$. We use the notation $\mathcal{G}_{\mathbf{Tr}}(\mathbf{x})$ to denote the class label assigned to the sample $\mathbf{x}$ by the classifier $\mathcal{G}$ built on the training data set $\mathbf{Tr}$.

As a sub-branch of (supervised) pattern recognition, a pattern classification system also contains the five basic stages in a pattern recognition system as shown in Figure 2.2. Extended from pattern recognition system, the detailed block diagram

of a pattern classification system is shown in the following Figure 2.3.

Figure 2.3: Detailed block diagram of a pattern classification system

The implementation of a practical classification system can be divided into two phases: system construction and system application. In the first phase, samples with known output labels are utilized to build the classification system including determining parameters in each stage. Then in the second phase, the constructed system will be applied to new generated samples to predict their class labels. The first phase is a learning process while the second is the application of the learned system. In some cases, these two phases are separate which means the constructed system will not change after the learning process. This is suitable for situations where the distribution of the new generated data match very well the distribution of data used in the learning process and will not change with time going on. But

more generally, these two phases will interact with and boost each other, that is, the constructed system in the first phase will be updated after a period of application when new samples with convinced labels are obtained, which makes the constructed system perform better on the future generated samples.

2.3 Classification Techniques

In pattern classification, a classifier can be viewed as a mapping from pattern space to class label space, or more specifically, a decision boundary in the pattern space which segments the pattern space into sections. There exist a great number of classifiers but different kinds classifiers will fit different classification problems. Based on distinct categorizing rules, the existing classifiers can be divided into distinct types, e.g. linear vs. non-linear (Theodoridis and Koutroumbas, 2006), or parametric vs. non-parametric (Sampat et al., 2005). In this thesis, we introduce another more sophisticated taxonomy which categorizes all existing classification techniques into following four types:

- ◇ Classification techniques based on similarity, e.g. the minimum mean distance classifier, nearest neighbor and k -nearest neighbor classifier (Duda et al., 2001).
- ◇ Classification techniques based on probabilistic estimation, e.g. Bayesian decision rule, Naive Bayes classifier (Domingos and Pazzani, 1996, 1997), logistic regression classifier (Hosmer and Lemeshow, 2000).
- ◇ Classification techniques based on optimizing some error criteria, e.g. linear/quadratic discriminant analysis (Hastie et al., 2009), artificial neural net-

works and support vector machine (Vapnik, 1998).

- ◇ Classification techniques based on logic, e.g. decision trees such as CART (Breiman et al., 1984), ID3 (Quinlan, 1986), C4.5 (Quinlan, 1993), and rule-based classification (Fürnkranz, 1999, 2001).

2.3.1 Classification Techniques Based on Similarity

The simplest and most intuitive classifier is designed based on the concept of similarity, under the hypothesis that similar patterns should be assigned to the same class.

Nearest Neighbor Classifier

Nearest neighbor classifier is a classifier based on distance between pairs of samples. It is very simple as no parameter is involved. Further more, it is very powerful in complex boundary curve. Given an unknown input sample $\mathbf{x}$, nearest neighbor classifier computes the distances between $\mathbf{x}$ and each training sample $\mathbf{x}(i)$ for $i = 1, \dots, n$ and assigns the unknown input sample $\mathbf{x}$ to the class of its nearest neighbor in training sample set.

$$L(\mathbf{x}) = L \left[\arg \min_{\mathbf{x}(i)} \text{dist}(\mathbf{x}, \mathbf{x}(i)) \right] \quad (2.2)$$

where $L(\cdot)$ denotes the class label of an input vector and $\text{dist}(\mathbf{x}, \hat{\mathbf{x}})$ is the distance between two samples $\mathbf{x}$ and $\hat{\mathbf{x}}$. There are several ways to define the distance metrics and the most commonly employed way is Euclidean distance.

Nearest neighbor classifier is also called instance-based learner (Aha et al., 1991;

Domingos, 1995) or memory-based reasoning method (Rachlin et al., 1994), for classifying unseen data based on samples (instances) that are stored in memory. Further more, since classification by nearest neighbor classifier is delayed until the unknown input sample $\mathbf{x}$ arrives, it is also referred as lazy learner. One disadvantage of nearest neighbor classifier is that it needs to store all the training data and calculate the distances from the unknown input sample to all the training samples, which could be very slow in classification, especially when the training sample size is large.

k -Nearest-Neighbor Classifier

The k -nearest-neighbor (k NN) classifier (Cover and Hart, 1967; Song et al., 2007) is a variation of nearest-neighbor classifier. Instead of looking at the nearest neighbor of input, it finds k training samples that are closest to input sample $\mathbf{x}$, called the neighborhood of $\mathbf{x}$ denoted as $\mathcal{N}_{\mathbf{x}} = \{\langle \mathbf{x}(i), y(i) \rangle\}$ for $i = 1, \dots, k$. The parameter k is generally predefined as $k = 3, 5, \dots$. Then, the input sample $\mathbf{x}$ is labeled by the majority class in the neighborhood $\mathcal{N}_{\mathbf{x}}$.

$$L(\mathbf{x}) = \arg \max_c \frac{k_c}{k} \quad (2.3)$$

where k_c is number of samples that belong to class c in the neighborhood $\mathcal{N}_{\mathbf{x}}$.

Compared with nearest neighbor classifier, k -nearest neighbor classifier is more robust to noise because of using k nearest neighbors instead of one. Similar to nearest neighbor classifier, k -nearest neighbor's classification is also inefficient due to the demand of memorizing all the training data and calculating the distances of

the unknown input sample to all the training samples.

With a proper distance metric defined, a k -nearest neighbor classifier can be applied to any data sets, such as data with nominal features by employing overlapping metric (Aha et al., 1991).

2.3.2 Classification Techniques Based on Probability Estimation

The second category of classifiers is designed based on estimation of the probability distributions of different classes in given feature space. Any unlabeled sample would be classified to the class with the maximum posterior probability.

Bayesian Decision Rule

Let $c = 1, 2, \dots, C$ be the set of C possible classes, $p(\mathbf{x}|c)$ be the conditional probability density function for $\mathbf{x}$, $p(c)$ be the *a priori* probability of class c . The posterior probability $p(c|\mathbf{x})$ can be computed from $p(\mathbf{x}|c)$ according to Bayesian formula (Duda et al., 2001):

$$p(c|\mathbf{x}) = \frac{p(\mathbf{x}|c)p(c)}{p(\mathbf{x})} \quad (2.4)$$

where the evidence $p(\mathbf{x})$ is defined as:

$$p(\mathbf{x}) = \sum_{c=1}^L p(\mathbf{x}|c)p(c) \quad (2.5)$$

and the Bayesian decision rule is given as:

$$L(\mathbf{x}) = \arg \max_c p(c|\mathbf{x}) \quad (2.6)$$

If the conditional probability density is known, the Bayes decision rule yields an optimal classifier with minimum risk of misclassification (Howson and Urbach, 1993), in the case of the same loss function. The probability density estimation can be either parametric (e.g. Gaussian distributions for continuous features, multi-nominal distribution for nominal features) or non-parametric (e.g. k -nearest neighbor or Parzen window estimators for continuous features, contingency table for nominal features). In practice, however, the estimation of conditional probability $p(\mathbf{x}|c)$ is difficult.

Naive Bayes Classifier

If any feature is dependent on class label but conditionally independent from other features in each class, the joint probability density $p(\mathbf{x}|c)$ is merely the product of the component densities $p(x_j|c)$:

$$p(\mathbf{x}|c) = \prod_{j=1}^p p(x_j|c) \quad (2.7)$$

where x_j is the value of feature j in sample $\mathbf{x}$. This simplification is called Naive Bayes classifier (NBC) (Domingos and Pazzani, 1997) because its independent assumption is too naive to be true in real world problems. However, surprisingly good performance of Naive Bayes classifier have been reported by many authors (Domingos and Pazzani, 1996; Rish, 2001), even though there are clear dependen-

cies between features. Domingos and Pazzani (1996) reported that the key to these results lies in the distinction between classification and probability estimation: correct classification can be achieved even when the probability estimates used contain errors. In fact, NBC often arrives a correct classification in spite of the inaccurate probability estimation (Rish, 2001). The reason of effectiveness of NBC might rely on the one-dimension based probability estimation, which can alleviate the problems from the curse of dimensionality (Bellman, 1961). Furthermore, NBC can handle the mixed data with both continuous and nominal features naturally by choosing different probabilistic estimators for different types of features separately.

2.3.3 Classification Techniques Based on Optimization of Error Criteria

This category of classifiers construct decision functions by minimizing some predefined error criteria.

Support Vector Machines

The most interesting developments recently in classification are the introduction of support vector machines (SVMs) (Cortes and Vapnik, 1995; Vapnik, 1995, 1998). SVM separates data from two classes by an *optimal separating hyperplane* which maximizes the distance to the closet training samples from either class (Cortes and Vapnik, 1995). By maximizing the margin (see below) between two classes on the training data, the optimal separating hyperplane leads to better classification performance on unseen data (Hastie et al., 2009).

Assuming a set of training samples $\langle \mathbf{x}(i), y(i) \rangle$, $i = 1, \dots, n$, each of which is

represented by p features and a class label $y(i) \in \{-1, +1\}$, the support vector machine separates samples by an optimal separating hyperplane (i.e. linear decision boundary):

$$\mathbf{w}^T \mathbf{x} + b = 0 \quad (2.8)$$

where the vector $\mathbf{w} = [w_1, w_2, \dots, w_p]^T$ and the bias b is a scalar. If this training data is linearly separable, we can find an empty margin around the separating hyperplane between two classes by constraining $\mathbf{w}$ and b by:

$$y(i)(\mathbf{w}^T \mathbf{x}(i) + b) \geq 1, \quad i = 1, 2, \dots, n \quad (2.9)$$

By using geometry, the width of margin around the optimal separating hyperplane is $\frac{2}{\|\mathbf{w}\|}$. Consider the optimal problem:

$$\begin{aligned} & \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 \\ & \text{subject to: } y(i)(\mathbf{w}^T \mathbf{x}(i) + b) \geq 1, \quad i = 1, 2, \dots, n \end{aligned} \quad (2.10)$$

This optimal problem is to maximize the separating margin, by choosing $\mathbf{w}$ and b . This is a convex optimization problem. The Lagrange function, to be minimized with respect to $\mathbf{w}$ and b , is:

$$\mathcal{L}(\mathbf{w}, b) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i [y(i)(\mathbf{w}^T \mathbf{x}(i) + b) - 1] \quad (2.11)$$

where $\alpha_i \geq 0$ are Lagrange multipliers. Setting the derivatives to zero, we obtain:

$$\mathbf{w} = \sum_{i=1}^n \alpha_i y(i) \mathbf{x}(i) \quad (2.12)$$

$$\sum_{i=1}^n \alpha_i y(i) = 0 \quad (2.13)$$

Substituting these into Equation (2.11), we obtain the dual form:

$$\mathcal{L}_p(\boldsymbol{\alpha}) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y(i) y(j) \mathbf{x}(i)^T \mathbf{x}(j) \quad (2.14)$$

subject to: $\alpha_i \geq 0, i = 1, 2, \dots, n$

It is a quadratic programming (QP) problem. To find an optimal classifier, we have to find the parameters $\boldsymbol{\alpha}^*$ that maximize Equation (2.14). Substitute $\boldsymbol{\alpha}^*$ to Equation (2.12), we obtain $\mathbf{w}^*$. The parameters $\boldsymbol{\alpha}$ can be explicitly determined using Equation (2.14), but the bias b can not. The bias b is determined by using the Karush-Kuhn-Tucker (KKT) complementarity conditions (see Hastie et al., 2009, page 420). These conditions state that the optimal solutions $\boldsymbol{\alpha}^*$ and b^* must satisfy:

$$\alpha_i^* [y(i)(\mathbf{w}^{*T} \mathbf{x}(i) + b) - 1] = 0. \quad (2.15)$$

Therefore, for any $\mathbf{x}(i)$ with $\alpha_i^* > 0$ (i.e. the samples on the margin), $b^* = y(i) - \mathbf{w}^{*T} \mathbf{x}(i)$. The class label of any input sample $\mathbf{x}$ is given by:

$$L(\mathbf{x}) = \text{sign} [\mathbf{w}^{*T} \mathbf{x} + b^*] = \text{sign} \left[\sum_{i=1}^n \alpha_i^* y(i) \mathbf{x}(i)^T \mathbf{x} + b^* \right] \quad (2.16)$$

Equation (2.12) implies that the decision boundary of SVM classifier is only the

function of the training samples on the margin ($\alpha_i^* > 0$), i.e. the support vectors. Informally speaking, support vectors are those most informative but difficult patterns for the classification task. From this point of view, SVM can be understood in terms of k -nearest-neighbor (k NN) classifier because the decision boundary of k NN is also determined by the patterns around boundary area.

In linearly non-separable case, the optimization problem as defined in Equation (2.10) cannot be solved as it has an empty separate margin between two classes. Slack variables ξ_i should be introduced to form a soft margin optimal problem as follows:

$$\begin{aligned} \min_{\mathbf{w}, b, \xi} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + \tilde{C} \sum_{i=1}^n \xi_i \\ \text{subject to:} \quad & y(i)(\mathbf{w}^T \mathbf{x}(i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, i = 1, 2, \dots, n \end{aligned} \quad (2.17)$$

where the slack variable ξ_i is the regression error for training sample $\mathbf{x}(i)$. Here $\tilde{C}$ is a regularization parameter to control the tradeoff between the number of nonseparable points allowed and the computational complexity.

The Lagrange function for the soft margin optimization problem is

$$\mathcal{L}(\mathbf{w}, b) = \frac{1}{2} \|\mathbf{w}\|^2 + \tilde{C} \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i [y(i)(\mathbf{w}^T \mathbf{x}(i) + b) - 1 + \xi_i] - \sum_{i=1}^n \beta_i \xi_i \quad (2.18)$$

where $\alpha_i \geq 0$ and $\beta_i \geq 0$ are Lagrange multipliers. Similarly, the dual problem is

$$\mathcal{L}_p(\boldsymbol{\alpha}) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y(i) y(j) \mathbf{x}(i)^T \mathbf{x}(j) \quad (2.19)$$

subject to: $0 \leq \alpha_i \leq \tilde{C}, i = 1, 2, \dots, n$

After finding the Lagrange multipliers α_i^* that maximize Equation (2.19), we obtain the solution for optimization problem (2.17),

$$\mathbf{w}^* = \sum_{i=1}^N y(i) \alpha_i^* \mathbf{x}(i) \quad (2.20)$$

$$b^* = y(i) - \mathbf{w}^{*T} \mathbf{x}(i), \text{ for any } i \text{ with } \alpha_i > 0 \quad (2.21)$$

The original optimal hyperplane algorithm proposed by Vladimir Vapnik in 1963 was a linear classifier, as reviewed above. Boser et al. (1992) suggested nonlinear SVM by applying the kernel trick. Nonlinear SVM is trained in two steps: (i) mapping patterns to a higher dimensional space by an appropriate nonlinear transformation $\varphi(\cdot)$, and then (ii) building a linear hyperplane in this transformed space with maximum margin between two classes.

The optimal separating hyperplane for a two-class problem derived by nonlinear SVM can be written as following using a kernel function $K(\mathbf{x}(i), \mathbf{x})$ of a unknown input sample $\mathbf{x}$ (to be classified) and a training sample $\mathbf{x}(i)$:

$$f(\mathbf{x}) = \sum_{i=1}^n \alpha_i^* y(i) K(\mathbf{x}(i), \mathbf{x}) + b^* \quad (2.22)$$

where $K(\mathbf{x}(i), \mathbf{x})$ is the *inner-product kernel* defined by $\varphi(\mathbf{x}(i))^T \varphi(\mathbf{x})$, which is a

scalar. Some common kernel functions include:

$$\begin{aligned}
 \text{Polynomial:} \quad & K(\mathbf{x}(i), \mathbf{x}(j)) = (\mathbf{x}(i)^T \mathbf{x}(j))^q \\
 \text{Radial Basis Function:} \quad & K(\mathbf{x}(i), \mathbf{x}(j)) = \exp(-\gamma \|\mathbf{x}(i) - \mathbf{x}(j)\|^2), \gamma > 0 \\
 \text{Sigmoid:} \quad & K(\mathbf{x}(i), \mathbf{x}(j)) = \tanh(k\mathbf{x}(i)^T \mathbf{x}(j) + c), \text{ for some } k > 0 \\
 & \text{and } c < 0
 \end{aligned}$$

In Equation (2.22), α_i^* can be obtained by solving the optimization problem,

$$\begin{aligned}
 \text{maximize:} \quad & \mathcal{L}_p(\boldsymbol{\alpha}) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n y(i)y(j)\alpha_i\alpha_j K(\mathbf{x}(i), \mathbf{x}(j)) \\
 \text{subject to:} \quad & \sum_{i=1}^N y(i)\alpha_i = 0, \quad \forall i : 0 \leq \alpha_i \leq C, \quad i = 1, 2, \dots, n
 \end{aligned} \tag{2.23}$$

and $b^* = y(i) - \mathbf{w}^{*T} \mathbf{x}(i)$, for any i with $\alpha_i > 0$.

Linear and Quadratic Discriminant Analysis

Assume that each class has multivariate Normal distribution and all classes have the same covariance matrices, we can get a set of linear discriminant functions, one for each class:

$$f_c(\mathbf{x}) = \mathbf{x}^T \Sigma^{-1} \boldsymbol{\mu}_c - \frac{1}{2} \boldsymbol{\mu}_c^T \Sigma^{-1} \boldsymbol{\mu}_c + \log p(c) \tag{2.24}$$

where $\boldsymbol{\mu}_c$ denoted as the mean vector of class c and Σ denoted as the covariance matrix. Equation (2.24) is an equivalent description of decision rule, corresponding to posterior probability of class c with input vector $\mathbf{x}$. It outputs an optimal class label with the maximum posterior probability:

$$L(\mathbf{x}) = \arg \max_c f_c(\mathbf{x}) \tag{2.25}$$

If the covariance matrix of each class is not assumed to be the same, the linear discriminant functions in Equation (2.24) could be modified to quadratic discriminant functions as below:

$$f'_c(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_c)^T \Sigma_c^{-1}(\mathbf{x} - \boldsymbol{\mu}_c) - \frac{1}{2} \log |\Sigma_c| + \log p(c) \quad (2.26)$$

where Σ_c is covariance matrix of the c -th class. Compared with Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA) itself requires more parameters and hence may introduce more variance into the system. Friedman (1989) proposed regularized discriminant analysis by combining LDA and QDA, where the regularized covariance matrix $\Sigma_c(\alpha)$ is defined as:

$$\Sigma_c(\alpha) = \alpha \Sigma_c + (1 - \alpha) \Sigma \quad (2.27)$$

where the parameter α ($\in [0, 1]$) allows a continuous model between LDA ($\alpha = 1$) and QDA ($\alpha = 0$).

2.3.4 Classification Techniques Based on Logic

In this section we will concentrate on two groups of logical learning methods: decision trees and rule-based classifiers.

Decision Trees

A decision tree in pattern classification is a tree-like model (or graph) that classifies patterns (samples or instances) starting from the root by sorting them based on

the values of features. Each node in a decision tree denotes a feature, the value of which is the basis to split the node into two or multiple branches (children). Given a training data set, a decision tree is trained by iteratively selecting an individual feature that is the most informative at each node of the tree and dividing the node into children. There are numerous methods for finding the feature that best divides the training data such as information gain (Hunt et al., 1966) and gini index (Breiman et al., 1984). The child “pure” enough will become a leaf and output the major class as a decision, while others will continue to grow until the predefined requirement of purity is met somewhere. Decision trees are suitable for the nature of nominal feature for its splitting mechanism. However, continuous features can also be integrated by using continuous feature discretization. Figure 2.4 shows an example of decision tree showing survival of passengers on the Titanic (“sibsp” is the number of spouses or siblings aboard). The numbers under the leaves denotes the probability of survival and the percentage of observations in the leaf ¹.

Figure 2.4: An example of decision tree

Decision trees are different depending on how to split, how to process continu-

¹The picture in Figure 2.4 is downloaded from online Wikipedia at: http://en.wikipedia.org/wiki/Decision_tree_learning.

ous features and how to prune the tree. CART (classification and regression tree) (Breiman et al., 1984) generates a binary tree, i.e. each node is split into two children nodes by asking, for example, the *sex* is *male* or not; the *age* is greater than a specified value (e.g. 9.5) or not? Besides binary splitting, multi-split is also available, as employed in ID3 (interactive dichotomizer version 3) (Quinlan, 1986) and C4.5 (Quinlan, 1993), where the number of splits depends on the number of distinct values in features. Multi-split may generate trees faster than binary split. However, it fragments the feature space faster and results in a fat tree with much bigger tree size. To avoid overfitting problem in a big tree, C4.5 prunes the node with a wide confidence interval on the training error in the error estimation (Quinlan, 1993).

Rule-Based Classifiers

Decision trees can be translated into a set of rules by creating a separate rule for each path from the root to a leaf in the tree (Quinlan, 1993). However, rules can also be directly induced from training data using a variety of rule-based algorithms. A rule-based classifier is a technique for classifying instances using a collection of rules, while each rule is usually in the “*if ... , then ...*” form.

Classification rules represent each class by disjunctive normal form (DNF). A k -DNF expression is of the form:

$$(R_1 \wedge R_2 \dots R_n) \vee (R_{n+1} \wedge R_{n+2} \dots R_{2n}) \vee \dots \vee (R_{(k-1)n+1} \wedge R_{(k-1)n+2} \dots R_{kn}) \quad (2.28)$$

where k is the number of disjunctions, n is the number of conjunctions in each disjunction.

The goal is to construct the smallest rule-set that is consistent with the training data. A separate-and-conquer algorithm (covering algorithms) search for a rule that explains a part of its training instances, separates these instances and recursively conquers the remaining instances by learning more rules, until no instances remain.

RIPPER is a well-known rule-based algorithm (Cohen, 1995). It forms rules through a process of repeated growing and pruning. During the growing phase the rules are made more restrictive in order to fit the training data as closely as possible. During the pruning phase, the rules are made less restrictive in order to avoid overfitting, which can cause poor performance on unseen instances. RIPPER handles multiple classes by ordering them from the least to the most prevalent and then treating each in order as a distinct two-class problem. Other fundamental learning classifiers based on decision rules include the AQ family (Michalski and Chilausky, 1980), CN2 (Clark and Niblett, 1989) and fuzzy rule-based classifiers (Bonarini, 2000).

2.4 Classification Performance Evaluation

Classification performance evaluation is an crucial stage in the construction of a pattern classification system. It would be an important index feeding back to previous stages for the improvement of each single stage or the whole classification system; it will also be a (sometimes the sole) way to reflect the ability of the classification system (e.g. generalization ability), which can then help to build confidence for users.

There exist abundant measures to evaluate classification performance, while they

are commonly based on the classification outcomes. For a binary-class classification problem, without loss of generality, assuming one class is “positive” and the other class is “negative”, then each classified (predicted) instance (sample) will belong to one of the following four categories:

- ◇ True positive (TP): the instance is predicted to be “positive” while it is actually positive.
- ◇ False positive (FP): the instance is predicted to be “positive” while it is actually negative.
- ◇ True negative (TN): the instance is predicted to be “negative” while it is actually negative.
- ◇ False negative (FN): the instance is predicted to be “negative” while it is actually positive.

The contingency table or confusion matrix is usually utilized as a tool to visualize the classification outcomes. A confusion matrix for the case of binary-class is shown in Table 2.1. Each column of the matrix contains the instances in a predicted class (either “Positive” or “Negative”), while each row contains the instances in an actual class. The numbers (“*a*”, “*b*”, “*c*”, “*d*”) denote the number of instances in each category; the last column and row are the total number of instances in the predicted and actual classes. One benefit from the confusion matrix is that it is easy to see if the classification system is confusing two classes (i.e. commonly mislabeling one as another).

		Predicted Class		total
		Positive	Negative	
Actual Class	Positive	TP (a)	FN (b)	$a + b$
	Negative	FP (c)	TN (d)	$c + d$
total		$a + c$	$b + d$	

Table 2.1: Contingency table

2.4.1 Error Rate and Accuracy

The error rate (ERR) of a classifier is the probability that the classifier classifies a randomly selected sample to the wrong class. The true error rate depends on the feature-label distribution and if the feature-label distribution is known, the error rate can be computed exactly. However, we cannot get the exact distribution in practice and we need to estimate the error rate by using the error estimators. With reference to Table 2.1, the error rate is the proportion of the total number of predictions that are wrong. It is determined using the equation:

$$ERR = \frac{b + c}{a + b + c + d} \quad (2.29)$$

and the accuracy (ACC) is:

$$ACC = \frac{a + d}{a + b + c + d} = 1 - ERR \quad (2.30)$$

Error rate is a widely used measure to evaluate the performance of a classifier. But when a data set is severely imbalanced (i.e. the number of samples in one class

is hugely greater than the number of samples in the other class), the error rate of a classifier will no longer be suitable to represent the true performance of the classifier. This can easily be understood by an example: suppose in a binary-class data set, the positive class has 10 samples while the negative class has 990 samples, if naively a classifier always classes samples into the negative class, then the classification error rate will be 1%, which is rather good from this absolute value. Actually this is not good performance as the classifier has a 100% recognition rate for the negative class while a 0% recognition rate for the positive class.

2.4.2 Sensitivity and Specificity

Even though the error rate (or accuracy) is very popular when evaluating the performance of a classifier, it may not be an adequate performance measure as stated above. There exist other measures which can reflect and remedy this problem, such as sensitivity and specificity.

Sensitivity

Sensitivity, also called *true positive rate* (TPR) or *recall*, is the proportion of correctly predicted positive samples (i.e. TP in the contingency table) in all actual positive samples. It is calculated using the following equation:

$$sensitivity = \frac{a}{a + b} \quad (2.31)$$

Like the error rate and accuracy, a similar measure is the *false negative rate*

(FNR) which is defined as:

$$FNR = \frac{b}{a + b} = 1 - \textit{sensitivity} \quad (2.32)$$

Specificity

Specificity is also called *true negative rate* (TNR), and it reflects the proportion of correctly predicted negative samples (i.e. TN in the contingency table) in all actual negative samples. It is calculated using the following equation:

$$\textit{specificity} = \frac{d}{c + d} \quad (2.33)$$

and the corresponding counterpart measure is *false positive rate* (FPR) which is defined as:

$$FPR = \frac{c}{c + d} = 1 - \textit{specificity} \quad (2.34)$$

Take the above example (with 10 positive samples and 990 negative samples) for instance, if a classifier classes all samples into the negative class, then the sensitivity and specificity will be 0% and 100%, respectively, the two of which can give an image of the true performance of the classifier.

2.4.3 ROC and AUC

Sensitivity and specificity together can represent the performance of a classifier, however, it is generally expected to utilize one rather than multiple measures to evaluate the performance. Besides, sensitivity and specificity are dependent and there is usually a trade-off between the two measures in practice. To explain their

trade-off, let us assume a classifier whose prediction value ranges from D^- (minimum) to D^+ (maximum). For a specific classification, a decision threshold value thv is determined, above which samples are considered as positive and below which samples are considered as negative. When thv decreases from $thv \geq D^+$ to $thv \leq D^-$, the corresponding sensitivity and specificity will vary from 0 to 1 and from 1 to 0, respectively. After thv is determined, a trade-off is also made between sensitivity and specificity. This trade-off can be represented graphically using an Receiver Operation Characteristic (ROC) curve.

An ROC curve is a graphical plot of the sensitivity (or *true positive rate*), vs. 1-specificity (or *false positive rate*), for a binary classifier system as its decision threshold value is varied. Figure 2.5 illustrates the relationship between the decision threshold value in a binary-class problem (left figure) and the corresponding ROC curve (right figure).

Figure 2.5: Illustration of a binary-class classification problem and the ROC curve

An ROC curve demonstrates several things:

1. It shows the trade-off between sensitivity and specificity (any increase in sensitivity will be accompanied by a decrease in specificity).
2. The closer the curve follows the left-hand border and then the top border of

the ROC space, the more accurate the classification is.

3. The closer the curve comes to the 45-degree diagonal of the ROC space, the less accurate the classification is.

The area under the ROC curve (AUC) is another widely used measure of classification performance. AUC is applied under the same setting as accuracy, but it is independent of the decision threshold value and it is a common belief that AUC serves as better measure than accuracy (Bradley, 1997; Ling et al., 2003).

2.4.4 Validation Techniques for Performance Evaluation

In the construction of a real pattern classification system, the available data are generally limited and they are divided into training and testing data. A classifier is first trained on the training data and then tested on the testing data. There are several validation techniques to estimate how a classification system will perform in practice. In this section, we will take the classification error *err* as an example of the performance measure to introduce some of the popular validation techniques.

Consider a data set $\mathbf{D}$ with n pairs of samples and labels, $\mathbf{D} = \{\langle \mathbf{x}(i), y(i) \rangle\}$, $i = 1, 2, \dots, n$, where $\mathbf{x}(i)$ is the i -th sample in the data set and $y(i)$ is its corresponding label, $y(i) \in \{-1, 1\}$ for a binary problem. The validation techniques are described as follows.

Resubstitution Method

In resubstitution estimate, all the n samples in data set $\mathbf{D}$ are first used as the training samples to train a classifier $\mathcal{G}_{\mathbf{D}}(\cdot)$, and then the same n samples are used as

the testing samples to test the previously built classifier. The resubstitution error (or apparent error), $\widehat{err}_{resub}$, is the ratio of the number of misclassified samples to the total number of training samples:

$$\widehat{err}_{resub} = \frac{1}{n} \sum_{i=1}^n \mathbf{I}(\mathcal{G}_{\mathbf{D}}(\mathbf{x}(i)) \neq y(i)) \quad (2.35)$$

where $\mathbf{I}(A)$ is an indicator function, i.e. $\mathbf{I}(A) = 1$ if the statement A is true and $\mathbf{I}(A) = 0$ otherwise. Resubstitution error estimate is usually severely optimistic, especially when the data set is small-sized.

Holdout Method

Holdout method first partitions all available samples are into two subsets: one is the training sample set $\mathbf{Tr}$ and the other is the testing sample set $\mathbf{Te}$ (which contains the “hold out” samples). Then a classifier $\mathcal{G}_{\mathbf{Tr}}(\cdot)$ is built based on the training sample set and tested on the testing sample set. The holdout error $\widehat{err}_{ho}$ is:

$$\widehat{err}_{ho} = \frac{1}{|\mathbf{Te}|} \sum_{\langle \mathbf{x}(i), y(i) \rangle \in \mathbf{Te}} \mathbf{I}(\mathcal{G}_{\mathbf{Tr}}(\mathbf{x}(i)) \neq y(i)) \quad (2.36)$$

where $|\mathbf{Te}|$ is the cardinality of the testing set $\mathbf{Te}$. A difficulty of the holdout method is that how to allocate the total n samples into the training set and testing set. Besides, the classification error estimated by this method will be unreliable when size of the data set is not large enough.

Leave-One-Out Method

When used to estimate the classification error, the leave-one-out method is almost unbiased. For a n -sample data set $\mathbf{D}$, each time a sample $\mathbf{x}(i)$ is selected as the testing sample, the remaining $n - 1$ samples (with their known labels) are used to build a classifier $\mathcal{G}_{\mathbf{D} \setminus \langle \mathbf{x}(i), y(i) \rangle}(\cdot)$ (the notation “ $\setminus$ ” denotes the operation of set subtraction) and the testing sample is classified by the built classifier. This procedure is repeated n times until all the n samples are selected as the testing samples once and only once. The leave-one-out error $\widehat{err}_{loo}$ is given by:

$$\widehat{err}_{loo} = \frac{1}{n} \sum_{i=1}^n \mathbf{I}(\mathcal{G}_{\mathbf{D} \setminus \langle \mathbf{x}(i), y(i) \rangle}(\mathbf{x}(i)) \neq y(i)) \quad (2.37)$$

k -fold Cross-Validation Method

k -fold cross-validation is very popular in estimation. In one round of k -fold cross-validation, a data set $\mathbf{D}$ is first randomly partitioned into k subsets (folds) $\mathbf{D}_1, \mathbf{D}_2, \dots, \mathbf{D}_k$, which are of approximately equal size and are mutually exclusive (i.e. $\mathbf{D}_1 \cup \mathbf{D}_2 \cup \dots \cup \mathbf{D}_k = \mathbf{D}$, and $\mathbf{D}_i \cap \mathbf{D}_j = \emptyset$ for $i \neq j$ and $i, j \in \{1, 2, \dots, k\}$). A classifier is then trained and tested k times, in each time t ($t \in \{1, 2, \dots, k\}$) a subset $\mathbf{D}_t$ is retained as the testing data and the remaining $k - 1$ subsets $\mathbf{D} \setminus \mathbf{D}_t$ as the training data. The estimated k -fold cross-validation error rate is the overall number of incorrectly classified samples in data set $\mathbf{D}$, divided by n , the total number of samples in the data set (Kohavi, 1995). Formally, let $\mathbf{D}_{(i)}$ be the data subset that

contains sample $\mathbf{x}(i)$, then the k -fold cross-validation error $\widehat{err}_{kcv}$ is given by:

$$\widehat{err}_{kcv} = \frac{1}{n} \sum_{i=1}^n \mathbf{I}(\mathcal{G}_{\mathbf{D} \setminus \mathbf{D}_{(i)}}(\mathbf{x}(i)) \neq y(i)) \quad (2.38)$$

In practice, for consideration of reliability of estimation, the k -fold cross-validation will be carried out multiple rounds with each round an independent random partition of the data set and the final estimated error rate is the average of the multiple rounds estimations.

Bootstrap Method

Bootstrap was first introduced in Efron (1983). Given a data set $\mathbf{D}$ with n labeled samples, bootstrap method produces a bootstrap data set $\mathbf{D}^*$ by uniformly sampling n samples from the original data set $\mathbf{D}$ with replacement. Because of the sampling with replacement, the probability of any given sample in $\mathbf{D}$ not appearing in $\mathbf{D}^*$ will be $(1 - \frac{1}{n})^n \approx e^{-1} = 0.368$, then the expected number of distinct samples appearing in $\mathbf{D}^*$ is $0.632n$.

In practice, the bootstrap error is approximated on a total of B independent bootstrap data sets $\mathbf{D}_b^*$, for $b = 1, 2, \dots, B$. Define $J_b^i = 1$ if sample $\langle \mathbf{x}(i), y(i) \rangle$ is not contained in the b -th bootstrap data set $\mathbf{D}_b^*$, and $J_b^i = 0$ otherwise. The bootstrap error estimated on the basis of B bootstrap data sets is computed as:

$$\widehat{err}_B = \frac{1}{n} \sum_{i=1}^n \frac{\sum_{b=1}^B \mathbf{I}(\mathcal{G}_{\mathbf{D}_b^*}(\mathbf{x}(i)) \neq y(i)) J_b^i}{\sum_{b=1}^B J_b^i} \quad (2.39)$$

$\widehat{err}_B$ is upwardly biased, and the .632 bootstrap error estimation $\widehat{err}_{B.632}$ is given

by:

$$\widehat{err}_{B.632} = 0.632\widehat{err}_B + 0.368\widehat{err}_{resub} \quad (2.40)$$

where $\widehat{err}_{resub}$ is resubstitution error estimated on the whole data set $\mathbf{D}$ as given in Equation (2.35).

2.5 Dimension Reduction

In a pattern recognition or classification system for high-dimensional problems, dimension reduction is usually an indispensable stage. Dimension reduction is the process to reduce the number of variables (features) under consideration and it is a way to overcome the curse of dimensionality when dealing with very high dimensional data or a way for modeling such kind of data. By a careful dimension reduction, it is expected that the data representation in the lower feature space will retrieve or preserve as much as possible the relevant information from the original feature space. The two techniques of dimension reduction, i.e. feature selection and feature extraction, are briefly introduced as follows.

2.5.1 Feature Selection

Feature selection, also known as variable selection, attribute selection, or gene selection when dealing with microarray gene expression data in bioinformatics community, is a commonly used dimension reduction technique. In most circumstances, feature selection seeks to find a subset of relevant or influential features from all original features (Kohavi and John, 1997; Guyon and Elisseeff, 2003; Saeys et al., 2007, 2008). In some special cases, feature selection is utilized to rank all features

to identify individually target-correlated features, for example, biomarker identification in bioinformatics research (Liu and Motoda, 1998; Ruiz et al., 2006; Zuber and Strimmer, 2009).

Several potential benefits can be obtained by feature selection (Guyon and Elisseeff, 2003; Rakotomamonjy, 2003), e.g. improving the prediction performance (accuracy, generalization, etc) of the classifiers (predictors), increasing the ability to build simpler and faster learning models, and providing a better understanding of the underlying process that generated the data. What's special, feature selection preserves the original semantics of the variables and hence offers the advantage of interpretability to domain experts (Saeys et al., 2007). One drawback of feature selection may be that, when the selected number of features are small, there would be unavoidable information loss, as many informative features are completely unused.

2.5.2 Feature Extraction

Feature extraction (or sometimes called feature construction) is a dimension reduction technique that reduces the dimension by transforming data from the original high dimensional space into a low dimensional space. Unlike feature selection which does not change the format of the original features, feature extraction creates (e.g. by linear combination) a set of new features using the whole set of the original features, hence the information of all original features may be utilized. However, building a new feature in feature extraction is an opportunity to incorporate domain knowledge into the data and can be very application specific.

There are a number of generic feature extraction methods, including linear (e.g. principle component analysis, independent component analysis and partial least

squares) and nonlinear (e.g. kernel principle component analysis, principle curves, topographic maps and neural networks) ones (Rosipal and Krämer, 2006; Fodor, 2002; Antoniadis et al., 2003).

2.6 Data Analyzed in Thesis

New technologies in the form of improved instrumentation have made it possible to measure a vast number of variables simultaneously. But in practice, it is difficult or too expensive to get enough samples of those variables, and most of the time, only a small number of samples are available. High-dimensional and small-sized (HDSS) data are emerging and becoming increasingly common in many fields of science, including chemometrics, medical imaging and genetic microarrays (Hall et al., 2005). Since pattern classification of the microarray gene expression data is the focus of this study, we next give a more detailed description of this kind of HDSS data.

2.6.1 A Brief Introduction to Microarrays

Microarrays are a high-throughput technology which can be used to simultaneously gauge the expression levels of thousands of genes or even all genes in a genome. Since its emergence, this technology has brought about a revolution to the research and to our understanding of the molecular mechanisms underlying normal and dysfunctional biological processes. Microarray gene expression data is one of the most typical HDSS data. cDNA microarrays and high-density oligonucleotide chips are two novel biotechnologies to produce this kind of data. The former technology is developed in the Brown and Botstein labs at Stanford (Derisi et al., 1997) while

the latter is from the Affymetrix Company (Lockhart et al., 1996) and can deal with more genes in one experiment. By allowing monitoring the changes of gene expression levels during important biological processes for thousands of genes simultaneously, microarray experiments may lead to a more comprehensive understanding of the molecular variations among tumors and hence to finer and more reliable classification.

A cDNA microarray consists of a flat, solid substrate (typically glass) with an organic coating, typically an organo-functional alkoxysilane (Conzone and Pantano, 2004). The coated glass is then grafted (by printing or in situ synthesis) with thousands of various known DNA sequences (probes) at predefined locations (spots). The cDNA microarray can be thought of as a glass-based, biological sensor that can contain over 30 000 (an estimate for the total number of genes in the human genome) distinct, known probes at prespecified locations. This powerful, glass-based, biological sensor provides researchers with the ability to characterize the human genomic state fully with a single, miniature experiment. The brief steps in a cDNA microarray experiment are as follows:

- ◇ Targets isolating and labeling: Two complex RNA populations are extracted and separated from healthy and cancerous tissues respectively. Then the two separate populations (targets) are labeled using different fluorescent dyes (e.g. a green fluorescent dye for healthy targets and a red fluorescent dye for cancerous targets).
- ◇ Targets and microarrays combination and hybridization: After labeling, the healthy and cancerous targets are combined in equal proportions and applied

to the surface of a cDNA microarray, which initiates the hybridization process. Hybridization occurs when the sequences of the target and probe are complementary, whereby strong hydrogen bonding forms.

- ◇ Microarray washing: Non-complementary targets and probes do not form strong hydrogen bonding and subsequently removed from the microarray surface by stringent washing.
- ◇ Image scanning and quantification: After hybridization and washing, a scanning process is conducted to image all the probes simultaneously and the image is quantified. The ratio of the red and green fluorescent intensities for each probe spot is indicative of the relative abundance of the corresponding DNA probe in the two nucleic acid targets. Spots fluorescing strongly red or green indicate strong gene activity in the cancerous or healthy tissues, respectively, while spots fluorescing yellow are indicative of weak differences of gene activity between healthy and cancerous tissues.

2.6.2 Micriarray Gene Expression Data Sets

In the experimental studies in this thesis, six microarray gene expression data sets will be utilized. Their details are in following:

Colon data (Alon et al., 1999): This data was first described by Alon et al. in 1999. The data set contains expression levels of 2000 genes for 62 samples including 22 normal samples and 40 colon cancer samples. The task is to distinguish between *normal* and *tumor* samples. The original data can be downloaded at: <http://microarray.princeton.edu/oncology/>.

Lymphoma data (Alizadeh et al., 2000): The task of this study is to differentiate two distinct types of diffuse large B-cell lymphoma (DLBCL) using gene expression data. There are 47 samples in the data set, 24 of them are from "germinal centre B-like" group while 23 are "activated B-like" group. The original raw data are available at: <http://llmpp.nih.gov/lymphoma/>.

Leukemia data (Golub et al., 1999): Introduced by Golub et al. in 1999, this data set contains expression levels of 7129 genes for 47 ALL (Acute lymphoblastic leukemia) leukemia patients and 25 AML (Acute myelogenous leukemia) leukemia patients. The original data can be downloaded at: <http://www-genome.wi.mit.edu/cgi-bin/cancer/datasets.cgi>.

Prostate data (Singh et al., 2002): This data set contains expression level of 12600 genes for 136 samples including 77 prostate tumors and 59 normal samples. The data set was first described by Singh et al. and the original data is available at: <http://www-genome.wi.mit.edu/cgi-bin/cancer/datasets.cgi>.

CNS data (Pomeroy et al., 2002): The goal of this study is the molecular investigation of treatment effectiveness for embryonal CNS (Central Nervous System) tumors. The task is to distinguish between *failed* and *succeed* treatment outcomes. There are 60 patients with 7129 genes in this data set, where 21 patients are survivors and 39 patients are failures. The original data are available at: <http://www-genome.wi.mit.edu/mpr/CNS/>.

DLBCL data (Shipp et al., 2002): This set of data contains 58 DLBCL (diffuse large b-cell lymphoma) samples and 19 FL (Follicular Lymphoma) samples with 7129 genes. The data are available at: <http://www-genome.wi.mit.edu/cgi-bin/cancer/datasets.cgi>.

A brief description including the number of features, the number of samples, the number of samples in each class and the *SDR* (sample-to-dimension ratio) of the above data sets is given in Table 2.2.

Table 2.2: Data sets description

Data	Features	Samples	Class1 vs. Class2	<i>SDR</i>
Colon (Alon et al., 1999)	2000	62	40 vs. 22	3.1%
Lymphoma (Alizadeh et al., 2000)	4026	47	24 vs. 23	1.17%
Leukemia (Golub et al., 1999)	7129	72	47 vs. 25	1.01%
Prostate (Singh et al., 2002)	12600	136	59 vs. 77	1.08%
CNS (Pomeroy et al., 2002)	7129	60	39 vs. 21	0.84%
DLBCL (Shipp et al., 2002)	7129	77	58 vs. 19	1.08%

In our experimental studies in the thesis, all the above data sets are pre-processed and standardized to zero mean and unit standard deviation as described by Dudoit et al. (2002). When constructing a classifier for each data, linear SVM was employed as the pattern classifier. The linear SVM was implemented using the LIBSVM (Chang and Lin, 2001) toolbox of version 2.88 and parameter C was empirically set to 0.01 for the first three data sets and 0.1 for the last three data sets. Due to small sample size of the data sets, classification error was estimated using .632 bootstrap (Chernick, 2007) with 300 repeats.

2.6.3 Formal Representation of Data Sets

The microarray gene expression data sets analyzed in the thesis are all of binary classes. In formal, given a binary-class problem with totally n samples and p features, the data set can be represented by n sample-label pairs:

$$\langle \mathbf{x}(1), y(1) \rangle, \langle \mathbf{x}(2), y(2) \rangle, \dots, \langle \mathbf{x}(n), y(n) \rangle \quad (2.41)$$

where $\mathbf{x}(i) = [x_1(i), x_2(i), \dots, x_p(i)]$ denotes the i -th sample and $y(i) \in \{-1, 1\}$ is its label. Then for a feature X_j , its realization in the problem can be denoted by a vector $\mathbf{x}_j$ with a label vector $\mathbf{y}$:

$$\mathbf{x}_j = [x_j(1), x_j(2), \dots, x_j(n)]^T \quad (2.42)$$

$$\mathbf{y} = [y(1), y(2), \dots, y(n)]^T \quad (2.43)$$

Chapter 3

Feature Selection and the Robustness Issue

3.1 Feature Selection: What, Why and How?

The problem of feature selection can be defined as finding a subset of features with certain size d , from all original p ($p > d$) features, that lead to the largest possible generalization or equivalently to the minimal risk (e.g. in pattern classification, to minimize the prediction error with a selected feature subset). More formally, given a full set of p features $\mathbf{X} = \{X_1, X_2, \dots, X_p\}$, we model a feature subset of $\mathbf{X}$ by an indication vector $\mathbf{v} \in \{0, 1\}^p$ with variables v_i : $v_i = 1$ indicating presence of feature X_i in a feature subset while $v_i = 0$ indicating absence of X_i ($i = 1, 2, \dots, p$). Assume we have obtained a parameterized family of classification or regression functions $f : \mathcal{R}^n \rightarrow \mathcal{R}, (\boldsymbol{\theta}, \mathbf{x}) \mapsto f(\boldsymbol{\theta}, \mathbf{x})$, then the objective of feature selection is to find a vector of indicator variables $\mathbf{v}^* \in \{0, 1\}^p$ and a $\boldsymbol{\theta}$ that minimize the expected risk:

$$risk(\mathbf{v}, \boldsymbol{\theta}) = \int L[f(\boldsymbol{\theta}, \mathbf{v} \circledast \mathbf{x}), y] dP(\mathbf{x}, y) \quad (3.1)$$

where $\otimes$ denotes the point-wise product of two vectors, L is a loss function and P is a measure on the domain of the training data. In some cases we will also have the additional constraint $s(\mathbf{v}) \leq v_0$, where $s : \{0, 1\}^p \rightarrow \mathcal{R}^+$ measures the sparsity of a given indication vector $\mathbf{v}$. For example, s could be defined as $s(\mathbf{v}) := l_o(\mathbf{v})$, that is to bound the zero norm $l_o(\mathbf{v})$ – which counts the number of nonzero entries in $\mathbf{v}$ – from above by some number v_0 .

Data collected in real world are generally of coarse-quality. In addition to the noises in the measuring procedure, many features themselves are not necessarily relevant to a specific learning task. For example, for the purpose of identifying a disease with microarray gene expression data, most genes are noisy and irrelevant, and only a small portion of genes are disease-related, among which large redundancy (correlation between genes) may also exist. On the other hand, it is well acknowledged that many practical machine learning algorithms will degrade in performance when unnecessary features exist (Kohavi and John, 1997). For example, the performance of decision tree-based algorithms including ID3, C4.5 and CART, and the instance-based algorithms such as IBL (Aha et al., 1991), degrades when facing many features that are unnecessary for predicting the targets. Algorithms such as Naive Bayes are robust with respect to irrelevant features (i.e. their performance degrades very slowly as more irrelevant features are added) but their classification performance may degrade quickly if correlated features are added, even if those features are target-relevant. Therefore, feature selection algorithms are required to choose the target-relevant but uncorrelated features for the purpose of improving classification performance. Furthermore, when considering the curse of dimensionality existing in many practical problems such as gene expression data, feature selection

is also a very easy way to alleviate the effect of the curse of dimensionality to improve prediction performance. By removing most irrelevant and redundant features from the data, feature selection can obtain many other potential benefits including facilitating data visualization and data understanding, reducing the measurement and storage requirements, reducing training and utilization times of the learning process, improving model's interpretability and generalization capability, etc.

As depicted in Figure 3.1, a typical feature selection process consists of four key steps (Dash and Liu, 1997), namely subset generation, subset evaluation, stopping criterion and result validation. With the input of training samples in the full feature space, a subset generation (or search) method first produces a candidate feature subset, then this candidate feature subset is assessed by a subset evaluation criterion and the goodness is compared with that of the previous best feature subset: if current candidate feature subset is found to be better, then it replaces the previous best feature subset. The loop of these two steps continues until a stopping criterion is satisfied and the feature selection process outputs a selected feature subset to a validation procedure. More details of the four key steps are discussed in next section.

Figure 3.1: Four key steps of feature selection

3.2 Components in Feature Selection Procedure

Feature selection is the process of selecting relevant and non-redundant features from the original features for a pattern recognition task. A feature selection algorithm consists of the following components:

3.2.1 Subset Evaluation Criteria

In order to differentiate between candidate features or feature subsets and guide the search process to choose the “best” feature subset, we need a means of evaluating the goodness of a given feature subset, that is, a feature subset evaluation criterion. For good pattern classification performance, the “goodness” of a feature subset is generally based on its ability to discriminate between two or more classes. In formal, a feature subset evaluation criterion is defined as a mapping Q :

$$Q : \mathcal{F} \rightarrow \mathcal{R} \quad (3.2)$$

where $\mathcal{F}$ is the feature set space and $\mathcal{R}$ is the real value space.

The meaning of “best” or “optimal” of a feature subset is always relative to a certain evaluation criterion, that is, for a given feature subset $\mathbf{F} \in \mathcal{F}$ which is optimal under an evaluation criterion Q_a , $\mathbf{F}$ may not be optimal under another evaluation criterion Q_b . There exist many evaluation criteria in literature which can be divided into five categories: *distance-based*, *information (uncertainty)-based*, *dependence-based*, *consistency-based*, and *classification error rate-based* (Ben-Bassat, 1982; Doak, 1992; Dash and Liu, 1997).

Distance-based criteria: This kind of criteria are also called separability, diver-

gence, or discrimination based criteria. For a two-class problem, a feature X is preferred to another feature Y if X induces a greater difference between the two-class conditional probabilities than Y . An example of this kind of criteria is the Euclidean distance.

Information-based criteria: These criteria typically determine the information gain from a feature. To determine a target, if more information can be gained from feature X than from feature Y , then feature X is said to be better than feature Y . An examples of this kind of criteria is the entropy (Biesiada et al., 2005).

Dependence-based criteria: Dependence (or equivalently correlation) measures the degree of relation between two variables by changing the value of one variable with observing the value of another variable. The coefficient is a classical dependence measure and can be used to find the correlation between a feature and a class. If feature X is more correlated to a class than feature Y , then feature X is more important than feature Y . All evaluation criteria based on dependence can be divided between distance and information (Ben-Bassat, 1982). An example of dependence-based criteria is the Pearson correlation coefficient (Guyon and Elisseeff, 2003).

Consistency-based criteria: The basic idea behind the consistency-based criteria is that, the selected features must be consistent with the data set for the purpose of predicting the class values of the instances in the data set. That is, no two instances may have the same values on all predicting features if they have different class values. A basic consistency-based criterion was first used in FOCUS (Almuallim and Dietterich, 1991), as what they called the sufficiency test. Liu et al. (1998) proposed another consistency criterion which is independent of the search process

in feature selection. Many other kinds of consistency-based criteria has also been developed, a detailed review of them can be found in Arauzo-Azofra et al. (2008).

Classification error rate-based criteria: The classification error rate-base criteria is to evaluate a feature or a set of features through the classification error rate (or accuracy) estimated on the training data set (see Section 2.4 for more details about the classification error estimation).

3.2.2 Subset Search Strategies

To generate a subset of d features from a total of p features, we can use either the “bottom-up” method which starts with the empty feature set and builds up a subset of d features incrementally, or the “top-down” method that starts with the full feature set and removes noisy and redundant features successively.

Given a feature subset evaluation criterion Q as defined in Equation (3.2), and the assumption that the greater the Q value is, the better a feature subset is, then a feature subset search strategy is to iteratively search in the feature subset space a combination of features that will maximize the criterion Q . There are two kinds of search strategies including optimal and suboptimal strategies.

One optimal strategy is the exhaustive search which explore every possible combination of d features from the original p features. This strategy will always achieve the “best” feature subset under a given evaluation criterion but generally with a price of huge computational cost, especially when p is large. This is easy to understand as there will be $\binom{p}{d}$ combinations when selecting d features from p features.

Another optimal search strategy is the branch and bound algorithm (see Webb, 2002, page 312). This algorithm requires that the evaluation criterion satisfies the

monotonicity property, that is, for two feature subsets $\mathbf{X}$ and $\mathbf{Y}$:

$$\mathbf{X} \subset \mathbf{Y} \Rightarrow Q(\mathbf{X}) < Q(\mathbf{Y}) \quad (3.3)$$

When considering the computational feasibility, we often use a suboptimal search method which makes a trade-off between the computational complexity and the quality of the feature subset. The branch and bound algorithm may not be appropriate when the monotonicity property does not hold. It may also be computationally infeasible since the number of possibilities to be examined is an exponential function of the number of features. Suboptimal search methods that do not examine every combination of features is a practical choice when the number of candidate features is huge. The following part of this section briefly reviews some suboptimal search methods.

Best Individual d

The best individual d may be the simplest method for searching a feature subset. It is used when a feature evaluation criterion individually assesses each feature in the original feature set $\mathbf{X} = \{X_1, X_2, \dots, X_p\}$. Thus, the features are ordered as a feature ranking $\mathbf{R} = \{X'_1, X'_2, \dots, X'_p\}$ so that¹:

$$Q(X'_1) \geq Q(X'_2) \geq \dots \geq Q(X'_p) \quad (3.4)$$

where $X'_j \in \mathbf{X}$ and the top d features will be selected as the d -feature subset.

¹Note that a feature subset containing one feature X_j is $\{X_j\}$, and thus the evaluation of feature X_j should be $Q(\{X_j\})$. For simplicity, we use $Q(X_j)$ instead of $Q(\{X_j\})$ in this thesis.

In some cases this method can produce reasonably good feature subsets, especially when the features in the original feature set are uncorrelated. However, if the features in the original feature set are correlated, the selected feature subset will be suboptimal, as the discrimination information involved in a feature may be covered by other features in the subset. Therefore, some individually informative features will become non-informative in the context of a feature subset.

Sequential Forward Selection

Sequential forward selection (SFS) is a bottom-up search strategy that adds only one new feature into a feature subset at a time until the final feature subset is acquired. Suppose a feature subset with k features, $\mathbf{F}^{(k)}$, has been obtained, each of the remaining features in $\mathbf{X}$, say a feature X_j , is evaluated as:

$$Q(X_j) = Q(\mathbf{F}^{(k)} + X_j) \quad (3.5)$$

where $\mathbf{F}^{(k)} + X_j$ denotes the feature subset after adding a feature X_j to the feature subset $\mathbf{F}^{(k)}$.

The feature with the maximal evaluation value will be selected as the one adding to $\mathbf{F}^{(k)}$ to form a feature subset $\mathbf{F}^{(k+1)}$. The main disadvantage of this method is that it does not include a mechanism for deleting the features from the current feature subset once further added features make them non-informative.

When a number of r ($r > 1$) features are added each time, the above method becomes the generalized sequential forward selection (GSFS).

Sequential Backward Selection

The sequential backward selection (SBS) or sequential backward elimination, is a top-down search strategy analogous to SFS. Feature set is initialized as the original feature set, and one feature is deleted from the current selected feature subset until the final feature subset is acquired. Suppose a feature subset with $k + 1$ features, $\mathbf{F}^{(k+1)}$, has been obtained, each feature in $\mathbf{F}^{(k+1)}$, say a feature X_l , is evaluated as:

$$Q(X_l) = Q(\mathbf{F}^{(k+1)} - X_l) \quad (3.6)$$

where $\mathbf{F}^{(k+1)} - X_l$ denotes the feature subset after removing a feature X_l from the feature set $\mathbf{F}^{(k+1)}$.

The feature with the maximal evaluation value (i.e. decrease Q the least) is selected and deleted from $\mathbf{F}^{(k+1)}$ to form a feature subset $\mathbf{F}^{(k)}$. The process is repeated until a stopping criterion, e.g. a predefined number d features, is satisfied. In addition to the same disadvantage of SFS, this method has another disadvantage over SFS that it demands more computational burden since the evaluation function Q is evaluated over larger sets of features.

When a number of r ($r > 1$) features are deleted each time, the above method becomes the generalized sequential backward selection (GSBS). A special and widely used GSBS is in the famous feature selection algorithm SVM-RFE (support vector machine recursive feature elimination) where half of the left features are deleted in each iteration Guyon et al. (2002).

Plus l -Take Away r Selection

This is a search strategy that allows some backtracking in the feature selection process. If $l > r$, it is a bottom-up procedure. l features are added to the current feature subset using SFS and then the worst r features are removed from the obtained feature subset using SBS. This strategy conquers the problem of nesting as existed in SFS or SBS. If $l < r$, then this strategy becomes a top-down procedure, starting with the full original feature set, removing r then adding l features until a stopping criterion is satisfied.

When the GSFS and GSBS are used in the above strategy, then the method becomes the generalized plus l -take away r selection.

Floating Search Methods

The floating search methods, including sequential forward floating selection (SFFS) and sequential backward floating selection (SBFS), may be regarded as a development of the plus l -take away r selection in which the values of l and r are allowed to 'float', that is they may change at different stages of the selection procedure.

Suppose at stage k we have obtained k feature subsets $\mathbf{F}^{(1)}, \mathbf{F}^{(2)}, \dots, \mathbf{F}^{(k)}$ of sizes 1 to k . Let the corresponding evaluation values of these k feature subsets be $Q_1, Q_2, \dots, Q_k$. At the k -th stage of the SFFS, do the following:

1. Select the feature X_j from $\mathbf{X} - \mathbf{F}^{(k)}$ that increases the value Q the greatest and add it to the current subset, $\mathbf{F}^{(k+1)} = \mathbf{F}^{(k)} + X_j$.
2. Find the feature, X_i , in the current subset, $\mathbf{F}^{(k+1)}$, that reduces the value Q the least; if this feature is the same as X_j then set $Q_{k+1} = Q(\mathbf{F}^{(k+1)})$; increment k ;

go to step 1; otherwise remove it from the subset to form $\mathbf{F}^{(k)'} = \mathbf{F}^{(k+1)} - X_i$.

3. Continue removing features from the subset $\mathbf{F}^{(k)'}$ to form reduced subsets $\mathbf{F}^{(k-1)'}$ while $Q(\mathbf{F}^{(k-1)'}) > Q_{k-1}$; $k = k - 1$; or $k = 2$; then continue with step 1.

The above algorithm is initialized by setting $k = 0$ and $\mathbf{F}^{(0)} = \emptyset$ and using the SFS method until a set of size 2 is obtained.

3.2.3 Stopping Criteria

A stopping criterion in a feature selection procedure determines when to jump out the loop as shown in Figure 3.1. Without a suitable stopping criterion the feature selection process may run exhaustively or forever through the space of subsets. Search strategies and evaluation criteria can both give influence to the choice for a stopping criterion.

Stopping criteria based on a search strategy include:

1. whether a predefined number of features (e.g. d features) are selected; and
2. whether a predefined number of iterations reached (to restrict the run time).

Stopping criteria based on an evaluation criterion can be:

1. whether addition (or deletion) of any feature does not produce a better subset (i.e. the goodness of any candidate feature subset will not increase); and
2. whether an optimal subset according to some evaluation criterion is obtained.

3.2.4 Result Validation

Strictly speaking, the result validation procedure is not a part of the feature selection process itself, but a feature selection method (in practice) must be validated. It tries to test the validity of the selected feature subset by carrying out different tests, and comparing the results with previously established results, or with the results of competing feature selection methods using artificial data sets, real-world data sets, or both. A more general way to validate the selection result in literature is through the result of evaluating a pattern recognition, for example the classification performance with the selected feature subset in a pattern classification system. For more details of the classification performance evaluation, readers can refer to Section 2.4 in the previous chapter. Another way to assess the selection result is through the robustness which is introduced in Section 3.4 and 3.5.

3.3 Feature Selection Techniques

The existing feature selection techniques typically fall into two categories: feature ranking and feature subset selection (Guyon and Elisseeff, 2003). As indicated by their names, feature ranking (or ranking-based feature selection) is to rank features by differentiating the importance between features, while the feature subset selection (or set-based feature selection) assesses a subset of features as a group and tries to select a “best” feature subset under some evaluation criterion. Depending on whether or how the feature selection process interacts with a classification algorithm, feature subset selection techniques are further broken into three types: *filter methods*, *wrapper methods* and *embedded methods* (Kohavi and John, 1997; Guyon

and Elisseeff, 2003; Saeys et al., 2007).

As pointed out by Guyon and Elisseeff (2003), feature ranking is actually a filter method, by which a ranking of features can often be produced. In the following part of this section, we discuss and review the existing feature selection techniques according to filter, wrapper and embedded methods.

3.3.1 Filter Methods

The term *filter* was first introduced by John et al. (1994). Filter methods use the intrinsic merits of the data to assess the features without specifying the learning algorithm. They often employ a heuristic approach where the evaluation criterion is not directly linked to the performance of a particular learning algorithm. Therefore, features selected by filter methods are completely independent of the learning algorithm. This kind of methods will provide a general approach to feature selection making the solution suitable for a large family of learning algorithms. The commonly used evaluation criteria include the probabilistic distance measures, Pearson correlation, entropy, or other information-theoretic measures such as Markov Blanket (Koller and Sahami, 1996; Aliferis et al., 2003). Filter methods have two subclasses: univariate and multivariate (Saeys et al., 2007). Univariate filter methods evaluate each feature separately and thus ignore the dependencies among features, while multivariate filter methods consider the incorporation of feature dependencies to some degree but are slower than univariate methods.

Because of their independence to the learning algorithms, filter methods generally are simple and fast to execute, which makes filter methods easily scale to very high-dimensional problems such as microarray data where other methods may be

infeasible due to their computational complexity. However, a disadvantage of filter methods also arises from their non-interaction with learning algorithms. That is, the optimality of the selected features does not necessarily guarantee the best possible performance for a specific learning algorithm.

Numerous filter methods have been proposed in literature and in the following part we will give a brief review of the popular methods according to the evaluation criteria as discussed in Section 3.2.1.

Filter Methods Based on Information, Dependence and Consistency

Hall and Smith (1998) proposed a Correlation based Feature Selection algorithm (CFS) under the hypothesis that *Good feature subsets contain features highly correlated with (predictive of) the class, yet uncorrelated with (not predictive of) each other*. This hypothesis is interpreted by a heuristic criterion from psychological measurement:

$$J_{CFS}(\mathbf{F}^{(k)}) = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \quad (3.7)$$

where $J_{CFS}(\mathbf{F}^{(k)})$ is the merit of feature subset $\mathbf{F}^{(k)}$ containing k features; $\overline{r_{cf}}$ and $\overline{r_{ff}}$ are averaged class-feature correlation and feature-feature correlation, respectively. Their further analysis to this algorithm shows that the algorithm is quite aggressive in reducing the problem dimensionality with performance of learning algorithms designed with the selected features seeming to be competitive to wrappers (Hall, 1999, 2000). However, CFS can not identify strong associations between features because it evaluates features independently.

Following the idea of Hall and Smith (1998), Yu and Liu (2003) developed their

algorithm of Fast Correlation-Based Filter (FCBF) based on the concept of entropy and information gain which are both more powerful in capturing non-linear relationships between features. FCBF introduces a new framework of decoupling the relevancy and redundancy analysis into two steps including selecting a subset of class-correlated features and then selecting the predominant features from the previous subset of class-correlated features by filtering the redundant features. The algorithm is shown to be faster in speed and comparable or better in performance to several other examined methods including a faster but less accurate variant of the CFS algorithm proposed by Hall and Smith (1998).

Ding and Peng (August 2003) incorporated feature relevance and redundancy in the process of feature evaluation and proposed the minimal-redundancy-maximal-relevance (MRMR) feature selection framework. They realized their framework by simply combining feature relevance and redundancy through maximization of their difference or quotient for both discrete and continuous features. In their further work, Peng et al. (2005) theoretically proved the equivalence between the max-dependency criterion and the difference-based MRMR criterion when mutual information is used as the measure of both redundancy and relevance. Based on the MRMR with mutual information measure, they developed a two-stage feature selection algorithm which combines MRMR and other more sophisticated feature selectors such as wrappers to select more compact feature subsets.

Deisy et al. (2010) proposed a novel information theoretic-based interact (IT-IN) algorithm, which concerns the relevance, redundancy, and consistency of the features. The algorithm is implemented in two stages, where the stage one finds the feature relevance, which is ranked by information theoretic-based heuristic value and

the stage two handles the feature interaction through consistency measure, which is used to find the feature subset.

Filter Methods Based on Margins or Distances

Margins or equivalently Distances are very important indexes to indicate the confidence of a classifier with respect to its predictions. Two kinds of margins exist (Crammer et al., 2002) including the *sample margin* and the *hypothesis margin*. The sample margin is the distance of a sample to the decision boundary of a classifier just as used in support vector machines (Cortes and Vapnik, 1995); while the hypothesis margin, which is used in the AdaBoost (Freund and Schapire, 1997), is the distance that the classifier can travel without changing the way it labels any of the sample points.

The Relief algorithm may be one of the most discussed filter method that was proposed by Kira and Rendell (1992). The algorithm is based on the hypothesis margin of a 1-NN classifier and it is a randomized technique imposing ranking of features. In principle, Relief accumulates difference distances between randomly selected samples and their nearest neighbors from the same class (*nearest hit*) and from the different class (*nearest miss*) and then sorts features according to the accumulated values. A more detailed description of Relief is given in Section 4.3.3. The basic Relief was later extended to RELIEF-F for multi-class problems and mixed-typed input features by Kononenko (1994) and RELIEF-S that integrates sample selection for data with huge number of patterns by Liu et al. (2004). Some other variants of Relief can be found in Koller and Sahami (1996); Liu et al. (2002); Robnik-Sikonja and Kononenko (1997, 2003).

Furthermore, most existing algorithms are not single-criterion based (i.e. not just based on one kind of criterion such as distance) but usually in a hybrid form. For example, Zhou et al. (2010) improve the trace-based class separability criterion by imposing extra redundancy constraints to avoid selecting redundant features and propose the redundancy-constrained feature selection (RCFS) algorithm.

3.3.2 Wrapper Methods

Popularized by Kohavi and John (1997), wrapper methods offer a simple and powerful way to address the problem of feature selection. Wrapper methods use a learning algorithm to measure the goodness of a subset of features without incorporating knowledge about the specific structure of the learning algorithm, and can therefore be combined with any learning algorithms. For example, in pattern classification, a feature subset is evaluated by the classification performance (e.g. error rate) estimated on the training data with an expectation to generalize to unknown testing data. Therefore, in a wrapper method, the learning algorithm acts as a *black box* which makes wrappers remarkably universal and simple. Moreover, the feature subsets selected by a wrapper method will be very specific to the learning algorithm and are of course optimized for the preselected learning algorithm but may not be optimal for another learning algorithm.

To search the space of all feature subsets, a search strategy is then “wrapped” around the learning model. However, as the space of feature subsets grows exponentially with the number of features, when applied to very high-dimensional problems, wrapper methods might become computationally very expensive and even intractable. Another drawback of wrapper methods is that they have a higher risk

of overfitting than filter methods. Past research, e.g. the work by Reunanen (2003), showed that coarse (rather than delicate) search strategies such as sequential forward search (SFS) might not only reduce the computational burden, but also alleviate the problem of overfitting.

Before the appearance of the term “wrapper”, several work which are wrapper methods existed. For example, Moore and Lee (1994) proposed a feature subset selection model using the instance-based algorithm and the leave-one-out cross-validation (LOOCV) to assess candidate feature subsets. Langley and Sage (1994a) used a similar method but with nearest neighbour as the classifier. Caruana and Freitag (1994) tested the forward and backward stepwise methods on the calendar apprentice domain using the wrapper model and a variant of ID3 as the classifier.

John et al. (1994) put forth the term “wrapper” and used it as one of the general categories for grouping feature selection algorithms in the context of machine learning. They employed the k -fold cross-validation ($k = 25$ in their algorithms) with the learning algorithm ID3 and C4.5 for evaluating candidate feature subsets and the sequential forward and backward search to generate candidate feature subsets. The ID3-based and C4.5-based wrapper methods were tested on 9 (3 artificial and 6 real-world) data sets, and the experimental results showed that feature selection did not significantly change the generalization performance but with a main advantage of obtaining reduced structures of the decision trees.

Langley and Sage (1994b) proposed the selective naive Bayes which uses a wrapper method to remove redundant features. The wrapper method was implemented using naive bayesian classifier, the hold-out validation and the sequential forward search for feature subset selection. Their method was tested on six real-world data

sets and the experimental results showed that selective Naive Bayes significantly outperformed plain naive Bayes when redundant features were contained. Some other naive Bayes-based wrapper methods for feature subset selection have also been reported, see for example Kohavi (1995); Kohavi and John (1997).

Feature selection has already been extended to many specific domains, such as the microarray domain where gene selection is usually termed. Several researchers have applied wrapper methods for gene selection (Li et al., 2001; Xiong et al., 2001; Ooi and Tan, 2003; Cho et al., 2004; Zhou and Mao, 2005). Inza et al. (2002) implemented wrapper using the leave-one-out cross-validation and four well known classification algorithms including IB1 (Aha et al., 1991), Naive Bayes (Duda et al., 2001), C4.5 (Quinlan, 1993) and CN2 (Clark and Niblett, 1989). They carried out experiments on three gene expression data sets and indicated that by using gene selection, the classification accuracy was significantly improved and the number of genes used for building the classification models was notably reduced.

3.3.3 Embedded Methods

In contrast to filter and wrapper methods, in embedded methods the learning part and the feature selection part can not be separated - the structure of the learning algorithm (e.g the classification function of a classifier) under consideration plays a crucial role. For example, decision trees such as C4.5 (Quinlan, 1993), have a built-in mechanism to perform feature selection (with remaining nodes in the tree composing the selected feature subset)- this feature selection is very specific to the special structure of C4.5 trees and thus it is not suitable to other leaning methods such as Bayesian classifiers.

Embedded methods have the advantage that they include the interaction with the classification model just as wrapper methods do, but they may present superiority in several respects: they make better use of the available data without the need to split the training data into training and validation sets; they reach a solution faster by avoiding retraining a predictor from scratch for every variable subset investigated.

Decision trees are iteratively built by splitting the data depending on the value of a specific feature. The “splitting” feature is chosen according to its importance measured by some criterion such as information gain (Hunt et al., 1966) and gini index (Breiman et al., 1984). In many cases, only a subset of the features is included into a decision tree. Thus, feature selection is implicitly built into the algorithm and therefore decision trees can be understood as an embedded method.

SVM-based methods have formed a cluster of embedded feature selection methods (Weston et al., 2000; Guyon et al., 2002; Rakotomamonjy, 2003; Weston et al., 2003; Neumann et al., 2004). The SVM-RFE (Support Vector Machine Recursive Feature Elimination) is the most popular one as it is often used as a benchmark method for feature subset selection in literature (Furlanello et al., 2005; Tang et al., 2007). After trained on training data, SVM (linear) obtains a weight vector $\mathbf{w}$ containing weights w_i for all involved features, the square (or absolute) of which (i.e. w_i^2 or $|w_i|$) can be an index to measure the importance of a feature. In each round of SVM-RFE, Guyon et al. (2002) proposed to rank all remaining features by training a SVM and then eliminate the worst one. This elimination process is continued until a predefined number of features are retained. The algorithm can be accelerated by removing more than one feature in each round. In their experimental studies

on two gene expression data sets, Guyon et al. (2002) implemented SVM-RFE by eliminating half of the left features in each round.

3.4 Robustness Issue in Feature Selection

Feature selection has become more and more important with the abundance of very high dimensional data sets. It is now a prerequisite in many practical areas, e.g. microarray gene expression data analysis, mass spectra analysis and text mining. Most of the existing literature about feature selection focuses on its classification performance, i.e. selecting a subset of features with the cardinality as small as possible, to obtain the classification performance as good as possible. One important but relatively overlooked issue in past research of feature selection is the robustness (stability) of feature selection algorithms.

Robustness or stability is to measure the sensitivity of the solutions generated by a model to the changed conditions (e.g. inputs to the model or parameters of that model). Like the stability of a classification algorithm given by Turney (1995), the stability of a feature selection algorithm can be defined as the degree to which the feature selection algorithm generates repeatable selection results (feature subsets or feature rankings), given different batches of data from the same process. In practice, different batches of data are usually obtained by repeatedly splitting the data set into training and testing parts (e.g. by resampling with replacement), as the data distribution is unknown and only a limited number of samples are available during the learning process.

Robustness of feature selection is very crucial in many practical applications,

especially when the subsequent feature analysis and validation process is needed but it is quite time-consuming or money-costly to carry out that process. A much stabler selection result can give more confidence to domain experts on those selected features, thus can alleviate the time and money costs for the feature validation. Take analysis of microarray genes expression data for example, biological researchers need to find out which group of genes is really responsible for the generation of corresponding cancers. If the feature selection algorithm is very unstable, it may produce very distinct groups of genes accountable for the same cancer when only slight differences occur between the samples it studies. The distinct gene selection results thus produced could cause confusions to biological researchers and result in loss of their enthusiasm and confidence in applying machine learning techniques to solve biological problems. To improve the confidence, a very large group of genes need to be retained in the selection result (note that the more genes are retained, the more similar the selected gene groups will be), thus biological researchers need to conduct a great number of experiments to validate all those genes in a large group, which is both time and money expensive.

As pointed out by Krížek et al. (2007), the intrinsic reason for instability in feature selection is that the feature scores assigned by a feature selection algorithm to express the importance of features are random values, and the randomness of those values originates from the fact that the training samples input to a feature selection algorithm are randomly sampled. The randomness of the training samples cannot be avoided in practice. Therefore, a feature selection algorithm outputting more consistent selection results will be preferable to the one generating highly volatile results under same data variations.

Although robustness of feature selection has great significance, we must emphasize here that there is no evidence indicating a good robustness will necessarily lead to a good classification performance. Feature selection results should be investigated together with classification performance (such as accuracy), because domain experts are not interested in a strategy that yields very stable feature subsets, but leads to a bad predictive model (e.g., arbitrarily picking the same set of features under training data variation). In practice, a balance between classification and robustness performance is usually achieved.

Although robustness of feature selection for high-dimensional and small-sized problems has received attention in recent years, in the literature, just a few work exists. We give a review as below.

To our knowledge, the work done by Dunne et al. (2002) may possibly be the earliest work addressing the robustness issue in feature selection as defined and explained above. They illustrated the instability of some wrapper methods by examples and proposed an ensemble-like remedy which is based on the aggregation of multiple runs of sequential search. A robustness measure based on Hamming distance was also proposed. Their comparative studies with and without aggregation showed the superiority of the proposed methods in improving robustness with probably slightly better classification performance.

The ensemble idea in feature selection is similar to ensemble learning for classification (Dietterich, 2000): a number of different feature selectors are first used, and then the outputs of these separate selectors are aggregated to form the final (ensemble) result. Saeys et al. (2008) extended this idea to other filter and embedded feature selection methods and proposed an instance perturbation-based ensemble

scheme for single feature selection algorithm. Their experimental results of four feature selection algorithms on six gene expression data sets exhibited the great promise of the ensemble scheme in improving robustness of feature selection. Abeel et al. (2010) further applied the concept of ensemble feature selection introduced by Saeys et al. (2008) to the robust biomarker discovery in bioinformatics. They focused on the analysis of ensemble feature selection techniques using linear SVMs and RFE (Recursive Feature Elimination) as the feature selection mechanism, and they showed by the experimental results that the robustness of SVMs for biomarker identification can be substantially increased by using ensemble feature selection techniques, while at the same time improving upon classification performances. Another research direction of ensemble in bioinformatics is the aggregation of data from different institutions and platforms for biomarker discovery, for example, the work by Wang et al. (2006).

A few other work aiming at enhancing robustness with novel ideas different from ensemble can also be found. Yu et al. (2008) proposed a general feature selection framework that identifies dense feature groups based on kernel density estimation, and developed an efficient algorithm DRAGS (Dense Relevant Attribute Group Selector) which can directly provide stable feature selection results. In their further work, Loscalzo et al. (2009) proposed a consensus group-based framework which alleviates the problem of small sample size, and developed Consensus Group Stable Feature Selection (CGS) algorithm. Both DRAGS and CGS could achieve good robustness without sacrificing classification performance.

In addition to the above efforts of improving robustness of feature selection, some comprehensive comparisons of the robustness of various existing feature selection

algorithms have been conducted. Kalousis et al. (2007) examined three categories of existing stability measures (Pearson's correlation coefficient, Spearman's rank correlation coefficient and Tanimoto distance based metric) in high dimensional space and compared the robustness of a number of feature ranking and weighting algorithms under training data variation. Their experimental results demonstrated that different algorithms which performed similarly well on classification had a wide difference in terms of robustness. Therefore, their work constructed robustness profiles for several well-known feature selection algorithms which can then be used to provide advice for choosing feature selection algorithms for practical problems. Gulgezen et al. (2009) studied the robustness and classification accuracy of the MRMR (Minimum Redundancy Maximum Relevance)-based feature selection. They implemented MRMR with two feature evaluation criteria: MID (Mutual Information Difference) and MIQ (Mutual Information Quotient), and verified by results on different data sets that the two criteria produced similar accuracies, but MID is more stable.

Besides the above work, other contributions to new and efficient robustness measures also exist. Křížek et al. (2007) developed an Shannon entropy-based measure for assessing robustness of feature subsets. Kuncheva (2007) proposed a consistency measure based on cardinality of the intersection and a correction for chance. Somol and Novoviov (2008) proposed another new consistency measure and its variants with reduced computational complexity, and compared them with the measures introduced in Kalousis et al. (2007).

3.5 Evaluation of Robustness

As it can be found in the past work, the robustness of feature selection was mostly assessed based on the similarity between a pair of feature selection results (Dunne et al., 2002; Kalousis et al., 2007; Kuncheva, 2007; Křížek et al., 2007). In this section, we first discuss methods to gauge similarity between two selection results, and then we give an introduction to some popular robustness measures based on similarity.

3.5.1 Similarity

Different feature selection algorithms give different types of feature selection results including feature weighting, feature ranking and feature subset. In the first type, a weight or score is assigned to each feature indicating its importance. The second type of representation is a simplification of the first where instead of weights, ranks are assigned to features indicating their order in a feature ranking. The third type consists of a subset of selected features in which no weighting or ranking is considered. Obviously, any weighting schema can be cast as a ranking schema, which, in turn, can be cast as a set of features by setting a threshold on the ranks or asking for a given number of features.

More formally, suppose the original data is described by p features denoted as $\mathbf{X} = \{X_1, X_2, \dots, X_p\}$. For feature weighting, the similarity $J_{ij}(p)$ between two weightings, $\mathbf{w}_i = (\mathbf{w}_{i1}, \mathbf{w}_{i2}, \dots, \mathbf{w}_{ip})$ and $\mathbf{w}_j = (\mathbf{w}_{j1}, \mathbf{w}_{j2}, \dots, \mathbf{w}_{jp})$ produced by a feature selection algorithm from the i -th and j -th batch of data, can be measured

using Pearson's correlation coefficient:

$$J_{ij}(p) = \frac{\sum_{l=1}^p (\mathbf{w}_{il} - \mu_{\mathbf{w}_i})(\mathbf{w}_{jl} - \mu_{\mathbf{w}_j})}{\sqrt{\sum_{l=1}^p (\mathbf{w}_{il} - \mu_{\mathbf{w}_i})^2 \sum_{l=1}^p (\mathbf{w}_{jl} - \mu_{\mathbf{w}_j})^2}} \quad (3.8)$$

where $\mathbf{w}_{il}$ is the weight of feature X_l in feature weighting $\mathbf{w}_i$ and $\mu_{\mathbf{w}_i}$ is the mean of all weights in $\mathbf{w}_i$. The similarity $J_{ij}(p)$ takes values in $[-1, 1]$: a value of 1 means that the weightings are perfectly correlated; a value of 0 means that there is no correlation; a value of -1 means that they are anti-correlated.

For two feature ranking results, $\mathbf{r}_i = (\mathbf{r}_{i1}, \mathbf{r}_{i2}, \dots, \mathbf{r}_{ip})$ and $\mathbf{r}_j = (\mathbf{r}_{j1}, \mathbf{r}_{j2}, \dots, \mathbf{r}_{jp})$, their similarity can be measured by Spearman's correlation coefficient:

$$J_{ij}(p) = 1 - 6 \sum_{l=1}^p \frac{(\mathbf{r}_{il} - \mathbf{r}_{jl})^2}{p(p^2 - 1)} \quad (3.9)$$

where $\mathbf{r}_{il} \in \{1, 2, \dots, p\}$ is the order of feature X_l in feature ranking result $\mathbf{r}_i$. Here too, the possible range of values is in $[-1, 1]$. A value of 1 means that the two rankings are identical; a value of 0 means that there is no correlation between the two rankings; a value of -1 means that they have exactly inverse orders.

Finally, for two feature subsets $\mathbf{F}_i^{(k)}$ and $\mathbf{F}_j^{(k)}$ produced from the i -th batch j -th batch of data, their similarity can be measured by the Jaccard Index:

$$J_{ij}(k) = \frac{|\mathbf{F}_i^{(k)} \cap \mathbf{F}_j^{(k)}|}{|\mathbf{F}_i^{(k)} \cup \mathbf{F}_j^{(k)}|} \quad (3.10)$$

where k is the cardinality of $\mathbf{F}_i^{(k)}$ and $\mathbf{F}_j^{(k)}$, $|\mathbf{F}_i^{(k)} \cap \mathbf{F}_j^{(k)}|$ is the number of common features between $\mathbf{F}_i^{(k)}$ and $\mathbf{F}_j^{(k)}$ and $|\mathbf{F}_i^{(k)} \cup \mathbf{F}_j^{(k)}|$ is the total number of features without reduplication in $\mathbf{F}_i^{(k)}$ and $\mathbf{F}_j^{(k)}$. $J_{ij}(k)$ takes values in $[0, 1]$, with 0 meaning

that there is no overlap between the two sets and 1 that the two sets are identical.

3.5.2 Robustness Measures Based on Similarity

In a pattern classification system, we generally need to produce multiple, e.g. totally T , batches of training data (and thus T results) to obtain a much confident estimate of the final performance. After defining the similarity measures between two selection results, we need to estimate the total robustness. There are generally two ways to do it, one is the pairwise way and the other is the one-to-all way. We will take the Jaccard Index for example to show how to compute the total robustness.

Total Robustness based on Pairwise Similarity

This kind of total robustness is defined as the average over all pairwise similarities between the different feature selection results:

$$J_{total}(k) = \frac{2 \sum_{i=1}^T \sum_{j=i+1}^T J_{ij}(k)}{T(T-1)} \quad (3.11)$$

Total Robustness based on One-To-All Similarity

In practice, the training data are usually generated by resampling techniques such as resampling with replacement in bootstrap. Therefore, each training data only contains part of the whole data. In the one-to-all way, “one” means the result produced on the whole data (denoted with a subscript “0”, e.g. $\mathbf{F}_0^{(k)}$ for a feature subset selected with all samples in the data set) and “all” denotes all the T results generated with the T batches of training data. The similarity between $\mathbf{F}_0^{(k)}$ and $\mathbf{F}_i^{(k)}$,

i.e. $J_{i0}(k)$, is first calculated and then all T similarities are averaged:

$$J_{total}(k) = \frac{1}{T} \sum_{i=1}^T J_{i0}(k) \quad (3.12)$$

Chapter 4

Improving Robustness of Feature Ranking by Multi-Criterion Combination with Novel Feature Importance Transformation

4.1 Introduction

Feature ranking is an important category of feature selection techniques and plays its role in data mining due to its simplicity and computational efficiency. In addition, in some research areas where the purpose is to find target-related features, e.g. identifying biomarkers in microarray gene expression data analysis, feature ranking will be much preferred to set-based feature selection techniques.

Generally speaking, there are two types of inconsistency problems in feature ranking (and also in feature subset selection). One is from the perturbations of training data such as adding or removing a few samples in the training data, after which different feature selection results will be produced by the same feature

ranking criterion (algorithm). The other inconsistency originates from different feature ranking criteria which generate much distinct feature ranking results. The inconsistencies could bring troubles to researchers especially to domain experts. For example, the inconsistent gene selection results could cause confusions to biological researchers in biomarker identification and even make them lose their enthusiasm and confidence in applying machine learning techniques to solve biological problems.

A simple but widely employed strategy to alleviate or solve inconsistency problems is to reach a consensus of opinions from multiple criteria through a combination process. For example, in pattern classification, the classification performance including accuracy and stability can generally benefit from combining multiple classifiers (Dietterich, 2000; Kim and Ko, 2005; Lzoray and Cardot, 2008). In feature selection, some work related to multi-criterion combination has already been done (see for example Yan (2007); Chen et al. (2007); Li and Zeng (2010) and the references therein). An extremely crucial but relatively neglected problem in multi-criterion combination is the score normalization. Since the feature importance scores from different feature ranking criteria are usually heterogeneous, score normalization is required to transform the scores into a comparable scale prior to combining them. Furthermore, different score normalization methods often affect the combination results greatly, especially in the situation of very high dimensionality. A well-designed score normalization could be beneficial to multi-criterion combination in feature selection.

Altmann et al. (2010) proposed the PIMP algorithm to compute the PIMP score as a new measure of feature importance which can reflect the real significance of features with low feature importance bias. Theoretically, this measure can be a good

choice for normalizing scores from different feature ranking criteria into a comparable scale. But when applied to the high dimensional and small sized gene expression data, the PIMP algorithm might encounter problems such as score saturation problem (for details please refer to Section 4.5). In this chapter, we exhibit and analyze the problems in existing combination and normalization methods, and develop the RPI (resampling and permutation feature importance) algorithm by advancing the PIMP algorithm after considering the existing problems. The RPI algorithm transforms the original feature scores computed by general feature ranking criteria to the RPI scores, which are then used for score normalization in multi-criterion combination. During the RPI score transformation, a resampling-based score correction was also employed. Experimental studies were conducted on four gene expression data sets and the experimental results showed that the combination based on RPI score produced gene rankings with improved robustness compared to those from other two commonly used combination methods.

4.2 Illustration of Inconsistency Problems in Feature Selection

As stated above, there are two types of inconsistency problems in feature selection. In this section, we illustrate these two inconsistency problems by experiments.

The first inconsistency problem relates to robustness of a feature selection algorithm. Ideally, a feature selection algorithm should be robust enough to produce very similar selection results even if the training data is subject to perturbations such as addition or deletion of a few samples. But this may not necessarily be true when

the data is with high dimensionality and small sample size. To illustrate this point, the following experiment was conducted. Firstly, a perturbation to training data was performed by randomly removing l samples from the training data to produce a perturbed training data. Secondly, Fisher's ratio-based feature selection algorithm (please refer to Section 4.3 for details) was applied to the perturbed training data to select top 10 features. Assume $\mathbf{F}_0$ and $\mathbf{F}_{-l}$ are the feature subsets selected using the original training data and the perturbed training data with l samples removed. The number of common features in $\mathbf{F}_0$ and $\mathbf{F}_{-l}$ reflects the sensitivity of the feature selection algorithm to training data perturbation. A feature selection algorithm is considered to be robust or insensitive to perturbation of training data if the features in $\mathbf{F}_0$ and $\mathbf{F}_{-l}$ are similar.

To obtain reliable evaluation, the above experiment was repeated 300 times for each l value, and the average number of common features in $\mathbf{F}_0$ and $\mathbf{F}_{-l}$ was calculated. The curve "FisherRatio: CNS data" in Figure 4.1 shows the results of Fisher's ratio-based feature selection on CNS gene expression data, which has 60 samples and 7129 genes (details of the data are given in Section 2.6.2 of Chapter 2). From Figure 4.1, it is observed that when only 1 sample is removed, the number of common genes among the top 10 genes is reduced from 10 to 8 on average, and the number is further reduced to 5 when 5 samples are removed. This result tells us that the commonly used Fisher's ratio is actually sensitive to training data perturbations.

To further investigate the robustness of Fisher's ratio-based feature selection, experimental study on simulated data was also conducted. Two artificial data sets containing 7129 features with 60 and 1000 samples were generated respectively using

Figure 4.1: Average number of common genes vs. number of samples removed when a subset of 10 genes is selected

the Matlab code in Guyon (2001). As shown in Figure 4.1 (the curve “FisherRatio: SimulationData60”), the feature selection results of Fisher’s ratio on the simulated data set with 60 samples are not stable. But for the data set with 1000 training samples (see the curve “FisherRatio: SimulationData1000” in Figure 4.1), the selection results are less sensitive to removal of training data. This result explains why robustness is not an issue in feature selection for data with large sample size. But for small sized data like gene expression data, the robustness of feature selection is indeed an issue that should be considered seriously because such a minor variation/perturbation of training data is very likely to occur in practice, such as addition of a few new cases or samples.

Figure 4.1 shows the results of Fisher’s ratio as an example, but our study found that almost all commonly used feature selection algorithms are prone to perturbation of small-sized training data.

In addition to the above inconsistency problem, the inconsistency in selection

results generated by different feature selection algorithms should also be paid attention to. To illustrate this inconsistency problem, experiments on full CNS data were carried out. Table 4.1 shows the top 10 genes (denoted by their serial numbers in the table) selected by five different feature selection algorithms, including Fisher’s ratio, Relief, ADC (asymmetric dependency coefficient), AW-SVM (absolute weight of SVM) and SVM-RFE (Guyon et al., 2002). The first four are ranking-based feature selection algorithms and the last is a set-based feature selection algorithm, details of the feature ranking algorithms are given in the following Section 4.3. Obviously, the selection results either between the ranking-based feature selection or between the set-based and ranking-based feature selection are mostly different. So far, there is no agreed ways to decide which result is superior and should be adopted, as their classification performances are data-dependent and often don’t make much difference.

Criterion	Top 10 genes									
	1	2	3	4	5	6	7	8	9	10
Fisher’ratio	327	348	2196	2695	2496	1352	1320	3320	844	3645
Relief	1431	1791	6165	1615	5978	3389	2912	43	1281	6891
ADC	4588	1320	3185	3783	327	2695	2854	1474	3731	4174
AW-SVM	4503	6252	3389	5812	2914	1697	2700	942	4546	4247
SVM-RFE	2671	5581	5389	2404	4576	5061	6555	2914	1478	2917

Table 4.1: Top 10 genes selected by five different feature selection criteria on CNS data with all 60 samples as training data

The inconsistency problems as shown in Figure 4.1 and Table 4.1 bring great challenges to researchers. There is still no feature selection algorithm which can overwhelm others in performance and how to solve the inconsistency problems is still under study. A widely accepted strategy is to integrate multiple feature selection

algorithms with expectation of alleviating or solving both the above inconsistency problems. The following parts are focusing on the multi-criterion combination for feature ranking algorithms and the criterion normalization is studied for improving the robustness of feature ranking.

4.3 Basis Criteria for Multi-Criterion Combination

Before the combination of multiple criteria, a set of feature selection criteria needs to be selected. There exist sufficiently large number of feature ranking criteria in literature. In multi-criterion combination process, however, it is unnecessary and impractical to employ all of the existing feature ranking criteria. In this study, the feature ranking criteria are manually selected before combination and they are termed *basis criteria*.

It was found in preliminary experiments that if two feature ranking criteria produce similar results, combination of such two criteria does not help. This finding provides us a rough guideline for basis criteria selection: basis criteria should exhibit diversity. The deeper reason for diversity is that the diverse results generated by multiple basis criteria may complement each other. Furthermore, the diversity in basis criteria can help to prevent the combination results from being dominated by criteria that produce similar results.

For computational simplicity and performance diversity analyzed above, the basis criteria used in this study are Fisher's ratio, ADC (asymmetric dependency coefficient), Relief and AW-SVM (absolute weight of SVM), which are of different

categories or based on different theoretical hypotheses. Fisher's ratio is a univariate filter method evaluating each feature individually, while Relief is a multivariate filter method taking into account dependencies between features. ADC is an information theory-based filter method, and AW-SVM is an embedded method that ranks features based on their corresponding coefficients in the SVM classifier. Details of the four basis criteria are presented below.

4.3.1 Fisher's ratio

Fisher's ratio is an individual feature evaluation criterion that measures the linear discriminating power of a feature X_j by the ratio of squared inter-class difference to intra-class spread:

$$FR(X_j) = \frac{(m_{j(1)} - m_{j(2)})^2}{\sigma_{j(1)}^2 + \sigma_{j(2)}^2} \quad (4.1)$$

where $m_{j(c)}$ and $\sigma_{j(c)}^2$ are, respectively the sample mean and variance of feature X_j within class c , for $c = 1, 2$. The larger the FR value a feature has, the more discriminative the feature is.

4.3.2 ADC

The Asymmetric Dependency Coefficient (ADC) is an information gain based feature ranking criterion that evaluates the discriminative power of feature X_j by its dependency to class label Y (Sridhar et al., 1998):

$$ADC(Y, X_j) = \frac{MI(Y, X_j)}{H(Y)} \quad (4.2)$$

where $H(Y)$ is the entropy of Y and $MI(Y, X_j)$ is the mutual information between label Y and feature X_j defined as:

$$H(Y) = - \sum_y p(Y = y) \log p(Y = y) \quad (4.3)$$

$$H(X_j) = - \sum_x p(X_j = x) \log p(X_j = x) \quad (4.4)$$

$$MI(Y, X_j) = H(Y) + H(X_j) - H(Y, X_j) \quad (4.5)$$

4.3.3 Relief

Relief is inspired by instance-based learning and it calculates the “Relevance” of features to targets through iterative updates (Robnik-Sikonja and Kononenko, 2003). In each iteration, a sample $\mathbf{x}$ is first randomly selected, and its nearest-Hit $\mathbf{x}_H$ (nearest neighbor from the same class) and nearest-Miss $\mathbf{x}_M$ (nearest neighbor from other class) are identified, and the relevance is then updated as:

$$W(X_j) = W(X_j) - \frac{\text{diff}(X_j, \mathbf{x}, \mathbf{x}_H)}{n} + \frac{\text{diff}(X_j, \mathbf{x}, \mathbf{x}_M)}{n} \quad (4.6)$$

where $W(X_j)$ is the relevance of feature X_j to the targets and it is initialized to zero; $\text{diff}(X_j, \mathbf{x}, \mathbf{x}')$ denotes the difference of feature X_j between samples $\mathbf{x}$ and $\mathbf{x}'$. For continuous features, $\text{diff}(X_j, \mathbf{x}, \mathbf{x}')$ is the actual difference normalized to interval $[0, 1]$:

$$\text{diff}(X_j, \mathbf{x}, \mathbf{x}') = \frac{|x_j - x'_j|}{x_{j \max} - x_{j \min}} \quad (4.7)$$

where x_j and x'_j are the values of feature X_j in $\mathbf{x}$ and $\mathbf{x}'$, $x_{j\max}$ and $x_{j\min}$ are the maximal and minimal values of feature X_j in all samples.

The updating process in equation (4.6) iterates n times for a training data set with n samples to get the final relevance.

4.3.4 AW-SVM

Support vector machine (SVM) is a popular classification algorithm suitable for high-dimensional data because of its insensibility to data dimensionality (Furey et al., 2000; Cortes and Vapnik, 1995). Among the several variants of the basic SVM, the *linear binary* SVM with soft margin is employed. The linear SVM classifier is in fact a hyper-plane defined by:

$$\sum_{j=1}^p w_j x_j + b_0 = 0 \quad (4.8)$$

where p is the total number of features and w_j is the weight of feature X_j . The weight w_j indicates the importance of feature X_j , and hence the absolute weight of SVM (AW-SVM) can be used to evaluate and rank the features. For more details about SVM, readers can refer to Section 2.3.3.

4.4 Basic Combination Methods

Generally, the multi-criterion combination methods are divided into two categories: combination based on feature scores and combination based on feature rankings. The latter, ranking-based combination, can be viewed as a special score-based com-

ination. Details are as below.

4.4.1 Score-Based Combination

In score-based multi-criterion combination, each basis criterion first produces a score vector containing scores of all features (e.g. the FR values in Fisher's ratio criterion), a score combination algorithm is then employed to aggregate the multiple score vectors into one consensus score vector, and a feature ranking is finally generated by sorting the features according to their consensus scores. The score-based multi-criterion combination procedure is illustrated in Figure 4.2.

Figure 4.2: Score-based multi-criterion combination

There are several schemes to generate the consensus score vector from multiple score vectors, for example, summation of scores, multiplication of scores, maximum or minimum of scores, etc. In our study, the popular score summation scheme is employed.

In score summation (as well as in other score combination schemes), it is vital to ensure that the scores from different basis criteria are comparable. Since scores generated by different basis criteria are usually heterogeneous, score normalization

should be accomplished before carrying out the score combination. Assume $\mathbf{u}_i$ is the score vector produced by basis criterion i , a widely used score normalization method is:

$$\mathbf{u}_i' = \frac{\mathbf{u}_i - \mathbf{u}_{i \min}}{\mathbf{u}_{i \max} - \mathbf{u}_{i \min}} \quad (4.9)$$

where $\mathbf{u}_{i \min}$ and $\mathbf{u}_{i \max}$ are the minimum and maximum values in vector $\mathbf{u}_i$.

Then the final consensus score vector from m basis criteria is:

$$\mathbf{u} = \frac{1}{m} \sum_{i=1}^m \mathbf{u}_i' \quad (4.10)$$

where m is the total number of basis criteria used in combination.

4.4.2 Ranking-Based Combination

Ranking-based multi-criterion combination is to generate a consensus feature ranking from multiple feature rankings produced by basis criteria. In the process of combination, a basis criterion first produces a feature ranking, where each feature has a *position* (or *order*) value; and then a ranking combination algorithm is applied onto all feature rankings to generate the final consensus ranking. The ranking-based multi-criterion combination procedure is illustrated in Figure 4.3.

Compared to the score-based combination in Figure 4.2, the ranking-based combination requires a basis criterion to produce a feature ranking rather than a feature score vector.

Ranking combination (usually termed as rank aggregation) is a common problem in many fields and has been extensively studied (Dwork et al., 2001; Hsu and Taksa,

Figure 4.3: Ranking-based multi-criterion combination

2005). Among the existing aggregation methods, *Borda count*, which is originally a voting method based on rankings, is a simple yet effective rank aggregation method (van Erp and Schomaker, 2000). Suppose there are m voters (i.e. basis criteria in this study) and a fix set of p candidates (i.e. features). In Borda count, voter i first gives points to all candidates to generate a point vector $\mathbf{v}_i$ as follows: the top ranked candidate is given p points, the second ranked candidate is given $p - 1$ points, and so on. The final points of candidates are the sum of points from the m voters:

$$\mathbf{v} = \sum_{i=1}^m \mathbf{v}_i \quad (4.11)$$

and the aggregated ranking is obtained by descendingly sorting the final points in $\mathbf{v}$.

4.5 Problems in Existing Methods

4.5.1 Domination Problem in Score Combination

In the process of multi-criterion combination, the domination problem is ubiquitous when basis criteria are heterogeneous. Because of the differences among basis criteria, the scoring scales between basis criteria are usually distinct, which makes one or a few basis criteria dominate. A broadly used strategy to solve this problem is score normalization, though it does not always guarantee its validity. To demonstrate this problem, the gene scores on full Colon data set are first calculated by each of the four basis criteria and then sorted in descending order. The score curves are shown in Figure 4.4.

Figure 4.4: Scores generated by each basis criterion on Colon data: (a) Original scores and (b) Scores after normalization.

Figure 4.4(a) shows the original scores while Figure 4.4(b) shows the scores after normalization using Equation (4.9). From Figure 4.4(a), it can be seen that the combination result will be dominated by Fisher'sRatio and Relief, and the other

two criteria, especially AW-SVM, will hardly affect the final combination result because of their low scores compared to those from Fisher'sRatio and Relief. After normalization (see Figure 4.4(b)), the domination problem will be alleviated for ranking about the top 100 genes, but the combination result may still be dominated by Relief for ranking the remaining genes.

4.5.2 Score Inaccuracy Problem under Small Sample Size

In practice, a feature ranking is produced by sorting feature scores computed by a feature ranking criterion using limited samples. If the feature scores are accurate or insensitive to sample variation, the feature rankings generated will be stable (e.g. the most stable case is that feature orders keep the same when samples alter). But for small-sized gene expression data, it is difficult or even impossible to obtain very accurate feature scores. To illustrate this point, the following experiment was conducted on Colon data. Firstly, a training data was generated by resampling with replacement on the full data set (62 samples) and secondly, a feature ranking criterion was applied to the training data to compute the scores of all genes. The above process was repeated 100 times and the box-plots of the scores of the top 10 genes are exhibited in Figure 4.5. The top 10 genes were obtained by descendingly sorting scores computed with full data set.

The box-plots of gene scores from two basis criteria, AW-SVM and ADC, are shown in Figure 4.5(a) and Figure 4.5(b), respectively. From the box-plots, it can be observed that the gene scores varied greatly when the training data changed, especially those by AW-SVM. Furthermore, if the genes are re-sorted by their median values (represented by the red line in each box), the gene order would change

Figure 4.5: Scores of top 10 genes on Colon data: (a) By AW-SVM and (b) By ADC.

greatly comparing to the one generated by scores computed with full data set. For instance, after re-sorting, the gene ranking of the 10 genes in Figure 4.5(b) becomes (8,7,2,1,4,10,6,9,5,3), which is rather different from the original one (1,2,3,4,5,6,7,8,9,10). This is also an indication of the score inaccuracy problem on small-sized gene expression data.

4.5.3 Score Saturation Problem in PIMP

The PIMP algorithm uses p -value (denoted as $Pvalue$ in following parts) to correct the importance of a feature based on the estimation of distribution in a non-informative setting. A feature's importance (score) with the PIMP algorithm is computed in two steps: 1) the label vector in a data set is permuted (which makes the features perform non-informatively) l times and then the non-informative or random importance distribution (Gaussian is assumed in this study) of the feature is estimated; 2) the $Pvalue$ of the feature's true score (i.e. score computed with the

original unpermuted label vector) is calculated and serves as the feature's corrected importance.

Under the situation of small sample size, there may exist risks for the PIMP algorithm to be of very low resolution or saturated when differentiating between the top-ranked features by assigning scores very close to the maximum value of 1. To illustrate the score saturation problem, the PIMP scores were computed on the full Colon data set with $l = 50$ and the scores in descending order are shown in Figure 4.6(a) for each basis criterion. Note that the PIMP scores in Figure 4.6(a) equal $(1 - Pvalue)$, and therefore the greater of the PIMP score, the more important of a gene.

Figure 4.6: PIMP scores from each basis criterion

For clearer illustration, the part in Figure 4.6(a) containing scores of the top 300 genes is zoomed in and shown in Figure 4.6(b). From the zoomed-in figure, it can be clearly seen that the PIMP scores of the top 50 genes by Relief as well as PIMP scores of the top 200 genes by Fisher's ratio and ADC are all approximately

1, which proves the existence of the score saturation problem.

Score saturation brings instability into the process of ranking features for single criterion as well as for multi-criterion combination. One possible reason for score saturation is the inaccurately estimated parameters of the random importance distribution under the situation of small sample size. For example, the parameter σ is downwardly biased for Gaussian distribution.

4.6 Resampling and Permutation-Based Feature Importance Transformation

The resampling and permutation-based feature importance (RPI) transformation is our efforts to deal with the aforementioned problems. The RPI algorithm comprises two steps: score correction and score transformation. The first step, score correction, is to correct the feature scores computed by the feature ranking criterion, while the second step is to transform the corrected scores into the RPI scores. Details of the algorithm are as follows.

(1) *Score correction.* This step is based on resampling of the training data and is adopted in consideration of the score inaccuracy problem. A number b of bags (sub-samples) are first generated using the resampling with replacement technique, and then a feature ranking criterion is performed on each of the bags. For every feature, this leads to a vector of b scores which are used to estimate the score distribution of the feature. For a feature, the relationship between its score S_0 (calculated with the full training data) and the score distribution (Gaussian is assumed, and the center S_r is the mean of the b scores) is depicted in Figure 4.7. The corrected score S_c is

defined:

$$S_c = S_0 + (S_r - S_0) \times |Pvalue - 0.5| \times 2 \quad (4.12)$$

Figure 4.7: Score distribution

(2) *Score transformation.* This step is to transform the corrected feature score S_c into the RPI score. The RPI score transformation uses resampling and permutation simultaneously, rather than only permutation as used in the PIMP algorithm, to estimate the random importance distribution. A bag is first generated by resampling of the training data and then the corresponding label vector of the bag is permuted. After estimating the random importance distribution, the *Pvalue* of the corrected score S_c is computed and the RPI score is $(1 - Pvalue)$.

Figure 4.8 shows the gene RPI scores from the four basis criteria on full Colon data set. Compared to the original scores in Figure 4.4(a) and the PIMP scores in Figure 4.6, it can be observed that the criterion domination problem and the score saturation problem have been greatly alleviated or solved.

Figure 4.8: RPI scores from each basis criterion

4.7 Experimental Studies

4.7.1 Robustness Measure for Feature Ranking

Here we propose to use a robustness measure for feature ranking results. This measure is based on the occurrences (frequencies) of features among multiple feature rankings that are produced with different batches of data. The details are described below.

Suppose there are in total T batches of data and R_i^k denotes the first k features in the feature ranking R_i which is produced by a feature ranking algorithm using the i^{th} batch of data. In the $T \times k$ feature ranking matrix R^k , the appearance times of each feature are counted and for the top k appeared features, their appearance times are $a_i, i = 1, 2, \dots, k$, then the robustness when selecting k features is:

$$J(k) = \frac{\sum_{i=1}^k a_i}{T \times k} \quad (4.13)$$

The above measure is effective and with highly improved time efficiency compared to the commonly used pair-wised Jaccard index employed in Saeys et al. (2008). It is similar to the consistency measure defined in Somol and Novoviov (2008), but with a difference that the above measure is designed for feature ranking results and uses only the top occurrent features rather than all features as used in Somol and Novoviov (2008).

4.7.2 Experimental Results and Discussions

The experiments were conducted on four widely used binary-class gene expression data sets: Colon data, Lymphoma data, Leukemia data and Prostate data. Details of the data sets can be found in Section 2.6.2 in Chapter 2.

The experimental results including the robustness and the estimated classification error rate of the multi-criterion combination based on original scores, Borda method and RPI scores (depicted in the figures as OriScore, Borda and RPI, respectively) on the four gene expression data sets are exhibited in Figure 4.9 to Figure 4.12.

From the comparison of robustness performance, it can be observed that gene ranking results produced through the RPI score combination perform the best on all the four gene expression data sets. For the classification performance, there are no significant differences on Colon and Leukemia data sets, while one exceptions occur on the other two data sets that the RPI score combination performs the best on DLBCL data and a little worse on Prostate data.

Figure 4.9: Experimental results on Colon data: (a) Robustness and (b) Classification error rate.

Figure 4.10: Experimental results on Lymphoma data: (a) Robustness and (b) Classification error rate.

Figure 4.11: Experimental results on Leukemia data: (a) Robustness and (b) Classification error rate.

Figure 4.12: Experimental results on Prostate data: (a) Robustness and (b) Classification error rate.

4.8 Summary of Chapter

In this chapter, we have focused on improving robustness of feature ranking through multi-criterion combination. We analyze problems in existing combination methods and develop the RPI algorithm for score normalization to solve these problems in multi-criterion combination. Experimental studies on four popular gene expression data sets were conducted and the experimental results show that the robustness of feature ranking by RPI score combination has been improved.

Chapter 5

Robust Feature Subset Selection

Based on Fusion of Multiple

Criteria

5.1 Introduction

Depending on whether to identify target-related features or to find a compact subset of discriminative features, feature selection falls into two categories including feature ranking and feature subset selection. The previous chapter focuses on feature ranking. In this chapter, we will mainly address feature subset selection.

Feature subset selection encounters similar inconsistency problems existing in feature ranking including the inconsistency (instability) problem from perturbations of data and inconsistency between different feature selection criteria (see 4.2 in previous chapter for details). Encouraged by the promising results from combination (fusion) of multiple feature ranking criteria, in this chapter, we propose to improve robustness of feature subset selection through fusion of multiple criteria. Based on this idea, a multi-criterion fusion-based recursive feature elimination (MCF-RFE)

algorithm is developed. As revealed in Kalousis et al. (2007); Saeys et al. (2008), the various feature selection algorithms are sensitive even to a minor variation of training data. We believe this might be due to inaccurate estimation of statistical parameters such as sample mean and standard deviation employed in the feature evaluation criterion. By using multiple criteria, the merit evaluation of features tends to be less sensitive to the inaccurate estimation of the statistical parameters and hence the robustness of the feature selection algorithm is improved. In addition, the proposed new algorithm alleviates the disagreement between different feature selection algorithms by getting their consensus, and hence improves the credibility of the selected features. Experimental studies on five gene expression data sets show that the new MCF-RFE algorithm produces feature subsets with good stability and classification accuracy.

5.2 An Intensive Consideration of Integrating Multiple Criteria

In Chapter 4, we empirically use multi-criterion combination to rank features and focus on the score normalization problem. In this section, we give our intensive consideration for integrating multiple criteria in feature selection.

In pattern classification, it is well acknowledged that combining or integrating multiple classifiers, especially uncorrelated weak ones, could greatly improve the classification performance (Kim and Ko, 2005; Tumer and Ghosh, 1996; Dietterich, 2000). Motivated by the success of multiple classifiers combination, in this study, we propose to improve the robustness of feature selection by integrating multiple

feature selection criteria (algorithms). The proposed feature subset selection algorithm termed multi-criterion fusion-based recursive feature elimination (MCF-RFE) will be detailed in the next section.

The reasons for integrating multiple criteria are manifold. Firstly, different feature selection algorithms produce different feature subsets. As shown in Table 4.1 in Chapter 4 of the top 10 genes selected by five different feature selection algorithms, including Fisher’s ratio, Relief, ADC , AW-SVM and SVM-RFE, the selection results are mostly different. Among the different selection results, which result is superior and should be adopted? Actually, there are no agreed ways to make a decision. Integrating different “opinions” from multiple feature selection criteria to yield a consensus seems to be a reasonable solution. Secondly, a model built upon weak assumptions usually performs more robust than a model built upon stringent assumptions. The existing feature selection criteria are generally built upon certain assumption(s) of data distribution. But the distribution of the learning data is usually rather complicated (e.g., a mixture of many different distributions) and unknown. Even if the distribution is precisely available, it may violate the assumptions to some extent. A criterion that aggregates multiple feature selection criteria can help to weaken the assumptions and consequently improve the robustness. Thirdly, feature subsets produced by different feature selection criteria may exhibit complementary effects because of the non-independence among features, and thus a fusion of these feature subsets may produce a better representation in feature space to describe the data. Fourthly, each feature selection criterion usually has its own specific but restrained ability to search in the feature space and thus may be stuck at a local optimum, while fusion of multiple criteria utilizes and aggregates the search abilities

of each of the criteria to obtain a wider “vision” that may help to get closer to a global optimal solution.

5.3 Multi-Criterion Fusion-Based Recursive Feature Elimination (MCF-RFE) Algorithm

Chapter 4 presents a robust feature evaluation and ranking method based on multi-criterion fusion (combination). If the purpose of feature selection is to improve classification, the feature ranking method may not necessarily be a good choice. It is well acknowledged that a collection of the best features does not necessarily produce the best feature subset because of the existence of correlations between features. In order to obtain a feature subset to produce good classification results, the multi-criterion fusion-based feature evaluation method must be combined with a search strategy to generate feature subsets with reduced correlation levels among features.

Recursive feature elimination (RFE) is a frequently used search strategy in feature selection, see for example (Guyon et al., 2002). The RFE search procedure can be briefly summarized as follows:

RFE procedure:

- (i) Given the full feature set $\mathbf{X}_0$, set $i = 0$.
- (ii) Evaluate the merit of each feature in the feature set $\mathbf{X}_i$.
- (iii) Remove the least important feature(s) from $\mathbf{X}_i$ to obtain feature set $\mathbf{X}_{i+1}$.

- (iv) Set $i = i + 1$ and goes to Step 2 until a stopping criterion is satisfied.

The RFE algorithm generates a group of nested feature subsets, i.e. $\mathbf{X}_0 \supset \mathbf{X}_1 \supset \mathbf{X}_2 \dots$. The original RFE eliminates one feature at each iteration, and could be computational intensive if it is applied to high-dimensional data such as gene expression data. For computational efficiency, a variant of the RFE is to eliminate a portion of the features at each iteration (e.g. 50%). Based on the RFE strategy and SVM, Guyon et al. (2002) proposed a feature subset selection algorithm SVM-RFE (Support Vector Machine Recursive Feature Elimination), where the merit of a feature is evaluated in terms of its corresponding coefficient in the SVM classifier. SVM-RFE produces feature subsets leading to good classification performance and is often used as a benchmark algorithm (Furlanello et al., 2005; Tang et al., 2007; Mundra and Rajapakse, 2010). Despite of its popularity, SVM-RFE lacks robustness. Motivated by strengths of multi-criterion fusion, we formulate a robust feature selection algorithm by combining the multi-criterion fusion-based feature evaluation and the RFE search strategy. We name the new algorithm as MCF-RFE (Multi-Criterion Fusion-Based Recursive Feature Elimination) whose procedure is described in Figure 5.1.

In multiple criteria fusion, both score-based and ranking-based fusion methods are used: a score-based fusion method is first used to generate a feature ranking, which is then added to the m feature rankings produced by individual basis criteria. After that, the $m + 1$ feature rankings are aggregated by a combination to generate the final feature ranking. The fusion procedure is illustrated in Figure 5.2.

Figure 5.1: The procedure of MCF-RFE

Figure 5.2: Score & ranking-based multi-criterion fusion

5.4 An Improved Measure for Evaluating Robustness of Feature Selection Algorithms

The similarity measure of Jaccard Index as defined in Equation 3.10 in Chapter 3 takes into account only the common features between two feature subsets. But there generally exists a great number of features that are highly correlated in real world problems such as gene expression data. In order to give a more general and precise measure of the similarity between two feature subsets, we employ the following similarity index $JC(\in [0, 1])$ that takes into account the correlations (Pearson correlation in the study) between the different features of two feature subsets. Assume $\mathbf{F}_i^{(k)}$ and $\mathbf{F}_0^{(k)}$ denote feature subsets selected using the i -th batch of re-sampled data and the full data respectively, the JC is:

$$JC_i(k) = \frac{|\mathbf{F}_i^{(k)} \cap \mathbf{F}_0^{(k)}| + SC_i}{k} \quad (5.1)$$

where SC_i is the sum of absolute correlation values between the dissimilar features from $\mathbf{F}_i^{(k)}$ and $\mathbf{F}_0^{(k)}$.

The idea behind the index is illustrated in Figure 5.3, where $\mathbf{F}'_i$ and $\mathbf{F}'_0$ are two feature subsets after removing the common features between $\mathbf{F}_i^{(k)}$ and $\mathbf{F}_0^{(k)}$, and each node represents a feature and each edge denotes the correlation between the corresponding features. The final sum of correlations between feature subsets $\mathbf{F}'_i$ and $\mathbf{F}'_0$ is $SC_i = |Corr(\mathbf{f}_{i1}, \mathbf{f}_{01})| + |Corr(\mathbf{f}_{i3}, \mathbf{f}_{02})| + |Corr(\mathbf{f}_{i2}, \mathbf{f}_{03})| = |-0.9| + |0.8| + |0.7| = 2.4$, where $|Corr(\mathbf{f}_{ij}, \mathbf{f}_{0j})|$ is the absolute value of correlation between features $\mathbf{f}_{ij}$ and $\mathbf{f}_{0j}$. The above similarity index resembles the one proposed in Yu et al. (2008) where

SC_i is calculated in an optimal way. For the sake of computational simplicity, SC_i is computed using the greedy search algorithm in our study.

Figure 5.3: Computation of sum of correlation

Assume m batches of data are generated by re-sampling and m feature subsets are selected. The robustness or stability measure of the feature selection algorithm is defined as:

$$\overline{JC}(k) = \frac{\sum_{i=1}^m JC_i(k)}{m} \quad (5.2)$$

For the experimental study in Section 4.2 in Chapter 4, the robustness indices calculated by Equation (5.2) are shown in Figure 5.4, where the number of removed samples is fixed to 10. These results are in line with those in Figure 4.1 in Chapter 4.

The above robustness index provides a sensible evaluation of the stability of a feature selection algorithm. But it should be again emphasized that a robust feature selection does not necessarily guarantee good classification performance because the measure is independent of a classification model. In practice, both stability and classification performance should be considered when evaluating a feature selection algorithm because a stable but classification-ineffective selection result does not

Figure 5.4: Robustness Index when 10 samples are removed in gene selection

make any sense.

5.5 Experimental Studies

5.5.1 Experimental Setup

In this section, extensive experiments were conducted on five gene expression data sets including Colon data, Leukemia data, Prostate data, CNS data and DLBCL data. The number of genes and the number of samples in the five data sets vary from 2000 (Colon data) to 12600 (Prostate data), and 60 (CNS data) to 136 (Prostate data), respectively, which shows a good diversity to cover the commonly used gene expression data. Therefore, it is expected that the performance in these data sets can have a good generalization to other microarray problems.

In addition to the classification error rate, the AUC (area under the ROC curve) was also employed in the evaluation of classification performance after considering the class imbalance in the data sets. The robustness of feature selection algorithms

were evaluated by the robustness measure in Equation (5.2).

For a pattern classification system without a feature selection component, the variance of classification error estimation reflects the sensitivity of a classifier to variations in training and testing samples. For a pattern classification system that includes a feature selection component, although the variance of classification error estimation is the combined effects of both the feature selection algorithm and the pattern classifier, variance still can be used as an indication of the robustness of the feature selection algorithm. This is because a feature selection algorithm sensitive to training data variation usually produces feature subsets leading to large variance in the classification error estimation. For this reason, the standard deviation of the classification error estimation for each feature selection algorithm was investigated in the experiments.

5.5.2 Comparative Study of MCF-RFE with Basis Criteria and SVM-RFE

Figure 5.5 to Figure 5.9 show the classification results including estimated classification error, standard deviation of classification error estimation and AUC values, as well as the robustness of six feature selection methods including Fisher's ratio (FR), ADC, Relief, AW-SVM, SVM-RFE and MCF-RFE, on the five data sets.

We first compare MCF-RFE with the four ranking-based basis criteria: Fisher's ratio (FR), ADC, Relief and AW-SVM. From the estimated classification error on the five data sets (Figures 5.5(a), 5.6(a), 5.7(a), 5.8(a), 5.9(a)), we can observe that MCF-RFE outperforms the basis criteria on all five data sets. For example,

on DLBCL data (Figure 5.9(a)), MCF-RFE produces the least classification error of 2.6% with only 120 features, while the best basis criterion for DLBCL data, Fisher's ratio, produces an error of 3.8% with the same number of features, and achieves its least classification error of 2.8% with 290 features. The AUC results (see Figure 5.5(c), 5.6(c), 5.7(c), 5.8(c), 5.9(c)) are in line with those of the classification error except on CNS data (Figure 5.8(c)) where MCF-RFE is slightly inferior to AW-SVM when the number of selected features is less than 130. The comparison of the standard deviation of error estimation (Figure 5.5(b), 5.6(b), 5.7(b), 5.8(b), 5.9(b)) also proves the effectiveness of MCF-RFE. Take again the results on DLBCL data as an example (see Figure 5.9(b)), the standard deviation is 0.024 for MCF-RFE and 0.031 for Fisher's ratio when 120 features are selected. As for robustness, we find that MCF-RFE does not perform the best but it produces a compromised result of the four basis criteria, and the difference between MCF-RFE and the stablest basis criterion is not much.

In evaluating a feature selection algorithm, both classification and robustness performance should be considered. But classification performance should be the first consideration and robustness should be the secondary because a stable but classification-ineffective selection result does not make any sense. Based on the above considerations, we think the MCF-RFE outperforms the basis criteria because it produces feature subsets with better classification performance and reasonably good robustness.

When it comes to the performance comparison between MCF-RFE and SVM-RFE, we observe that MCF-RFE achieves substantial improvements over SVM-RFE in the robustness performance on all five data sets (refer to Figure 5.5(d), 5.6(d),

5.7(d), 5.8(d), 5.9(d)). MCF-RFE also produces better classification performance than SVM-RFE except the AUC on CNS data. Here we still take the results on DLBCL data (see Figure 5.9) as an example. With a subset of 120 features, the respective values of estimated classification error, standard deviation of error estimation, AUC and robustness of MCF-RFE vs. SVM-RFE are 2.6% vs. 3.6%, 0.024 vs. 0.030, 0.996 vs. 0.992 and 80.7% vs. 67.2%. The good results of MCF-RFE again prove the strengths of multiple criteria fusion.

5.5.3 Comparative Study of MCF-RFE with Bagging-Based Ensemble Technique (BBET)

This part conducts a comparative study between MCF-RFE and the bagging-based ensemble feature selection technique (BBET) proposed by Saeys et al. (2008). BBET is based on instance perturbation and can be applied to any of ranking-based feature selection algorithms. In this study, the procedure of BBET is as follows: first, a number of bags (i.e. subsamples) are generated from the training samples using re-sampling with replacement technique, and a feature ranking algorithm is performed on each of the bags to produce separate feature rankings; then all the feature rankings are combined to form a final feature ranking using an aggregation method. In the experiment, BBET was applied to each of the four basis criteria with 40 bags (which is the same as in Saeys et al. (2008)) and *Borda count* was used as the linear aggregation method. Figure 5.10 to Figure 5.14 show the results.

We first compare the robustness performances because BBET is originally designed to improve robustness of feature selection algorithms. From the robustness

Figure 5.5: Performance comparisons on Colon data: (a) Classification error. (b) Standard deviation of error estimation. (c) AUC and (d) Robustness.

Figure 5.6: Performance comparisons on Leukemia data: (a) Classification error. (b) Standard deviation of error estimation. (c) AUC and (d) Robustness.

Figure 5.7: Performance comparisons on Prostate data: (a) Classification error. (b) Standard deviation of error estimation. (c) AUC and (d) Robustness.

Figure 5.8: Performance comparisons on CNS data: (a) Classification error. (b) Standard deviation of error estimation. (c) AUC and (d) Robustness.

Figure 5.9: Performance comparisons on DLBCL data: (a) Classification error. (b) Standard deviation of error estimation. (c) AUC and (d) Robustness.

results, it can be observed that MCF-RFE performs better than or comparable to the best basis criterion with BBET. For the classification performance including estimated classification error, standard deviation of error estimation and AUC, MCF-RFE performs better with exceptions of AUC and standard deviation on CNS data.

After comparing carefully the results of the four basis criteria with BBET with those without BBET, we find that BBET may not always be beneficial, which is different from the conclusion in Saeys et al. (2008) that BBET generally provides more robust results. For example, in most of the cases, *AW-SVM* benefits from BBET while *Fisher's ratio* does not. We think one possible reason is the small sample size of gene expression data. In addition, due to the resampling with replacement used in .632 bootstrap in our experiments, only about $63.2\% \times 63.2\% = 39.9\%$ of the original samples were retained for construction of each feature selector in the ensemble, while in Saeys et al. (2008) $90\% \times 63.2\% = 56.9\%$ of the original samples were used to construct each feature selector.

5.6 Summary of Chapter

In this chapter, we have analyzed and discussed multi-criterion fusion for feature selection on high-dimensional and small-sized data. Motivated by the strengths of fusion of multiple criteria and the recursive feature elimination (RFE) search strategy, we have proposed a feature subset selection algorithm—MCF-RFE. Extensive experimental studies of MCF-RFE with Fisher's ratio, Relief, ADC, AW-SVM and the commonly used benchmark algorithm SVM-RFE based on four performance in-

Figure 5.10: MCF-RFE and BBET comparisons on Colon data: (a) Classification error. (b) Standard deviation of error estimation. (c) AUC and (d) Robustness.

Figure 5.11: MCF-RFE and BBET comparisons on Leukemia data: (a) Classification error. (b) Standard deviation of error estimation. (c) AUC and (d) Robustness.

Figure 5.12: MCF-RFE and BBET comparisons on Prostate data: (a) Classification error. (b) Standard deviation of error estimation. (c) AUC and (d) Robustness.

Figure 5.13: MCF-RFE and BBET comparisons on CNS data: (a) Classification error. (b) Standard deviation of error estimation. (c) AUC and (d) Robustness.

Figure 5.14: MCF-RFE and BBET comparisons on DLBCL data: (a) Classification error. (b) Standard deviation of error estimation. (c) AUC and (d) Robustness.

dices including classification error, standard deviation of error estimation, AUC and robustness have been conducted, and the results show that MCF-RFE outperforms in classification performance with reasonably good robustness. An comparative study between our proposed algorithm and the bagging-based ensemble technique (BBET) has also proved the strengths of multi-criterion fusion.

Chapter 6

Regularizing Linear Discriminant

Analysis with Emphasis on

Minority Class for Feature Subset

Selection

6.1 Introduction

Linear discriminant analysis (LDA) is mostly used for pattern classification, but it can also be used as an effective measure to evaluate the separative ability of a feature subset in feature subset selection (Ye et al., 2006; Wu et al., 2009; Huetta et al., 2010; Paliwala and Sharma, 2010). Instead of using the classification error of LDA classifier as evaluation criterion, LDA-based feature selection evaluates features based on the ratio of between-class difference to within-class scatter of projected samples on the dimension that maximizes the ratio. LDA projection involves inverse operation of the scatter matrix. For the high dimensional and small sized (HDSS) data, for example microarray data, however, the dimensionality is far

greater than the number of samples and the scatter matrix is therefore singular. To deal with this problem, regularization techniques can be used. One practice of regularization is to add a small positive constant to the diagonal of scatter matrix to ensure its positive definiteness. Another practice of regularization is to shrink individual scatter matrices towards pooled scatter matrix or/and shrink the pooled scatter matrix towards its diagonal (see for example Hastie et al. (2009)).

Besides the singularity problem, we found that LDA-based feature selection for the HDSS data encounters three more problems. The first problem is overfitting. Due to very small sample size, LDA tends to overfit training data, and the feature subsets selected exhibit poor generalization performance on testing data. The overfitting problem could be alleviated by the above regularization practice, but our experiment shows that the improvement is below expectation. The second problem is overwhelming problem. For many practical problems such as gene expression data, classes are generally imbalanced (i.e. the number of samples in one class is much greater than the number of samples in the other class). Majority class is prone to overwhelm the minority class by dominating the feature selection process, which may further aggravate the already overfitting problem in minority class. The third problem is the prohibitive computational overhead resulted from inverse operations of a great number of high-dimensional scatter matrices.

To address the above problems, we propose a new form of regularization to LDA. Instead of giving more emphasis to majority class as done in the conventional forms of regularization such as shrinking individual scatter matrices towards the pooled scatter matrix, our regularization will emphasize more on the minority class. The rationale behind our regularization is that in the situation of small sample size and

class imbalance, the minority class is prone to be overlooked during evaluating a feature subset because of the rather small number of samples in the minority class, which in turn would result in overfitting problem in minority class. Therefore, the influence of the minority class should be enhanced when evaluating a feature subset in feature selection. Our experimental studies show that the proposed regularization produces feature subsets leading to improved classification performance as well as robustness performance. Furthermore, in order to avoid the prohibitively high computational complexity in the direct implementation of LDA-based feature selection, an incremental implementation is introduced and employed in the study.

6.2 Regularized LDA-Based Techniques for Feature Subset Evaluation

6.2.1 From Fisher's Ratio to LDA for Feature Subset Evaluation

Fisher's ratio is a univariate feature evaluation criterion while LDA can be viewed as its multivariate form which considers correlations between features. The details about Fisher's ratio and LDA for feature evaluation are as follows.

Given a binary-class problem with a total of n samples, each with p features as described in Section 2.6.3, Fisher's ratio individually evaluates each feature by the FR value (see Equation (4.1)). Note that $FR \in [0, +\infty)$, then for a feature subset containing t features:

$$\mathbf{F}^{(t)} = \{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_t\}$$

Fisher's ratio will evaluate the overall discriminating power of $\mathbf{F}^{(t)}$ by simply adding the FR values of all features:

$$FR(\mathbf{F}^{(t)}) = FR(\mathbf{X}_1) + FR(\mathbf{X}_2) + \cdots + FR(\mathbf{X}_t) \quad (6.1)$$

Selecting a feature subset of t features through Equation (6.1) is equivalent to decreasingly ranking all candidate features by their individual importance first and then retaining the top t features to form the feature subset, which is actually how Fisher's ratio is used in practice. Because of its individual characteristic, Fisher's ratio is extremely fast to implement for feature selection. Together with its generally good performance in many practical problems, Fisher's ratio has been widely accepted and employed, especially for problems with very high dimensionality such as gene microarray problems. But the drawback of Fisher's ratio is also obvious. The individual evaluation of features makes Fisher's ratio relinquish any correlation information between features when evaluating a feature subset. Therefore, a feature subset selected by Fisher's ratio may contain individually important but mutually highly correlated features. Those individually weak but combinatorially important features will be hardly selected, which will greatly limit the performance of Fisher's ratio in feature subset selection. To alleviate or overcome this problem and to yield a good feature subset for effective pattern classification, correlation information between features must be considered during evaluating a feature subset.

Linear discriminant analysis (LDA) makes use of the correlation information between features when evaluating the overall discriminating power of a group of features. Given a t -feature subset $\mathbf{F}^{(t)}$, LDA first projects all samples from t -

dimensional space onto a new dimension, and then evaluates goodness of the feature subset $\mathbf{F}^{(t)}$ by the Fisher's ratio calculated from the projected samples on the new dimension. Suppose $\tilde{\mathbf{x}}(i), (i = 1, 2, \dots, n)$ is the i -th sample in the t -dimensional space, the projected samples are then given by:

$$z(i) = \tilde{\mathbf{x}}(i)\mathbf{w}^T$$

The Fisher's ratio of the projected samples on the new dimension will be:

$$FR(\mathbf{F}^{(t)}) = \frac{\mathbf{w}\mathbf{S}_B\mathbf{w}^T}{\mathbf{w}\mathbf{S}_W\mathbf{w}^T} \quad (6.2)$$

where $\mathbf{w}$ is the $1 \times t$ projection vector that maximizes Equation (6.2) and is given by (Hastie et al., 2009):

$$\mathbf{w}^T = \mathbf{S}_W^{-1}(\mathbf{m}_{(1)} - \mathbf{m}_{(2)})^T \quad (6.3)$$

$\mathbf{S}_B$ and $\mathbf{S}_W$ are the between-class and within-class scatter matrix defined as follows:

$$\mathbf{S}_B = (\mathbf{m}_{(1)} - \mathbf{m}_{(2)})^T(\mathbf{m}_{(1)} - \mathbf{m}_{(2)}) \quad (6.4)$$

$$\mathbf{S}_W = \mathbf{S}_{W(1)} + \mathbf{S}_{W(2)} \quad (6.5)$$

$$\mathbf{S}_{W(c)} = \frac{1}{n_c - 1} \sum_{\mathbf{x} \in \text{class } c} (\mathbf{x} - \mathbf{m}_{(c)})^T(\mathbf{x} - \mathbf{m}_{(c)}) \quad (6.6)$$

where $\mathbf{m}_{(c)}$ and $\mathbf{S}_{W(c)}$ are the $1 \times t$ mean vector and $t \times t$ within-class covariance matrix of class c , and n_c is the number of samples in class c , for $c = 1, 2$.

After substituting Equations (6.3) to (6.6) into (6.2), LDA will evaluate a feature

subset $\mathbf{F}^{(t)}$ as:

$$FR(\mathbf{F}^{(t)}) = (\mathbf{m}_{(1)} - \mathbf{m}_{(2)})\mathbf{S}_W^{-1}(\mathbf{m}_{(1)} - \mathbf{m}_{(2)})^T \quad (6.7)$$

For a given feature subset, LDA-based evaluation seems to be a straightforward process: find the optimal projection $\mathbf{w}$, project samples, and evaluate Fisher's ratio of projected samples on the new dimension. However, it is noted that the within-class scatter matrix $\mathbf{S}_W$ might be singular, particularly for gene microarray data which are HDSS. In addition, it is noted that LDA-based feature selection for HDSS data encounters overfitting problem which is often neglected. The consequence of overfitting in feature selection is that the feature subset selected produces perfect classification for training data, but very bad performance on testing data.

6.2.2 Conventional Forms of Regularization to LDA

To deal with the singularity and/or overfitting problems, regularization techniques can be used. The simplest form of regularization is the addition of a small positive constant to the diagonal of the scatter matrix as follows:

$$\mathbf{S}_W \leftarrow \mathbf{S}_W + \lambda \mathbf{I} \quad (6.8)$$

where λ is a small positive constant, and $\mathbf{I}$ is an identity matrix with the same size of $\mathbf{S}_W$. The small positive constant λ should ensure positive definiteness of the scatter matrix $\mathbf{S}_W$. Without specification, a very small number is added to the diagonal of the scatter matrix for all the other regularization techniques we will mention below.

Another more sophisticated form of regularization is the so called shrinkage tech-

nique, which shrinks individual within-class scatter matrices towards the pooled scatter matrix $\mathbf{S}_P$ (see for example Hastie et al. (2009)):

$$\mathbf{S}_{W(i)}(\alpha) = \alpha \mathbf{S}_{W(i)} + (1 - \alpha) \mathbf{S}_P \quad (6.9)$$

where $\alpha \in [0, 1]$ and $\mathbf{S}_{W(i)}$ is the within-class scatter matrix of class i . The pooled scatter matrix $\mathbf{S}_P$ is in the following form and it is sometimes used to replace the within-class matrix $\mathbf{S}_W$ in LDA (Hastie et al., 2009).

$$\mathbf{S}_P = \frac{(n_1 - 1)\mathbf{S}_{W(1)} + (n_2 - 1)\mathbf{S}_{W(2)}}{n_1 + n_2 - 2} \quad (6.10)$$

A diagonal element in the scatter matrix will partly reflect the importance of a single feature without considering its relationship to other features (refer to Equation (4.1)), while a non-diagonal element mainly reflects the relationship between a specific pair of different features. Different from the above forms of regularization which treat the diagonal and non-diagonal elements in the scatter matrix equally, another form of shrinkage is to shrink the scatter matrix towards its diagonal so as to give more emphasis to the diagonal elements. For example, shrink the within-class scatter matrix towards its diagonal (Thomaz et al., 2001):

$$\mathbf{S}_W(\rho) = \rho \mathbf{S}_W + (1 - \rho) \text{diag}(\mathbf{S}_W) \quad (6.11)$$

and/or shrink the pooled scatter matrix towards its diagonal after replacing the within-class scatter matrix $\mathbf{S}_W$ with the pooled scatter matrix $\mathbf{S}_P$ in Equation (6.7)

(Hastie et al., 2009; Thomaz et al., 2001):

$$\mathbf{S}_P(\rho) = \rho \mathbf{S}_P + (1 - \rho) \text{diag}(\mathbf{S}_P) \quad (6.12)$$

where $\rho \in [0, 1]$.

6.2.3 A New Regularization Giving More Emphasis to Minority Class

The conventional regularization techniques introduced in the above section can be united into two types of emphasis techniques: 1) class emphasis (see Equation (6.8) to (6.10)) and 2) diagonal emphasis (see Equation (6.11) and (6.12)). The second type is to be done when the scatter matrix is given. For example, Equation (6.11) is equivalent to multiplying a shrinkage factor ρ ($\in [0, 1]$) for all non-diagonal elements while keeping the diagonal elements unchanged, which also means the diagonal elements are emphasized. For the first type of emphasis, the details are as follows.

If we let $\tau = \frac{n_1-1}{n_1+n_2-2}$ ($\tau \in (0, 1)$), then $\mathbf{S}_P$ in Equation (6.10) becomes:

$$\mathbf{S}_P = \tau \mathbf{S}_{W(1)} + (1 - \tau) \mathbf{S}_{W(2)} \quad (6.13)$$

After substituting Equation (6.13) to (6.9) and then to (6.5), it can be obtained:

$$\mathbf{S}_W(\alpha) = \eta \mathbf{S}_{W(1)} + (1 - \eta) \mathbf{S}_{W(2)} \quad (6.14)$$

where $\eta = \frac{1}{2}\alpha + (1 - \alpha)\tau$ and $\eta \in (0, 1)$.

Therefore, the difference between the regularizations in (6.8) to (6.10) is only reflected on their different emphasis on the class scatter matrix $\mathbf{S}_{W(c)}$, for $c = 1, 2$. Without losing generalization, we assume $n_1 \geq n_2$ between classes, then the ranges of the parameters will become $\tau \in [0.5, 1)$ and $\eta \in [0.5, 1)$ (when $n_1 = n_2$, $\eta = \tau = 0.5$), which means in the regularizations (6.8) to (6.10), more or equal emphases will be given to the majority class.

Emphasizing the majority class in LDA is reasonable and is guaranteed by Bayesian theory for conventional data where sufficient samples are available to support accurate estimation of the data distribution in each class. But it may not necessarily be a good choice for data with very small number of samples, high dimensionality and usually imbalance between classes, where it is impossible to accurately estimate the data distributions. Inverse to the way treating the conventional data, our viewpoint is that more emphasis should be given to the minority class rather than to the majority class. The rationale behind this is that in situation of small sample size and class imbalance, feature selection may tend to fit better the majority class and overlook the effects of the minority class, which makes the majority class overwhelm the minority class, and consequently the lack of generalization ability may behave more serious on the minority class. Giving more emphasis to the minority class could help to draw more attention to the data in the minority class and hence to enhance their influence during evaluating a feature subset in feature selection, which may probably help to alleviate the overfitting problem in minority class.

In this study, we propose a new form of regularization which emphasizes the minority class first and then emphasizes on the diagonal. The details are shown as

below:

$$\mathbf{S}_W(\gamma) = \gamma \mathbf{S}_{W(1)} + (1 - \gamma) \mathbf{S}_{W(2)} \quad (6.15)$$

$$\mathbf{S}_W(\rho, \gamma) = \rho \mathbf{S}_W(\gamma) + (1 - \rho) \text{diag}(\mathbf{S}_W(\gamma)) \quad (6.16)$$

where $\gamma \in (0, 0.5)$ when $n_1 > n_2$ and $\gamma \in (0.5, 1)$ when $n_1 < n_2$.

We think a diagonal emphasis is very important and quite necessary for LDA-based feature selection. The reasons are as follows. As aforementioned, the diagonal and non-diagonal elements in the scatter matrix embody the information of features themselves and their pair-wised relationships, respectively. When evaluating a feature based on a large batch of selected features, its importance could be greatly concealed by its relationships to all selected features. Consequently, due to the small portion of diagonal elements in a scatter matrix compared to the number of non-diagonal elements, the importance of a feature subset originating from themselves may be concealed by their inter-relationships. Therefore, the diagonal of a scatter matrix should be emphasized to release the features from their inter-relationships when evaluating their importance.

6.3 Incremental Implementation of LDA-Based Feature Subset Selection

Any feature subset evaluation criterion can be used for feature subset selection when combined with a feature subset generation strategy. There are two widespread strategies in general: sequential forward searching (SFS) and sequential backward searching(SBS). Due to relatively lower searching complexity, SFS is more popular

in practice, especially for data with very high dimensionality. After considering the extensive complexity of LDA-based feature subset evaluation as well as the very high dimensionality in HDSS data, SFS is employed in this study.

The procedure of the LDA-based feature subset selection utilizing the sequential forward searching is summarized below:

- (i) At $t = 0$, initialize the full feature set $\mathbf{X} = \{X_1, X_2, \dots, X_p\}$ and the selected feature subset $\mathbf{F}^{(t)} = \{ \}$.
- (ii) Calculate the Fisher's ratio (note that Equation (6.7) is equivalent to Equation (4.1) when evaluating a single feature) for each feature in $\mathbf{X}$, select the most important feature, say X'_1 , let $\mathbf{F}^{(t)} = \mathbf{F}^{(t)} \cup \{X'_1\}$ and $\mathbf{X} = \mathbf{X} \setminus \{X'_1\}$. Set $t = 1$.
- (iii) For each candidate feature X_j in $\mathbf{X}$, generate a temporary feature subset $\mathbf{F}_{tmp}^{(t+1)} = \mathbf{F}^{(t)} \cup \{X_j\}$, and evaluate feature X_j using Equation (6.7) on $\mathbf{F}_{tmp}^{(t+1)}$.
- (iv) Select the most important feature in $\mathbf{X}$, say X'_t , let $\mathbf{F}^{(t)} = \mathbf{F}^{(t)} \cup \{X'_t\}$ and $\mathbf{X} = \mathbf{X} \setminus \{X'_t\}$. Set $t = t + 1$.
- (v) Repeat (iii) to (iv) until a predefined number of features are selected in $\mathbf{F}^{(t)}$.

As it can be seen in (iii) in the above procedure, an inverse of the within-class scatter matrix (i.e. $\mathbf{S}_W^{-1}$) is needed to be calculated when evaluating every candidate feature from the remaining feature set $\mathbf{X}$, which will incur very high computational burdens. Suppose a feature subset containing d features is required to be selected from totally p features, then the computational complexity merely involved in matrix inverse operations in the above procedure will reach to $O(p \cdot d^4)$. Take gene selection

for example, p and d are generally on the order of 10^4 and 10^2 , respectively, the computational complexity for calculating $\mathbf{S}_W^{-1}$'s will then be $O(10^{12})$. This would make the above LDA-based feature subset selection algorithm rather time-consuming and even computationally prohibitive for practical problems.

To reduce the computational overhead, Mao and Tang (2011) propose to compress the information in all selected features into one dimension and hence only a 2×2 matrix inverse is required for evaluating each candidate feature. Their implementation is very fast but has information loss due to information compression. In this study, an incremental implementation with recursively computing $\mathbf{S}_W^{-1}$ is employed. The details are as follows.

Assume a feature subset $\mathbf{F}^{(t-1)}$ containing $t-1$ features has already been selected and the corresponding within-class scatter matrix is $\mathbf{S}_{t-1}$. Other terms which have already been obtained include:

$$\begin{cases} \mathbf{S}_{t-1}^{-1}, \text{ and } (\mathbf{S}_{t-1}^{-1})^T = \mathbf{S}_{t-1}^{-1} \\ \Delta \mathbf{m}_{t-1} = [\mathbf{m}_{t-1,(1)} - \mathbf{m}_{t-1,(2)}] \\ FR(\mathbf{F}^{(t-1)}) = \Delta \mathbf{m}_{t-1} \mathbf{S}_{t-1}^{-1} \Delta \mathbf{m}_{t-1}^T \end{cases}$$

Then when a feature X_j is added into $\mathbf{F}^{(t-1)}$ to form $\mathbf{F}^{(t)}$, there will be:

$$\Delta \mathbf{m}_t = [\Delta \mathbf{m}_{t-1}, \Delta m_j] \quad (6.17)$$

where $\Delta m_j = m_{j(1)} - m_{j(2)}$ is the between-class difference of sample mean of feature X_j .

And the within-class scatter matrix $\mathbf{S}_t$ will be in the form:

$$\mathbf{S}_t = \begin{bmatrix} \mathbf{S}_{t-1} & \mathbf{b} \\ \mathbf{b}^T & c \end{bmatrix} \quad (6.18)$$

where $\mathbf{b}$ is a $(t-1) \times 1$ vector and c is scalar value.

Referring to the inverse of a block-wise matrix, it then can be obtained (Harville, 2008):

$$\mathbf{S}_t^{-1} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}$$

where

$$\begin{cases} \mathbf{A}_{11} = \mathbf{S}_{t-1}^{-1} + \frac{\mathbf{S}_{t-1}^{-1} \mathbf{b} \mathbf{b}^T \mathbf{S}_{t-1}^{-1}}{c - \mathbf{b}^T \mathbf{S}_{t-1}^{-1} \mathbf{b}} \\ \mathbf{A}_{12} = -\frac{\mathbf{S}_{t-1}^{-1} \mathbf{b}}{c - \mathbf{b}^T \mathbf{S}_{t-1}^{-1} \mathbf{b}} \\ \mathbf{A}_{21} = \mathbf{A}_{12}^T \\ \mathbf{A}_{22} = \frac{1}{c - \mathbf{b}^T \mathbf{S}_{t-1}^{-1} \mathbf{b}} \end{cases}$$

Let

$$\begin{cases} \mathbf{e} = \mathbf{S}_{t-1}^{-1} \mathbf{b} \\ a = c - \mathbf{b}^T \mathbf{S}_{t-1}^{-1} \mathbf{b} \end{cases} \quad (6.19)$$

then we have:

$$\mathbf{S}_t^{-1} = \begin{bmatrix} \mathbf{S}_{t-1}^{-1} + \frac{1}{a} \mathbf{e} \mathbf{e}^T & -\frac{1}{a} \mathbf{e} \\ -\frac{1}{a} \mathbf{e}^T & \frac{1}{a} \end{bmatrix} \quad (6.20)$$

After substituting Equation (6.17) and (6.20) to (6.7), the evaluation value of feature subset $\mathbf{F}^{(t)}$ becomes:

$$FR(\mathbf{F}^{(t)}) = FR(\mathbf{F}^{(t-1)}) + \frac{1}{a} (\Delta \mathbf{m}_t \mathbf{e} - \Delta m_j)^2 \quad (6.21)$$

Using the above equations, the LDA-based feature subset selection can be incrementally implemented as described in Algorithm 1. Note that $\mathbf{X}(j)$ is the j -th element in set $\mathbf{X}$. By the incremental implementation, the computational complexity involved in the scatter matrix inverse operations will be reduced to $O(p \cdot d^3)$ when selecting d features from a full set of p features. Take the Leukemia data (see Section 2.6.2 for details) for example, $p = 7129$ and $d = 300$, our experiments show that the averaged computational time is 64 seconds using Algorithm 1 but the time will increase to 5548 seconds when using the direct implementation procedure as described at the beginning of this section. When multiple repeats of experiments are needed, e.g. 300 repeats in our experimental studies, the computational time will be hardly endured when using the direct implementation procedure of the LDA-based feature subset selection.

Algorithm 1: Incremental Implementation of LDA-Based Feature Subset Selection

Input : n pairwised samples $\langle \mathbf{x}(1), y(1) \rangle, \langle \mathbf{x}(2), y(2) \rangle, \dots, \langle \mathbf{x}(n), y(n) \rangle$

Output: A feature subset $\mathbf{F}^{(d)}$ with d features

begin

 Evaluate all features in $\mathbf{X}$ using Fisher's ratio in Equation (4.1), select the most important feature $\mathbf{X}'_1$, let $\mathbf{F}^{(1)} = \{\mathbf{X}'_1\}$, $\mathbf{X} = \mathbf{X} \setminus \{\mathbf{X}'_1\}$.

 Construct $\Delta \mathbf{m}_1$ and $\mathbf{S}_1$, let $\mathbf{S}_1^{-1} = \frac{1}{\mathbf{S}_1}$, $FR(\mathbf{F}^{(1)}) = FR(\mathbf{X}'_1)$.

for $t \leftarrow 2$ **to** d **do**

for $j \leftarrow 1$ **to** $|\mathbf{X}|$ **do**

$\mathbf{F}_{tmp}^{(t)} = \mathbf{F}^{(t-1)} \cup \mathbf{X}(j)$;

 Construct $\Delta \mathbf{m}_t$ and $\mathbf{S}_t$ according to features in $\mathbf{F}_{tmp}^{(t)}$, obtain $\mathbf{b}$, c .

 Calculate $Q(j) = FR(\mathbf{F}_{tmp}^{(t)})$ using Equation (6.21).

end

$\mathbf{X}'_t = \arg_{\mathbf{X}(j)} \max Q(j)$;

$\mathbf{F}^{(t)} = \mathbf{F}^{(t-1)} \cup \{\mathbf{X}'_t\}$; $\mathbf{X} = \mathbf{X} \setminus \{\mathbf{X}'_t\}$;

$FR(\mathbf{F}^{(t)}) = \max Q(j)$;

 Update $\Delta \mathbf{m}_t$, $\mathbf{S}_t$ and $\mathbf{S}_t^{-1}$ using Equation (6.17) to (6.20);

end

end

6.4 Experimental Studies

In the experimental studies, extensive experiments were carried out on five microarray gene expression data sets: Colon data, Leukemia data, DLBCL data, CNS data and Prostate data. After referring to Section 2.6.2 in Chapter 2 for the details of each data set, it can be found that all the data sets are of high dimensionality, small sample size and class imbalance. In experiments, the estimated classification error rate and AUC were employed as the classification performance measures, while the robustness was evaluated using Equation (3.12) in Chapter 3.

6.4.1 Investigation of the Influence of Parameter γ in Class Emphasis

In all the forms of regularization we used in this study, there are parameters for class emphasis and/or diagonal emphasis in addition to the small positive constant λ to the diagonal of the scatter matrix. A central issue when using those forms of regularization is how to choose suitable values for the regularization parameters. One general way to determine the parameters on traditional large-sized data is to select the best value from a group of given values by conducting cross-validation on the training data. But for HDSS data, this is not suitable and even impractical because of the very small sample size. Therefore, we will not aim to obtain the best parameters for each training data but try to obtain reasonable values of the parameters by conducting experiments on the full data set.

Among the parameters in the regularizations, the λ can be set to any small positive constant, say 0.01, as long as it ensures the positive definiteness of the scatter

matrix. Actually, there are no significant differences observed in the additional experimental results when different small positive constants are set to λ . As a result, λ is fixed to 0.01 in all experimental studies in this chapter.

In our proposed form of regularization as shown in Equation (6.16), there are two parameters γ and ρ which are for class emphasis and diagonal emphasis, respectively. In order to provide a complete picture of the influence of each parameter, we will determine the two parameters separately. The following experiments in this section are to study the influence of the parameter γ in class emphasis.

In the experiments, the bootstrap testing with 100 repeats was used, and γ was set to 7 different values of 0.0, 0.1, 0.3, 0.5, 0.7, 0.9, 1.0 in its range (refer to Equation (6.15)), where $\gamma = 0.0$ and $\gamma = 1.0$ denote complete emphasis on class 2 and class 1, respectively. Figures 6.1 to 6.5 show the results including the classification and robustness performance on the corresponding five data sets.

Figure 6.1: Classification and robustness performance under different values of γ for class emphasis on DLBCL data: (a) Classification and (b) Robustness.

We first compare the results of complete emphasis on one class (see $\gamma = 0.0$

Figure 6.2: Classification and robustness performance under different values of γ for class emphasis on Prostate data: (a) Classification and (b) Robustness.

Figure 6.3: Classification and robustness performance under different values of γ for class emphasis on Colon data: (a) Classification and (b) Robustness.

and $\gamma = 1.0$). It can be observed that on DLBCL data (see Figure 6.1), both the classification and robustness performance from $\gamma = 0.0$ (complete emphasis on class 2) are much superior to those from $\gamma = 1.0$ (complete emphasis on class 1). The situation on Prostate data (see Figure 6.2) changes contrarily, where both the clas-

Figure 6.4: Classification and robustness performance under different values of γ for class emphasis on Leukemia data: (a) Classification and (b) Robustness.

Figure 6.5: Classification and robustness performance under different values of γ for class emphasis on CNS data: (a) Classification and (b) Robustness.

sification and robustness performance from $\gamma = 1.0$ (complete emphasis on class 1) are much superior to those from $\gamma = 0.0$ (complete emphasis on class 2). For the performance on the other three data sets (see Figures 6.3, 6.4, 6.5), better classification and robustness performance are obtained from $\gamma = 0.0$ (complete emphasis

on class 2). If we connect those performance observations on the five data sets with the majority/minority class of each data set in Table 2.2 (i.e. the minority class of Prostate data is class 1 and that of the rest data sets is class 2), it then can be concluded that a complete emphasis on minority class will be superior to a complete emphasis on majority class. This drops us a strong hint in class emphasis that more emphasis should be given to minority class.

Let's then focus on the results with $\gamma \in (0, 1)$. It can be observed that with increased emphasis to the minority class (e.g. γ decreases from 0.9 to 0.1 in Figure 6.1 or γ increases from 0.1 to 0.9 in Figure 6.2), there will be slight improvements in classification performance when selecting the first several genes. The best results including the classification and robustness performance are achieved when $\gamma = 0.1$ for DLBCL data in Figure 6.1 and $\gamma = 0.9$ for Prostate data in Figure 6.2, where major emphases are given to the minority class. For the results on the other three data sets, similar phenomenon has been observed.

After considering what we have observed from the experimental results in this section, the parameter γ in our proposed regularization was set to:

$$\begin{cases} \gamma = 0.1, & \text{when } n_1 > n_2 \\ \gamma = 0.9, & \text{when } n_1 < n_2 \end{cases}$$

where n_1 and n_2 are the number of samples in class 1 and class 2. In the application where $n_1 = n_2$, $\gamma = 0.5$ could be a good choice.

The experimental results in this section support our previous analysis to class emphasis: by putting greater emphasis on the scatter matrix of the minority class,

the overfitting problem could be alleviated. This result also shows another merit of the new regularization technique, that is the ease of setting the parameter γ .

There may be one doubt to the experimental results with $\gamma \in (0, 1)$: the differences in the results for different values of $\gamma \in (0, 1)$ are unobvious and only observable when selecting the first several dozen of features. The possible reason we think is that in the regularization the diagonal and non-diagonal elements of the scatter matrix are treated equally, which causes the concealment effects as analyzed in Section 6.2.3 and prevents from selecting really useful features. This will be confirmed by the experiments in the following two sections.

6.4.2 Determination of the Parameter ρ for Diagonal Emphasis

The experiments in this section are to determine the parameter ρ for diagonal emphasis. Diagonal emphasis is required in our proposed regularization as well as in the conventional regularizations (see Equation (6.11) and (6.12)). As analyzed in Section 6.2.3, the difference between those regularizations with diagonal emphasis is the different emphasis between classes. A preliminary experiment shows that the parameter ρ is not sensitive to the changes of parameter γ for class emphasis and hence there is no need to conduct experiments for each value of γ to find the corresponding optimal value of ρ . In the following experiments, we will fix $\gamma = 0.5$ (i.e. equal emphasis between classes) and then determine the commonly acceptable ρ based on the results on the full data sets.

To obtain reliable estimation in the experiments, the bootstrap with 100 repeats

was used. The parameter ρ was chosen to be 0.0, 0.05, 0.1, 0.2, 0.4, 0.6, 0.9, 1.0 from its range (i.e. $\rho \in [0, 1]$). The diagonal emphasis is actually achieved by multiplying all non-diagonal elements in the scatter matrix with the shrinkage factor ρ . If each non-diagonal element is viewed the correlation (or redundancy) between a pair of features, then the parameter ρ will control the redundancy deduction in feature selection. The less the ρ is, the less redundancy is reduced; with $\rho = 0$, LDA-based feature selection will degrade to feature ranking which considers absolutely no correlations between features.

Figure 6.6: Classification and robustness performance under different values of ρ for diagonal emphasis on DLBCL data: (a) Classification and (b) Robustness.

Figures 6.6 to 6.10 exhibit the experimental results including the classification and robustness performance on the corresponding five data sets. A reasonable relationship between redundancy and robustness should be that the increased redundancy deduction leads to deterioration in robustness performance. This is also what we can observe from the results. Take Figure 6.6(b) for example, the robustness deteriorates with the increase of ρ from $\rho = 0$ to $\rho = 1$. When solely considering

Figure 6.7: Classification and robustness performance under different values of ρ for diagonal emphasis on Prostate data: (a) Classification and (b) Robustness.

Figure 6.8: Classification and robustness performance under different values of ρ for diagonal emphasis on Colon data: (a) Classification and (b) Robustness.

the robustness performance, a choice of $\rho = 0$ will be the best. But when evaluating the performance of feature selection in practice, robustness performance should be a secondary consideration and classification performance the first. As to the classification performance in the experimental results (see Figure 6.6(a) and Figure 6.7(a))

Figure 6.9: Classification and robustness performance under different values of ρ for diagonal emphasis on Leukemia data: (a) Classification and (b) Robustness.

Figure 6.10: Classification and robustness performance under different values of ρ for diagonal emphasis on CNS data: (a) Classification and (b) Robustness.

for example), neither $\rho = 0$ nor $\rho = 1$ is the best choice, which means a proper level of redundancy reduction (i.e. $\rho \in (0, 1)$) will be required. After comparing the results on all five data sets, it is observed that the value of ρ should be in the range of $(0, 0.2)$. In the comparative study in next section, we set $\rho = 0.1$.

Name	Diagonal Emphasis?	Emphasized Class	Equation
$S_{Wi}ToS_P$	No	Majority Class	(6.9)or(6.14)
$S_W + \lambda * I$	No	Equal Emphasis	(6.8)
$Emph(S_{Wi})$	No	Minority Class	(6.15)
$Diag(S_P)$	Yes	Majority Class	(6.12)
$Diag(S_W)$	Yes	Equal Emphasis	(6.11)
$Emph+Diag$	Yes	Minority Class	(6.16)

Table 6.1: The forms of regularization

6.4.3 Comparative Studies of Different Forms of Regularization

This section carries out experiments for the comparative study among different forms of regularization. As listed in Table 6.1, there are six different forms of regularization and these regularizations can be divided into two groups: one is without diagonal emphasis (the first three in the table) and the other is with diagonal emphasis (the last three in the table). In each group, the three regularizations are of different class emphasis including emphasizing majority class, emphasizing minority class and emphasizing equally both classes. Our proposed regularization named “Emph+Diag” in the table belong to the second group with more emphasis to the minority class. For the regularization “ $S_{Wi}ToS_P$ ” in Equation (6.9), α was set to 0.5.

Figures 6.11, 6.12, 6.13, 6.14, 6.15 show the classification performance including estimated classification error rate, AUC, and robustness performance of the six forms of regularization in Table 6.1. In addition, the experimental results by SVM-RFE are also provided. The purpose of presenting the results by SVM-RFE here is to provide a reference to the generally achievable performance on each data set.

From the figures, it can be easily observed that our proposed regularization

Figure 6.11: Performance comparisons of different forms of regularization on Colon data: (a) Classification error rate. (b) AUC and (c) Robustness.

“Emph+Diag” achieves the best performances in both classification and robustness on all five gene expression data sets. If we focus further onto the comparison of “Emph+Diag” with its other two homogeneous rivals “Diag(S_W)” and “Diag(S_P)” from the group with diagonal emphasis, it is very interesting (and also what we have expected) to find that their performance in both classification and robustness deteriorate in the order of “Emph+Diag” $\rightarrow$ “Diag(S_W)” $\rightarrow$ “Diag(S_P)”, where the correspond-

Figure 6.12: Performance comparisons of different forms of regularization on Leukemia data: (a) Classification error rate. (b) AUC and (c) Robustness.

ing emphasized classes are changed from “Minority Class” → “Equal Emphasis” → “Majority Class” as indicated in Table 6.1. This is also a direct evidence to our statement that more emphasis should be given to the minority class rather to the majority class in situation of small sample size and class imbalance. Another evidence to our statement can also be found from the results of the three regularizations without diagonal emphasis where “ $\text{Emph}(S_{W_i})$ ” generally performs the best compared to its

Figure 6.13: Performance comparisons of different forms of regularization on DLBCL data: (a) Classification error rate. (b) AUC and (c) Robustness.

homogeneous rivals “ $S_W + λ * I$ ” and “ $S_{Wi}ToS_P$ ”.

As to the comparison of regularizations with and without diagonal emphasis, the superiority of regularizations with diagonal emphasis, to the regularizations without diagonal emphasis, is greatly obvious, which highly supports our analysis and argument to the importance of diagonal emphasis.

In order to investigate whether our proposed regularization alleviates the overfit-

Figure 6.14: Performance comparisons of different forms of regularization on CNS data: (a) Classification error rate. (b) AUC and (c) Robustness.

ting problem in the minority class, we first define the minority class as the positive class and the majority class as the negative class, and then calculate the averaged true positive rate (TPR, i.e. classification accuracy in minority class) and the averaged true negative rate (TNR, i.e. classification accuracy in majority class) on the testing data. The results on the five data sets are shown in Figures 6.16 to 6.20, respectively. From the experimental results, it can be observed that TPR is im-

Figure 6.15: Performance comparisons of different forms of regularization on Prostate data: (a) Classification error rate. (b) AUC and (c) Robustness.

proved with the emphasized class transferring from majority class to minority class (i.e. “ $\text{Diag}(S_p)$ ” $\rightarrow$ “ $\text{Diag}(S_w)$ ” $\rightarrow$ “ $\text{Emph}+\text{Diag}$ ”). This is an indication of alleviation of overfitting in the minority class. The situation in the majority class is more complex, where TNR could increase or decrease when more emphasis is given to minority class. The classification performance of TPR and TNR provides a direct evidence of the mitigation of the overwhelming problem. Meanwhile, the TNRs of

“ $\text{Diag}(S_W)$ ” and “ $\text{Diag}(S_P)$ ” are generally much higher than TPRs on all five data sets, this is also one reason why more emphasis should be given to minority class with a probable consequence of improved overall classification accuracy by achieving a balance between TPR and TNR. Furthermore, notice the fact in practice that the minority class often consists of diseased samples while the majority class consists of normal ones, this will make TPR mean more than TNR, as wrongly classifying a diseased sample to a normal one would produce greater risks and more serious consequences than wrongly classifying a normal sample to a diseased one. Therefore, an improvement in TPR may bring us more than what we have expected.

Figure 6.16: Classification accuracy of single class on Colon data: (a) True positive rate (TPR) and (b) True negative rate (TNR).

6.5 Summary of Chapter

Linear discriminant analysis (LDA) can be used for both classification and feature selection. When applied to small-sized data, LDA may encounter problems such as

Figure 6.17: Classification accuracy of single class on Leukemia data: (a) True positive rate (TPR) and (b) True negative rate (TNR).

Figure 6.18: Classification accuracy of single class on DLBCL data: (a) True positive rate (TPR) and (b) True negative rate (TNR).

singularity and overfitting, and hence regularization is needed. In this chapter, we have focused on regularizing LDA for feature subset selection on gene expression data that are generally of high dimensionality, small sample size and class imbalance. A new regularization technique consisting of class emphasis and diagonal emphasis has

Figure 6.19: Classification accuracy of single class on CNS data: (a) True positive rate (TPR) and (b) True negative rate (TNR).

Figure 6.20: Classification accuracy of single class on Prostate data: (a) True positive rate (TPR) and (b) True negative rate (TNR).

been proposed to deal with overfitting problem. Guidelines for setting regularization parameters have been given. The LDA-based feature selection is implemented incrementally with reduced computational burden and comparative studies on five gene expression data sets have been conducted. The experimental results validate

the effectiveness and superiority of our proposed regularization technique.

Chapter 7

A Framework of Reducing Feature Redundancy in Ranking-Based Feature Selection

7.1 Feature Redundancy: What Is Its Role?

Feature subset selection is popularly used and generally has good classification performance because of its ability to remove redundant features. The consideration of correlations between features in a feature selection process will lead to reduced feature redundancy in selected feature subsets, which could in turn result in improved classification performance. Although a feature ranking result can be easily transformed to a feature subset by choosing the top ranked features, the classification performance of a subset thus produced is usually inferior to that from set-based feature selection because of the existence of great feature redundancy. Feature redundancy, in past research, is often viewed as a detrimental factor to feature selection, and hence a popular way to treat redundancy in feature selection is to reduce it as much as possible. Due to compactness requirement in feature subset selection

(i.e. selecting feature subsets with size as small as possible, while with information as great as possible), feature redundancy reduction has attracted great attention and much related work has been done including reducing redundancy either in a preprocessing step, in a postprocessing step or in the process of feature importance evaluation. A detailed review of the related work will be given in the next section.

However, does redundancy always stand on the opposite side of a good feature selection? This is not necessarily true and we argue that a moderate degree of feature redundancy should be preserved in feature subsets for some problems, especially for problems with high dimensionality and small sample size, such as microarray gene expression data. There are several reasons. Firstly, a complete removal of feature redundancy can create a rather specific representation in feature subspace to the corresponding training data, and hence a possible overfitting problem (i.e. the selected features represent the training data well but badly to other unseen data) because of the very small number of samples. As a result, the classification performance is good on training data but awful on unseen testing data (Mao and Tang, 2011). Secondly, a very limited number of samples will result in inaccurate estimation to parameters such as correlations between features. Consequently, feature selection with excessive redundancy reduction may not only fail to eliminate really correlated features, but also reject non-redundant and very useful features. Thirdly, there will not be enough informative features left in a feature subset if feature redundancy is overly reduced, due to the fact that highly target-related features are often correlated in practical problems. A moderate degree of redundancy in feature selection can help to alleviate or solve these problems.

In addition, robust feature selection also requires a proper degree of redundancy

in the selection results. In feature selection, even though the key reason incurring instability is the small sample size (Yang and Mao, 2011), reduction of redundancy will further deteriorate the robustness in general, which will in turn degrade the interpretability and reliability of the selected features. Generally speaking, high robustness in selection results requires low compactness and equivalently high level of redundancy, and vice versa. Therefore, a moderate degree of redundancy retainment would be a good choice to keep a compromise between robustness and compactness of the selection results.

In this chapter, we propose a feature redundancy reduction framework for ranking-based feature selection techniques after considering the profusion of feature ranking algorithms in literature as well as their simplicity and generally good performance in both classification and robustness. The proposed framework is based on the relaxed orthogonal decomposition (ROD) and sequential forward feature search. After disposing feature redundancy in data level through ROD, a feature is then evaluated individually using a feature ranking evaluation criterion. Therefore the proposed framework is flexible and can be applied to any feature ranking algorithms. In addition, because of the employment of a feature ranking criterion, a feature selection algorithm based on the proposed framework will inherit the time-efficiency of the ranking-based feature selection algorithm and thus will be computationally much less intensive than the traditional set-based feature selection algorithms.

7.2 Review of Related Work

According to the existing work in literature that considered feature redundancy, approaches can basically be divided into three categories. 1) The pre-filtering methods which identify redundancy of all features in a preprocessing step and build their relationships on the basis of the redundancy for the subsequent feature selection. 2) The imbedding methods that implant redundancy deduction in the process of feature importance evaluation. 3) The post-filtering methods which reduce feature redundancy in a postprocessing step after all features are evaluated or ranked. The pre- and post-filtering methods are relatively simple and most of the past work focused on the second, the imbedding methods.

The first and the third categories of methods together can be grouped into the redundancy filtering methods which filtrate redundant features in a step independent of feature importance evaluation. The pre-filtering methods often use clustering or fuzzy clustering algorithms in redundancy detection. For example, Knijnenburg et al. (2005) clustered together correlated features by a Pearson correlation based complete linkage hierarchical clustering algorithm and selected the most important feature from each cluster to produce a non-redundant feature subset for the succeeding feature ranking. Jaeger et al. (2003) employed the fuzzy clustering first for the correlation-similar feature grouping and then selected one or more representative features from each cluster to form the final feature subset. As to the post-filtering methods, Yu and Liu (2004) proposed a new feature selection framework consisting of relevance analysis succeeded with redundancy analysis, and devised the FCBF (Fast Correlation-based Filter) algorithm as a realization of the new framework

for reduced redundancy feature selection. Wang and Ma (2005) developed a post-filtering feature selection algorithm that reduces redundant features by analyzing the correlation among features in the pre-selected feature subset. Osl et al. (2009) proposed a new algorithm to reduce redundant features through two iterative steps including redundancy detection and redundant feature re-positioning in the feature ranking.

Comparing to the relatively rare work in the pre- and post-filtering methods, the research employing the imbedding methods is prevalent. A brief review is given below.

Ding and Peng (August 2003) as well as their further work in Peng et al. (2005) incorporated feature relevance and redundancy in the process of feature evaluation and proposed the minimal-redundancy-maximal-relevance (MRMR) feature selection framework. A detailed review of their work is given in Section 3.3.1 in Chapter 3.

Ooi et al. (2006) advanced the MRMR by a weighted combination of relevance and redundancy and proved through extensive experiments on practical data sets that the best feature subsets were generally yielded by giving unequal-priority (rather than equal-priority) to relevance and redundancy, but how to determine the optimal priority was still an unresolved problem.

Apart from explicitly incorporating feature relevance and redundancy, implicit techniques also exist. For instance, Zhou et al. (2010) improved the trace-based class separability criterion by imposing extra redundancy constraints to avoid selecting redundant features. They then proposed the redundancy-constrained feature selection (RCFS) algorithm which is based on the TUM (totally unimodular) condi-

tion in integer programming to ensure its optimality and efficiency. Mao and Tang (2011) devised a recursive algorithm to employ the Mahalanobis separability measure for a reduced correlation feature selection. Their algorithm always calculates the Mahalanobis measure in a two dimensional space and can greatly decrease the computational burden.

Other work also includes all wrapper feature selection methods with sequential forward or backward feature search and the multivariate filter methods (Saeys et al., 2007) such as Relief, which consider feature interactions during evaluating feature importance.

7.3 Impact of Feature Redundancy: A Case Study

To investigate and illustrate the impact of redundancy to the performance of feature selection, a case study is conducted and presented in this section based on the univariate feature selection method of Fisher's ratio and its corresponding multivariate form linear discriminant analysis (LDA) based feature selection. The details about Fisher's ratio and LDA-based feature selection, and their relationship are described in Section 6.2.1.

The LDA-based feature selection is to evaluate a feature subset $\mathbf{F}^{(t)}$ in the form:

$$FR(\mathbf{F}^{(t)}) = (\mathbf{m}_{(1)} - \mathbf{m}_{(2)})\mathbf{S}_W^{-1}(\mathbf{m}_{(1)} - \mathbf{m}_{(2)})^T \quad (7.1)$$

The scatter matrix $\mathbf{S}_W$ contains the covariances of all pairs of features in $\mathbf{F}^{(t)}$ and the role of $\mathbf{S}_W^{-1}$ is to minimize the redundancy when evaluating the features in

the feature subset.

If no correlation between features is considered, the scatter matrix $\mathbf{S}_W$ will shrink to its diagonal:

$$\mathbf{S}_W \leftarrow \text{diag}(\mathbf{S}_W) = \text{diag}(\sigma_1^2, \sigma_2^2, \dots, \sigma_t^2) \quad (7.2)$$

where $\sigma_j^2 = \sigma_{j(1)}^2 + \sigma_{j(2)}^2$, and $\sigma_{j(c)}^2$ is the variance of feature X_j in class c , for $c = 1, 2$.

Then the LDA-based feature selection will reduce to the Fisher's ratio for evaluating the t features in $\mathbf{F}^{(t)}$:

$$FR(\mathbf{F}^{(t)}) = FR(X_1) + FR(X_2) + \dots + FR(X_t) \quad (7.3)$$

Therefore, retaining certain degree of redundancy (correlation) in feature subset can be achieved by partly shrinking $\mathbf{S}_W$ to its diagonal:

$$\mathbf{S}_W(\beta) = \beta \mathbf{S}_W + (1 - \beta) \text{diag}(\mathbf{S}_W) \quad (7.4)$$

where $\beta \in [0, 1]$ is a control parameter. The greater β is, the more redundancy is considered, and hence the less redundancy is retained in the feature subset in feature selection.

A LDA-based feature selection algorithm can be easily constructed through a straightforward combination of the LDA feature subset evaluation as shown in Equation (7.1) and sequential forward feature search. That is, a feature subset $\mathbf{F}^{(t-1)}$ containing $t - 1$ features has been selected by previous $t - 1$ steps, and then at step t , a candidate feature X_j from all remaining features is evaluated using Equation (7.1) on the candidate feature subset $\mathbf{F}_{tmp}^{(t)} = \mathbf{F}^{(t-1)} \cup \{X_j\}$ and the one with maximal

evaluation value is selected to form the current selected feature subset $\mathbf{F}^{(t)}$. This procedure continues until a stop criterion is satisfied, such as a predefined number of features are selected.

Theoretically, LDA-based feature selection will perform better than Fisher's ratio because of its consideration of feature redundancy. But this is not always true, particularly when problems are of high dimensionality and very small sample size. For the purpose of investigation and illustration of the impact of redundancy in feature selection, the following experiment has been carried out on the typical high dimensional and small sized gene expression Colon data (details can be found in Section 7.6 and the reference therein) based on LDA with different β values.

Firstly, a training data set is generated through resampling with replacement on the full data set (duplicated samples are removed). Secondly, the LDA-based feature selection is applied to the training data until d genes are selected. Here d is set to 50 because of the high computational complexity involved in LDA. In addition to the shrinkage as described in Equation (7.4), a small value $\lambda = 0.0001$ is added to the diagonal of $\mathbf{S}_W(\beta)$ for consideration of the possible singular problem in computing $\mathbf{S}_W^{-1}$.

To obtain reliable estimation of the performance, the above 2-step procedure is repeated 100 times. Figure 7.1 shows the experimental results including the estimated classification error rate (Figure 7.1(a)) and the robustness index (Figure 7.1(b), see Equation (3.12) for the robustness measure) in 9 different degrees of redundancy reduction.

According to our previous statement on the relationship among the redundancy, robustness and compactness, our expectation on the experimental results is that

Figure 7.1: Performance of different degrees of redundancy reduction: (a) Classification error rate and (b) Robustness.

the increased redundancy reduction degree leads to deterioration in robustness and improvement in classification performance. This is reasonable because a particular training data containing small samples may need a moderately specific representation in the feature subspace thus to have the best class separation ability. But the more specific in feature representations, the greater differences between the feature subsets, i.e. the instability problem in feature selection.

Each curve in Figure 7.1 denotes the results with a predefined feature redundancy reduction degree (denoted by β). It can be observed from Figure 7.1(b) that the robustness decreases with increased degrees of redundancy reduction, which is consistent with our expectation. The classification performance results (in Figure 7.1(a)) are a little complicated: 1) The best classification performance is obtained with $\beta = 0.4$ (rather than with $\beta = 1$). 2) When $\beta \leq 0.4$, the increase in β (i.e. increase in degrees of redundancy reduction) results in improvement in classification, which is in line with our expectation on classification performance. 3) When $\beta > 0.4$, the

increase in β continues to deteriorate the classification performance. This is probably because a higher degree of redundancy reduction will be much easier to exclude informative genes and thus to include noisy genes in the selected gene subset. 4) The worst performance in both classification and robustness is obtained with $\beta = 1$.

From the above observations and analyses to the experimental results, the following conclusions and suggestions could be made:

- ◇ The feature subset discrimination ability (i.e. classification) can benefit from redundancy reduction, but with a consequence of deterioration in robustness.
- ◇ A moderate rather than an acute degree of redundancy reduction should be used for the purpose of obtaining a good classification performance. An overly redundancy reduction will harm both classification and robustness performance.
- ◇ A compromise between classification and robustness may need to be made for a robust feature selection. E.g. $\beta = 0.1$ may be a good choice in the above case study.

7.4 A Feature Redundancy Reduction Framework Based on Orthogonal Decomposition and Sequential Forward Search

The LDA-based feature selection employed in the above case study needs to evaluate all remaining candidate features in each step and to compute the inverse of the

scatter matrix $\mathbf{S}_W$ for evaluating each candidate feature. Suppose we want to select a subset of d features from totally p features, then the computational complexity merely involved in the inverse operation will be $O(p \cdot d^4)$. Take gene selection for example, the p and d are generally on the order of 10^4 and 10^2 , respectively, the computational complexity in calculating $\mathbf{S}_W^{-1}$ will accordingly reach to $O(10^{12})$. The consequence of the high computational complexity could make LDA-based feature selection very time-consuming and even computationally prohibitive when applied to practical problems with very high dimensionality such as gene expression data. In order to reduce the computational overhead, we propose to utilize the orthogonal decomposition (OD) for decorrelation between features before using the feature ranking algorithms. As it will be seen below, this is a very time-efficient way to dispose correlation between features. After combined with the sequential forward search, an OD-based framework is developed for feature redundancy reduction. The details are described in the following parts.

7.4.1 Feature Evaluation Based on Orthogonal Decomposition

According to the Orthogonal Decomposition Theory (Golub and Van Loan, 1996), a Hilbert space $\mathbf{H}$ can be completely represented by its closed subspace $\mathbf{A}$ and the orthogonal complement space $\mathbf{A}^\perp$ of $\mathbf{A}$. Therefore, for every element $\mathbf{e} \in \mathbf{H}$, there exists a best approximation $\mathbf{e}_0$ of $\mathbf{e}$ in $\mathbf{A}^\perp$, such that $\mathbf{e} - \mathbf{e}_0 \in \mathbf{A}$. Our basic idea derived from this theory to decorrelate features and thus to reduce redundancy in feature selection is depicted as follows. Given a subspace $\mathbf{A}_t$ that is spanned by

t mutually orthogonal column vectors $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_t$ ($\mathbf{z}_i^T \mathbf{z}_{i'} = 0$ for all $i \neq i'$ and their corresponding features are $Z_1, Z_2, \dots, Z_t$), and a candidate feature X_j with its realization $\mathbf{x}_j$, we first decompose $\mathbf{x}_j$ into,

$$\mathbf{x}_j = \mathbf{v}_j + \sum_{i=1}^t \alpha_{ji} \mathbf{z}_i \quad (7.5)$$

where $\mathbf{v}_j \in \mathbf{A}_t^\perp$ is called the residual vector of $\mathbf{x}_j$ out of $\mathbf{A}_t$ and its corresponding feature V_j is the OD (orthogonal decomposition) feature of X_j . The orthogonal decomposition coefficients α_{ji} are given by:

$$\alpha_{ji} = \frac{\mathbf{x}_j^T \mathbf{z}_i}{\mathbf{z}_i^T \mathbf{z}_i} \quad (7.6)$$

It can be easily proved that $\mathbf{v}_j^T \mathbf{z}_i = 0$ and hence $\mathbf{v}_j \perp \mathbf{z}_i$, for $i = 1, 2, \dots, t$. If we assume at least one of the expected values of the features V_k and Z_i is zero, then $\mathbf{v}_j^T \mathbf{z}_i = 0$ also implies that the Pearson correlation coefficient calculated from $\mathbf{v}_j$ and $\mathbf{z}_i$ is zero, which means the corresponding variables V_j and Z_i are linearly uncorrelated.

From linear algebra and information theory, if a vector $\mathbf{b}$ can be completely represented by a linear combination of a group of vectors $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \dots]$, it is understood that there will be no new information (e.g. separation ability) added when vector $\mathbf{b}$ is added into $\mathbf{B}$ (i.e. the information embodied in $\mathbf{b}$ is completely covered by $\mathbf{B}$). Therefore, the orthogonal decomposition in Equation (7.5) will imply that the information hidden in $\mathbf{x}_j$ is disparded and the new information subject to $\mathbf{A}_t$ is contained in $\mathbf{v}_j$. In other words, for an information measurement criterion

$h(\cdot)$, the feature X_j can be evaluated:

$$h(\mathbf{x}_j|\mathbf{A}_t) = h(\mathbf{v}_j) \quad (7.7)$$

7.4.2 The Procedure to Reduce Feature Redundancy through Orthogonal Decomposition and Forward Search

Given the two-class problem as described in Equation (2.41) and a feature information (importance) measure $h(\cdot)$, the procedure is summarized below:

- (i) Normalize data to zero mean and unit standard deviation along dimension of all features.
- (ii) Initialize the selected feature subset $\mathbf{F}_S$ and its corresponding OD feature set $\mathbf{F}_{OD}$ to be empty.
- (iii) For all p features in $\mathbf{X} = \{X_1, X_2, \dots, X_p\}$, evaluate their importance using $h(\cdot)$ and select the most important feature, say, X'_1 , add X'_1 to $\mathbf{F}_S$ and remove it from $\mathbf{X}$. Let $Z'_1 = X'_1$ (their corresponding realizations are $\mathbf{z}'_1$ and $\mathbf{x}'_1$) and add it to $\mathbf{F}_{OD}$. Set $t = 1$.
- (iv) For all remaining $(p-t)$ features in $\mathbf{X}$, do the orthogonal decomposition subject to $\mathbf{F}_{OD}$ using Equation (7.5) and Equation (7.6) to find their residual vectors $\mathbf{v}'_s$. Let $t = t + 1$.
- (v) Evaluate the importance of all the remaining features in $\mathbf{X}$ using their residual vectors by $h(\cdot)$, select the most important feature, say, X'_t , add X'_t to $\mathbf{F}_S$ and remove it from $\mathbf{X}$. Let $Z'_t = V'_t$ (with realization $\mathbf{v}'_t$) and add it to $\mathbf{F}_{OD}$.

- (vi) Repeat steps (iv) to (v) until a stopping criterion is satisfied, say a predefined number of features in $\mathbf{F}_S$ are selected.

If we denote a feature by a circle and its information by the area in the circle, the relationship of the second selected feature X'_2 to X'_1 as well as the third selected feature X'_3 to $\{X'_1, X'_2\}$ can be depicted in Figure 7.2(a) and Figure 7.2(b), respectively. Figure 7.2(c) shows their space projection relationships in a three dimensional space.

In Figure 7.2(a), the information solely hidden in features X'_1 and X'_2 , are $h(X'_1) = a + b$ and $h(X'_2) = b + c$, respectively, and their common information is c . After orthogonal decomposition of X'_2 subject to X'_1 , a new feature Z'_2 (or OD feature) is obtained. The conditioned information of X'_2 is then fully contained in Z'_2 (i.e. $h(X'_2|X'_1) = h(Z'_2) = b$). The situation of Figure 7.2(b) is similar, where the conditioned information of the third selected X'_3 , is fully contained in the new OD feature Z'_3 (i.e. $h(X'_3|X'_2, X'_1) = h(Z'_3) = d$). In Figure 7.2(c), the second selected feature X'_2 is first projected onto the orthogonal space of Z'_1 (X'_1) and it is then evaluated based on its OD feature Z'_2 . In the same way, X'_3 is first projected and then evaluated based on its OD feature Z'_3 .

7.4.3 Saturation Problems in the OD-Based Feature Redundancy Reduction Framework

The basic idea of the above OD-based framework is to estimate the importance of a candidate feature X_j at t -th selection step using its residual obtained by projecting it onto the orthogonal complement space of $\mathbf{A}'_{t-1}$, where $\mathbf{A}'_{t-1}$ is the

Figure 7.2: Illustration of the process when evaluating a candidate feature in the OD-based framework: (a) Evaluating the second feature. (b) Evaluating the third feature and (c) The space relationship of the first three selected features.

space spanned by the OD features $\mathbf{F}_{OD} = \{Z'_1, Z'_2, \dots, Z'_{t-1}\}$ of all selected features $\mathbf{F}_S = \{X'_1, X'_2, \dots, X'_{t-1}\}$. In situation of small sample size and high dimensionality, i.e. $n \ll p$, two problems will occur. The first problem is the *space saturation problem* which means that at most n features from the total p features can be selected.

This is because after n features are selected, the space $\mathbf{A}'_{n+1}$ does not exist and hence a candidate feature cannot be projected for evaluation. This is an intrinsic problem in the setting of $n < p$. Meanwhile, n out of p features are generally far from adequate to represent the feature space of a classification problem, and readers can refer to the experimental results in Section 7.6 to see this inadequateness. For example, on Colon data, the number of training samples is about $62 \times 63.2\% \cong 39$, while the classification results (see Figure 7.7(a)) with 39 features are obviously not satisfying. If the selection procedure based on the above framework is forced to continue, new features can still be selected but all candidate features will be of weak difference as the residual vector of every candidate feature will approach to a zero vector, which will manifest as the *score saturation problem*, the second problem incurred by $n \ll p$. In addition, when n is very small, the samples may probably not precisely describe a feature and consequently the estimated relationships (e.g. angles in space) between features will also be imprecise, which is another reason that leads to the score saturation problem during evaluating candidate features.

The space saturation problem is easy to understand and here we will focus on the score saturation problem. To illustrate this problem the following experiment was conducted. The Fisher's ratio was used as the feature importance evaluation criterion in the OD-based framework and then the framework was applied on the full Colon data. The scores (i.e. Fisher's ratio values after orthogonal decomposition) of the top 100 genes were calculated and the accumulated scores were shown in Figure 7.4. From the figure, the accumulated scores do not increase after about 30 genes are selected, which means the scores have saturated. Actually, it is obtained from the experimental results that the increased scores are less than 10^{-3} in each

step after 34 genes are selected. Notice that the gene scores computed using Fisher's ratio will never be less than 0, hence the scores of all left candidate genes (e.g. there are 1966 candidate genes when selecting the 35-*th* gene) will be in a very small range of $[0, 10^{-3})$, which may result in selecting a noisy gene rather than a really useful one in selecting 34-*th* gene onwards.

Figure 7.3: The accumulated Fisher's ratio scores from the OD-based framework.

7.5 Relaxed Orthogonal Decomposition-Based Framework

Even though the OD-based framework can theoretically select a group of most informative features after deducting their mutual dependencies, it encounters the saturation problems in practice as analyzed and illustrated above. The root cause of the problem is the small sample size. Another cause, we think, is the stringent orthogonal decomposition. That is, a candidate feature is required to orthogonally decomposed on basis of all selected features, which makes its residual approach zero

after a few features are selected. An intuitional way to alleviate or avoid the saturation problems is to introduce relaxation to the orthogonal decomposition and orthogonally decompose a candidate feature on basis of only part of, rather than all the selected features. The relaxed orthogonal decomposition-based (abbreviated as ROD-based) feature evaluation is described below.

When a candidate feature is orthogonally decomposed on basis of q of all selected features, this is termed the q -th-order orthogonal decomposition. Assume a number of k features have been selected, then the way to implement the q -th-order orthogonal decomposition for feature evaluation is: when $k \leq q$, a candidate feature is orthogonally decomposed on basis of all the k features and evaluated using its residual; otherwise all $\binom{k}{q}$ orthogonal decompositions are first done and then the candidate feature is evaluated by the average importance over these decompositions.

The order number q controls the extent of relaxation and the consequent computational complexity. When $q = k$ the relaxed orthogonal decomposition (ROD) will reduce to the conventional orthogonal decomposition as used in the OD-based framework. For simplicity and time efficiency, the first-order (i.e. $q = 1$) is used in our proposed ROD-based framework as implemented in Algorithm 2. Note that $|\mathbf{X}|$ is the cardinality (number of elements) of set $\mathbf{X}$, $\mathbf{X}(j)$ is the j -th element in set $\mathbf{X}$, and the function $ODecomp(\mathbf{X}_l, \mathbf{X}_j)$ orthogonally decomposes feature $\mathbf{X}_l$ on basis of feature $\mathbf{X}_j$.

The procedure in Algorithm 2 resembles that of the OD-based framework as described in Section 7.4.2. Their difference is in selecting the orthogonal decomposition order q when evaluating a candidate feature. The relationship of the first three selected features based on the ROD-based framework is depicted in Figure 7.4.

Algorithm 2: ROD-Based Framework

```

begin
  Set  $d$  //the number of features that are required to select
  Given the full feature set  $\mathbf{X} = \cup_{j=1}^p \{\mathbf{X}_j\}$ , and
  the selected feature subset  $\mathbf{F}_S = \{ \}$ , and
  the feature evaluation criterion  $h(\cdot)$ 

  Normalize data to zero mean and unit standard deviation along dimension
  of all features
  repeat
    for  $l \leftarrow 1$  to  $|\mathbf{X}|$  do
      if  $|\mathbf{F}_S| > 0$  then
        for  $j \leftarrow 1$  to  $|\mathbf{F}_S|$  do
           $Z_{lj} = ODecomp(\mathbf{X}(l), \mathbf{F}_S(j));$ 
        end
         $r_l = \frac{1}{|\mathbf{F}_S|} \sum_{j=1}^{|\mathbf{F}_S|} h(Z_{lj}) ;$ 
      else
         $r_l = h(\mathbf{X}(l));$ 
      end
    end
     $l^* = \arg \max \{r_l\};$ 
    //find the most important feature  $\mathbf{X}(l^*)$ 
     $\mathbf{F}_S = \mathbf{F}_S \cup \{\mathbf{X}(l^*)\}, \mathbf{X} = \mathbf{X} \setminus \{\mathbf{X}(l^*)\}$ 
  until  $d$  features are selected in  $\mathbf{F}_S$  ;
end

```

Figure 7.4: Illustration of the relationships when evaluating the first three selected features X'_1, X'_2, X'_3 in the ROD-based framework.

The ROD-based framework utilizing the first-order orthogonal decomposition is also computationally very efficient. In addition, other two advantages would be

obtained compared the OD-based framework. 1) The saturation problems described above could be avoided; and 2) the relaxation in orthogonal decomposition may help to retain a certain degree of feature redundancies, which may subsequently result in improved robustness and classification performance in feature selection.

An experiment similar to the one described in Section 7.4.3 was carried out and the log-transformed accumulated scores from both OD-based framework and ROD-based framework with Fisher's ratio criterion are shown in Figure 7.5. It can be seen that the score saturation problem has been overcome.

Figure 7.5: A comparison of accumulated Fisher's ratio scores from the OD-based framework and ROD-based framework.

We expect that the ROD-based framework can retain a good robustness performance by keeping a moderate degree of feature redundancy. To demonstrate this, the following experiment on Colon data was conducted. Firstly, a training data was generated by sampling with replacement (with duplicated samples removed) on the full Colon data. Secondly, a feature selection algorithm was applied to the training data to get the top 100 genes. Assume $\mathbf{F}_S^{(k)}$ contains the top k genes $X'_1, X'_2, \dots, X'_k$,

the correlation level in $\mathbf{F}_S^{(k)}$ is computed as:

$$CL(k) = \frac{\sum_{i=1}^{k-1} \sum_{j=i+1}^k |C_{pw}(i, j)|}{k(k-1)/2} \quad (7.8)$$

where $C_{pw}(i, j)$ is the correlation between a pair of genes X'_i and X'_j . In this study, $C_{pw}(i, j)$ employs Pearson correlation coefficient and is estimated using the full data set.

To obtain reliable evaluation of the correlation level, the above experiment was repeated 300 times and an averaged correlation level was calculated and is shown in Figure 7.6(a). The corresponding robustness measure (please refer to Equation (3.12) in Chapter 3) was also calculated and shown in Figure 7.6(b).

Figure 7.6: (a) Comparison of the correlation level and (b) Comparison of robustness

Figure 7.6 shows the results of three feature selection methods: OD-based framework with Fisher's ratio ("OD-basedFramework"), ROD-based framework with Fisher's ratio ("ROD-basedFramework") and the direct Fisher's ratio ranking ("Fisher'sRatioRanking"). Obviously, the above results are consistent with the theoretical expectation that the

OD-based framework will bring in the lowest correlation level but with the worst robustness performance, while the direct Fisher's ratio will have reverse results to those of OD-based framework and the ROD-based framework will achieve a balance between these two methods on both correlation level and robustness performance. For the results by OD-based framework, we are mainly interested in the range where the number of genes is below 30 because of its saturation problems as pointed out in the previous section. The robustness of the OD-based framework was rather bad even though it gained the lowest correlation level (i.e. greatest compactness) in gene subsets. After a more careful comparison, it can be observed that the ROD-based framework obtains a good robustness performance close to the direct Fisher's ratio ranking but with only a slight deterioration on correlation level relative to the OD-based framework.

7.6 Experimental Studies

To test the regularized framework proposed for redundancy reduction, experiments were carried out on the same microarray gene expression data sets used in Chapter 5. In the experiments, the estimated classification error rate and the AUC were given as the assessment of classification performance. In addition to the classification performance measures, robustness of the feature selection results was also evaluated using the robustness measure in Equation (3.12).

Figures 7.7, 7.8, 7.9, 7.10, 7.11 show the experimental results including the estimated classification error rate, the AUC and the feature subset robustness. In addition to the direct Fisher's ratio ("FR" in the figures) and the ROD-based framework

with Fisher’s ratio (“ROD+FR” in the figures), the SVM-RFE Guyon et al. (2002) was also used. In this study, the results by SVM-RFE are presented mainly for a reference to the generally achievable classification performance on each data rather than for the purpose to compare the other algorithms with SVM-RFE. Besides, the experimental results by the traditional OD-based framework with Fisher’s ratio (“OD+FR” in the figures) are also exhibited in the figures.

After comparing the results by the direct feature ranking (FR) and by the feature ranking with redundancy reduction (ROD+FR), it can be observed that there are improvements in the classification performance (lower estimated classification error rates and higher AUC values) after redundancy reduction, but generally with a corresponding light decrease in the robustness performance. This is in line with our expectation to the feature redundancy reduction as described in previous sections. One exception in the experiments occurred on the CNS data, where the classification performance was improved but still with an improvement in the robustness. Because of the saturation problems and the excessive redundancy reduction in the OD-based framework, the OD+FR overall performs awful in both classification and robustness.

Meanwhile, it can also be observed that SVM-RFE performs very badly in robustness even though it generally achieves a good performance in classification. Furthermore, if we assume that a lower robustness implies a higher degree of redundancy reduction as we can observe from Figure 7.1(b) in the case study in Section 7.3, then the experimental results may indicate that SVM-RFE demotes extensive redundancy and a more proper degree of redundancy has been preserved by ROD+FR because of its generally superior performance in classification than SVM-RFE and FR as well as its medial robustness performance between FR and SVM-RFE.

Figure 7.7: Performance comparisons on Colon data: (a) Classification error. (b) AUC and (c) Robustness.

7.7 Summary of Chapter

In this chapter, we have focused on the feature redundancy issue in feature selection and argued that a moderate degree of feature redundancy should be retained in feature selection for gene expression data. Coherent to this argument, we have proposed a feature redundancy reduction framework based on relaxed orthogonal

Figure 7.8: Performance comparisons on Leukemia data: (a) Classification error. (b) AUC and (c) Robustness.

decomposition and sequential forward feature search. The proposed framework is very efficient in computations and can be applied to any feature ranking criterion. Experimental studies on five gene expression data sets have been conducted and the experimental results show that the classification performance has been improved with good performance in robustness.

Figure 7.9: Performance comparisons on DLBCL data: (a) Classification error. (b) AUC and (c) Robustness.

Figure 7.10: Performance comparisons on CNS data: (a) Classification error. (b) AUC and (c) Robustness.

Figure 7.11: Performance comparisons on Prostate data: (a) Classification error. (b) AUC and (c) Robustness.

Chapter 8

Conclusions and Future Work

8.1 Thesis Conclusions

Many practical applications are characterized by the property of high dimensionality but relatively very small sample size. When carrying out data mining and machine learning tasks in those applications, the curse of dimensionality is encountered. Feature selection plays a very crucial role to the success of a mining task in high-dimensional and small-sized (HDSS) problems and much work has already been done. There are several discovered difficulties built in different feature selection techniques when applied to HDSS data, e.g. the nearly intractable computational complexity in most wrapper methods. Robustness, we think, is rather significant and ubiquitous in feature selection of HDSS problems, while it is relatively overlooked. Although some attention in recent years has been paid to robust feature selection, the attention is far from enough. The robustness of feature selection is now becoming increasingly important with the application of feature selection spreading to knowledge discovery where discovery of strongly target-correlated features can be used to direct future research. The intrinsic cause resulting in instability in feature selection of HDSS problems roots in the characteristic of the data. However, this can not prevent us from enhancing robustness by employing and devising more delicate

feature selection techniques. In this thesis, we focus on the robustness of feature selection for HDSS problems and do work in the following aspects:

Aspect 1: Background and preliminaries in pattern recognition and feature selection

An overview of typical pattern recognition systems is first presented in Chapter 2, with the introduction to the concepts and techniques in a pattern classification system. This provides readers the context where feature selection stays. The feature selection as well as the robustness issue of feature selection are then discussed in Chapter 3, which brings readers into the scope of feature selection: the basic concepts and definitions, the necessities of feature selection and its robustness issue, the exiting work where we can start our footsteps, and the directions to which we could go.

Aspect 2: Design of new feature selection algorithms for robust feature selection

This aspect contains the work in Chapter 4 and Chapter 5. In Chapter 4, we focus on investigating the combination of multiple criteria for feature ranking. The problems existing in multi-criterion combination are analyzed and a novel feature importance transformation algorithm is proposed for normalization between different feature selection criteria. The experimental results provide primary knowledge for further research. Based on the idea of multi-criterion combination and the study in Chapter 4, we develop the new feature selection algorithm: MCF-RFE (multi-criterion fusion-based recursive feature elimination) in Chapter 5. This new feature selection algorithm has been tested by extensive experiments and the results show that it can gain good performance in both robustness and classification.

Aspect 3: Improvement of existing feature selection algorithm for robust feature selection

This aspect refers to the work in Chapter 6. Linear discriminant analysis (LDA)-based feature selection is a good feature subset selection algorithm due to its consideration of correlations between features. But when applied to gene expression data which usually have high dimensionality, small sample size and class imbalance, LDA-based feature selection encounters problems such as singularity of scatter matrix, overfitting, overwhelming and prohibitive computational complexity. We propose a new regularization technique emphasizing more on minority class with the expectation of improving overall performance by alleviating overwhelming of majority class to minority class and overfitting in minority class. This regularization technique was compared with other existing regularization techniques by experimental studies on five gene microarray problems and the results show that LDA with the new regularization can produce gene subsets with better performance in both classification and robustness.

Aspect 4: Investigation of feature redundancy reduction and robustness in feature subset selection

This aspect refers to the work in Chapter 7. We explore the relationship between feature redundancy and feature selection robustness by experiments on gene expression data and conclude that a moderate feature redundancy reduction will help to achieve best classification performance while still retain good performance in robustness. Thus, we argue that a moderate rather than conventionally utmost reduction of feature redundancy should perform. Based on this argument, we propose a framework of reducing feature redundancy which is flexible and can be applied

to any feature ranking algorithms. An implementation of this framework on Fish's ratio algorithm and the experimental studies on gene expression data sets prove the effectiveness of the proposed framework.

8.2 Future Work Recommendations

The robustness is a very important yet a great challenge for feature selection of HDSS problems. Although there exist some efforts contributing to robust feature selection, these efforts are still rather insufficient. In our opinion, how to improve robustness will continue to be a very promising topic in future research of feature selection. Some possible directions may deserve more attention and concerns:

1. **Multi-criterion combination:** This is a well-known strategy to tackle the inconsistency problems deriving from the perturbations of input training data or the discrepancies between feature selection algorithms. There are two factors affecting the combination results: i) basis criteria used for combination and ii) combination methods. Past studies on classifier integration provide a rule that basis classifiers should be independent in the production of their errors. Our preliminary experimental results show that when two basis criteria have similar classification performance among several basis criteria, the combination performance will not change or change slightly if we remove one of the two performance-similar criteria. Therefore, basis criteria should be elaborately chosen. The existing combination methods are very limited and may possibly encounter problems as shown in Chapter 4 when applied to HDSS problems, hence, more effective combination methods are required.

2. **Imbalance between classes:** Data from practical domains may possibly have the characteristic of class imbalance, in addition to the high dimensionality and small sample size. Most existing research paid attention to the latter characteristic, but usually neglected the former one. As proved in our study in Chapter 6, a proper utilization of the class imbalance could be beneficial to both classification and robustness performance in feature selection. Therefore, we may expect that many other feature selection algorithms may benefit from a consideration of the class imbalance property in the data.
3. **Sample selection:** A sample selection process has been proved to be effective in the traditional classification problem. In the HDSS problems, an outlier in the small sample size data set may produce more negative or destructive effect than an outlier in a data set with large number of samples. Thus, a sample selection process may help to improve the feature selection performance. An oversampling process can also be considered before the sample selection.

Author's Publications

1. **F. Yang** and K. Z. Mao, "Robust Feature Selection for Microarray Data Based on Multi-Criterion Fusion," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 8, no. 4, pp. 1080–1092, 2011.
2. **F. Yang** and K. Z. Mao, "Improving Robustness of Gene Ranking by Multi-Criterion Combination with Novel Gene Importance Transformation," *International Journal of Data Mining and Bioinformatics*, accept.
3. K. Z. Mao, **F. Yang** and W. Tang, "Regularized Linear Discriminant Analysis and Its Recursive Implementation for Gene Subset Selection," *The Eighth Annual IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, Paris, France, 2011.
4. **F. Yang** and K. Z. Mao, "Improving Robustness of Gene Ranking by Resampling and Permutation Based Score Correction and Normalization," *Proceedings of the 2010 IEEE International Conference on Bioinformatics and Biomedicine*, Hong Kong, pp. 444–449, 2010.
5. **F. Yang**, and K. Z. Mao, "Entropy-based Criteria Dealing with the Ties Problem in Gene Selection," *Proceedings of 2009 Eighth IEEE/ACIS International*

- Conference on Computer and Information Science*, pp. 124-129, 2009.
6. G. Zhao, G. Li, C. Wen and **F. Yang**, “On the Convergence of Normalized Iterative Identification of Hammerstein Systems,” *Proceedings of 2011 International Conference on Automation and Robotics, ICAR2011*, Dubai, 2011.
 7. G. Li, C. Wen, Z. G. Li, **F. Yang** and K. Z. Mao, “Model Based Online Learning with Kernels,” *IEEE Transactions on Neural Networks*, Revised and Resubmitted (under second round review), 2011.
 8. **F. Yang**, K. Z. Mao and W. Tang, “Emphasizing Minority Class in Regularized LDA for Gene Subset Selection,” submitted to *IEEE/ACM Transactions on Knowledge and Data Engineering*.

Bibliography

T. Abeel, T. Helleputte, Y. Van de Peer, P. Dupont, and Y. Saeys. Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics*, 26:392–398, February 2010. ISSN 1367-4803.

D. W. Aha, D. Kibler, and M. K. Albert. Instance-based learning algorithms. *Machine Learning*, 6(1):37–66, January 1991. ISSN 0885-6125. doi: 10.1007/BF00153759.

C. F. Aliferis, I. Tsamardinos, and A. Statnikov. HITON: a novel markov blanket algorithm for optimal variable selection. *AMIA Annual Symposium proceedings*, pages 21–25, 2003. ISSN 1559-4076.

A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu, J. I. Powell, L. Yang, G. E. Marti, T. Moore, J. J. Hudson, L. Lu, D. B. Lewis, R. Tibshirani, G. Sherlock, W. C. Chan, T. C. Greiner, D. D. Weisenburger, J. O. Armitage, R. Warnke, R. Levy, W. Wilson, M. R. Grever, J. C. Byrd, D. Botstein, P. O. Brown, and L. M. Staudt. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769):503–511, 2000. ISSN 0028-0836. doi: 10.1038/35000501.

- H. Almuallim and T. G. Dietterich. Learning with many irrelevant features. In *Proceedings of the ninth National conference on Artificial intelligence - Volume 2*, AAAI'91, pages 547–552. AAAI Press, 1991. ISBN 0-262-51059-6.
- U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci. USA*, 96(12):6745–6750, 1999.
- A. Altmann, L. Tološi, O. Sander, and T. Lengauer. Permutation importance: a corrected feature importance measure. *Bioinformatics (Oxford, England)*, 26(10):1340–1347, May 2010. ISSN 1367-4811. doi: 10.1093/bioinformatics/btq134.
- A. Antoniadis, S. Lambert-Lacroix, and F. Leblanc. Effective dimension reduction methods for tumor classification using gene expression data. *Bioinformatics*, 19(5):563–570, 2003. doi: 10.1093/bioinformatics/btg062.
- A. Arauzo-Azofra, J. M. Benitez, and J. L. Castro. Consistency measures for feature selection. *J. Intell. Inf. Syst.*, 30:273–292, June 2008. ISSN 0925-9902. doi: 10.1007/s10844-007-0037-0.
- R. Bellman. *Adaptive Control Processes*. Princeton University Press, 1961.
- M. Ben-Bassat. Pattern recognition and reduction of dimensionality. In P. R. Krishnaiah and eds. L. N. Kanal, editors, *Handbook of Statistics*, pages 773–791. North Holland, 1982.
- J. Biesiada, W. Duch, A. Kachel, K. Maczka, and S. Palucha. Feature ranking methods based on information entropy with parzen windows. In *Proceedings of*

- the 9th International Conference on Research in Electrotechnology and Applied Informatics (REI'05)*, pages 109–119, Katowice-Kraków, Poland, 2005.
- C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 0387310738.
- A. Bonarini. An introduction to learning fuzzy classifier systems. In *Learning Classifier Systems, From Foundations to Applications*, pages 83–106, London, UK, 2000. Springer-Verlag. ISBN 3-540-67729-1.
- B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory, COLT '92*, pages 144–152, New York, NY, USA, 1992. ACM. ISBN 0-89791-497-X.
- A. P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recogn.*, 30:1145–1159, July 1997. ISSN 0031-3203. doi: [http://dx.doi.org/10.1016/S0031-3203\(96\)00142-2](http://dx.doi.org/10.1016/S0031-3203(96)00142-2).
- L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Statistics/Probability Series. Wadsworth Publishing Company, Belmont, California, U.S.A., 1984.
- R. Caruana and D. Freitag. Greedy attribute selection. In *International Conference on Machine Learning*, pages 28–36, 1994.
- C. C. Chang and C. J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.

- O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.
- J. Chen, C. A. Tsai, S. L. Tzeng, and C. H. Chen. Gene selection with multiple ordering criteria. *BMC Bioinformatics*, 8(1):74–90, 2007. ISSN 1471-2105. doi: 10.1186/1471-2105-8-74.
- M. R. Chernick. *Bootstrap Methods: A Guide for Practitioners and Researchers*. John Wiley & Sons, 2007.
- J. H. Cho, D. Lee, J. H. Park, and I. B. Lee. Gene selection and classification from microarray data using kernel machine. *FEBS Letters*, 571:93–98, 2004.
- P. Clark and T. Niblett. The cn2 induction algorithm. *Mach. Learn.*, 3:261–283, March 1989. ISSN 0885-6125. doi: 10.1023/A:1022641700528.
- W. W. Cohen. Fast effective rule induction. In *In Proceedings of the Twelfth International Conference on Machine Learning*, pages 115–123, 1995.
- S. D. Conzone and C. G. Pantano. Glass slides to dna microarrays. *Materials Today*, 7(3):20 – 26, 2004. ISSN 1369-7021. doi: DOI:10.1016/S1369-7021(04)00122-1.
- C. Cortes and V. Vapnik. Support-vector networks. *Mach. Learn.*, 20:273–297, September 1995. ISSN 0885-6125. doi: 10.1023/A:1022627411411.
- T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13:21–27, 1967.
- K. Crammer, R. Gilad-bachrach, A. Navot, and N. Tishby. Margin analysis of the lvq algorithm. In *Neural Information Processing Systems*, pages 462–469, 2002.

- M. Dash and H. Liu. Feature selection for classification. *Intelligent Data Analysis*, 1:131–156, 1997.
- C. Deisy, S. Baskar, N. Ramraj, J. Saravanan Koori, and P. Jeevanandam. A novel information theoretic-interact algorithm (it-in) for feature selection using three machine learning algorithms. *Expert Syst. Appl.*, 37:7589–7597, December 2010. ISSN 0957-4174.
- J. L. Derisi, V. R. Iyer, and P. O. Brown. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, 278:680–686, 1997.
- T. G. Dietterich. Ensemble methods in machine learning. In *Multiple Classifier Systems*, volume 1857 of *Lecture Notes in Computer Science*, pages 1–15. Springer Berlin / Heidelberg, 2000. 10.1007/3-540-45014-9-1.
- C. Ding and H. C. Peng. Minimum redundancy feature selection for gene expression data. In *IEEE Computer Society Bioinformatics Conference (CSB 2003)*, pages 523–529, Stanford, CA, August 2003.
- J. Doak. An evaluation of feature selection methods and their application to computer security. Technical report, CA: University of California, Department of Computer Science, 1992.
- P. Domingos. Rule induction and instance-based learning a unified approach. In *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 2*, pages 1226–1232, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1-55860-363-8.

- P. Domingos and M. J. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Mach. Learn.*, 29:103–130, November 1997. ISSN 0885-6125.
- P. Domingos and M. J. Pazzani. Beyond independence: Conditions for the optimality of the simple bayesian classifier. In *International Conference on Machine Learning*, pages 105–112, 1996.
- R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, 2 edition, November 2001. ISBN 0471056693.
- S. Dudoit, Ja. Fridlyand, and T. P. Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, 97(457):77–87, 2002.
- K. Dunne, P. Cunningham, and F. Azuaje. Solutions to instability problems with sequential Wrapper-Based approaches to feature selection. Technical Report TCD-CD-2002-28, Department of Computer Science, Trinity College, Dublin, Ireland, 2002.
- C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation revisited. Technical report, 2001.
- B. Efron. Estimating the error rate of a prediction rule: Improvement on cross-validation. *Journal of the American Statistical Association*, 78(382):316–331, 1983.
- I. Fodor. A survey of dimension reduction techniques. Technical report, 2002.

- Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55:119–139, August 1997.
- J. H. Friedman. Regularized discriminant analysis. *Journal of the American Statistical Association*, 84(405):165–175, Mar. 1989.
- T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Hausler. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10):906–914, 2000.
- C. Furlanello, M. Serafini, S. Merler, and G. Jurman. Semisupervised learning for molecular profiling. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2(2):110–118, 2005. doi: 10.1109/TCBB.2005.28.
- J. Fürnkranz. Round robin rule learning. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 146–153, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1-55860-778-1.
- J. Fürnkranz. Separate-and-conquer rule learning. *Artif. Intell. Rev.*, 13:3–54, February 1999. ISSN 0269-2821. doi: 10.1023/A:1006524209794.
- G. H. Golub and C. Van Loan. *Matrix Computations*. Baltimore, MD: Johns Hopkins University Press, 3rd ed. edition, 1996.
- T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537, October 1999.

- G. Gulgezen, Z. Cataltepe, and L. Yu. Stable and accurate feature selection. In Wray Buntine, Marko Grobelnik, Dunja Mladenic, and John Shawe-Taylor, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 5781 of *Lecture Notes in Computer Science*, pages 455–468. Springer Berlin / Heidelberg, 2009.
- I. Guyon. 2001. <http://www.clopinet.com/isabelle/Projects/NIPS2001/>. Retrieved in June, 2009.
- I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, March 2003. ISSN 1532-4435.
- I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.
- M. A. Hall. Correlation-based feature selection for discrete and numeric class machine learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 359–366, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.
- M. A. Hall. *Correlation-based Feature Selection for Machine Learning*. PhD thesis, University of Waikato, 1999.
- M. A. Hall and L. Smith. Practical feature subset selection for machine learning. In *Proc 21st Australian Computer Science Conference*, pages 181–191, Perth, Australia, 1998. Springer.
- P. Hall, J. S. Marron, and A. Neeman. Geometric representation of high dimension, low sample size data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(3):427–444, June 2005.

- D. A. Harville. *Matrix Algebra From a Statistician's Perspective*. Springer, New York, June 2008.
- T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, second edition, Feb 2009.
- D. W. Hosmer and S. Lemeshow. *Applied logistic regression (Wiley Series in probability and statistics)*. Wiley-Interscience Publication, 2 edition, 2000.
- C. Howson and P. Urbach. *Scientific Reasoning: The Bayesian Approach*. Open Court Publishing Company, December 1993.
- D. F. Hsu and I. Taksa. Comparing rank and score combination methods for data fusion in information retrieval. *Inf. Retr.*, 8(3):449–480, May 2005. ISSN 1386-4564.
- E. B. Huertaa, B. Duvalb, and J. K. Hao. A hybrid lda and genetic algorithm for gene selection and classification of microarray data. *Neurocomputing*, 73(13-15): 2375–2383, Aug 2010.
- E. B. Hunt, J. Marin, and P. J. Stone. *Experiments in induction*. Academic Press, New York, NY, 1966.
- I. Inza, B. Sierra, R. Blanco, and P. Larranaga. Gene selection by sequential search wrapper approaches in microarray cancer class prediction. *J. Intell. Fuzzy Syst.*, 12:25–33, January 2002. ISSN 1064-1246.
- J. Jaeger, R. Sengupta, and W. L. Ruzzo. Improved gene selection for classification of microarrays. *Pac Symp Biocomput*, pages 53–64, 2003.

- G. H. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *International Conference on Machine Learning*, pages 121–129. AAAI Press, 1994.
- A. Kalousis, J. Prados, and M. Hilario. Stability of feature selection algorithms: a study on high-dimensional spaces. *Knowl. Inf. Syst.*, 12(1):95–116, May 2007. ISSN 0219-1377. doi: 10.1007/s10115-006-0040-8.
- E. Kim and J. Ko. Dynamic classifier integration method. In Nikunj C. Oza, Robi Polikar, Josef Kittler, and Fabio Roli, editors, *Multiple Classifier Systems*, volume 3541 of *Lecture Notes in Computer Science*, pages 916–917. Springer Berlin / Heidelberg, 2005. 10.1007/11494683-10.
- K. Kira and L. A. Rendell. The feature selection problem: Traditional methods and a new algorithm. In *AAAI*, pages 129–134, Cambridge, MA, USA, 1992. AAAI Press and MIT Press.
- T. A. Knijnenburg, M. J. T. Reinders, and L. F. A. Wessels. The selection of relevant and non-redundant features to improve classification performance of microarray gene expression data. In *11th Annual conference of the advanced school for computing and imaging*, 2005.
- R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 2*, pages 1137–1143, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

- R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.
- D. Koller and Me. Sahami. Toward optimal feature selection. In *International Conference on Machine Learning*, pages 284–292, 1996.
- I. Kononenko. Estimating attributes: Analysis and extensions of relief. In *ECML*, pages 171–182, 1994.
- L. I. Kuncheva. A stability index for feature selection. In *Proceedings of the 25th conference on Proceedings of the 25th IASTED International Multi-Conference: artificial intelligence and applications*, pages 390–395, Anaheim, CA, USA, 2007. ACTA Press.
- P. Křížek, J. Kittler, and V. Hlaváč. Improving stability of feature selection methods. In *Proceedings of the 12th international conference on Computer analysis of images and patterns*, CAIP’07, pages 929–936, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 978-3-540-74271-5.
- P. Langley and S. Sage. Oblivious decision trees and abstract cases. In D. W. Aha, editor, *Case-Based Reasoning: Papers from the 1994 Workshop (Technical Report WS-94-01)*, Menlo Park, CA, 1994a. AAAI Press.
- P. Langley and S. Sage. Induction of selective Bayesian classifiers. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pages 399–406, Seattle, WA, 1994b. Morgan Kaufmann.
- O. Lézoray and H. Cardot. Comparing combination rules of pairwise neural networks

- classifiers. *Neural Process. Lett.*, 27(1):43–56, February 2008. ISSN 1370-4621. doi: 10.1007/s11063-007-9058-5.
- L. P. Li, C. R. Weinberg, T. A. Darden, and L. G. Pedersen. Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the ga/knn method. *Bioinformatics*, 17(12):1131–1142, 2001.
- Y. M. Li and X. P. Zeng. Sequential multi-criteria feature selection algorithm based on agent genetic algorithm. *Applied Intelligence*, 33(2):117–131, October 2010. ISSN 0924-669X. doi: <http://dx.doi.org/10.1007/s10489-008-0153-8>.
- C. Ling, J. Huang, and H. Zhang. Auc: A better measure than accuracy in comparing learning algorithms. In Yang Xiang and Brahim Chaib-draa, editors, *Advances in Artificial Intelligence*, volume 2671 of *Lecture Notes in Computer Science*, pages 991–991. Springer Berlin / Heidelberg, 2003.
- H. Liu and H. Motoda. *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer Academic Publishers, Norwell, MA, USA, 1998. ISBN 079238198X.
- H. Liu, H. Motoda, and M. Dash. A monotonic measure for optimal feature selection. In *Proceedings of the 10th European Conference on Machine Learning*, pages 101–106, London, UK, 1998. Springer-Verlag. ISBN 3-540-64417-2.
- H. Liu, H. Motoda, and L. Yu. Feature selection with selective sampling. In *Proceedings of the Nineteenth International Conference on Machine Learning, ICML '02*, pages 395–402, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

- H. Liu, H. Motoda, and L. Yu. A selective sampling approach to active feature selection. *Artif. Intell.*, 159:49–74, November 2004. ISSN 0004-3702.
- D. J. Lockhart, H. Dong, M. C. Byrne, M. T. Follettie, M. V. Gallo, M. S. Chee, M. Mittmann, C. Wang, M. Kobayashi, H. Horton, and E. L. Brown. Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nature Biotechnology*, 14(13):1675–1680, December 1996. ISSN 1087-0156.
- S. Loscalzo, L. Yu, and C. Ding. Consensus group stable feature selection. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 567–576, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-495-9. doi: <http://doi.acm.org/10.1145/1557019.1557084>.
- K. Z. Mao and W. Y. Tang. Recursive mahalanobis separability measure for gene subset selection. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 8(1):266–272, 2011. ISSN 1545-5963. doi: <http://doi.ieeecomputersociety.org/10.1109/TCBB.2010.43>.
- R. S. Michalski and R. L. Chilausky. Learning by being told and learning from examples: An experimental comparison of the two methods of knowledge acquisition in the context of developing an expert system for soybean disease diagnosis. *International Journal of Policy Analysis and Information Systems*, 4:2, 1980.
- A. W. Moore and M. S. Lee. Efficient algorithms for minimizing cross validation error. In *Proceedings of the 11th International Conference on Machine Learning*, pages 190–198. Morgan Kaufmann, 1994.

- P. A. Mundra and J. C. Rajapakse. Svm-rfe with mrmr filter for gene selection. *IEEE Trans Nanobioscience*, 9(1):31–37, Mar 2010. doi: 10.1109/TNB.2009.2035284.
- J. Neumann, C. Schnrr, and G. Steidl. Svm-based feature selection by direct objective minimisation. In *Schlkopf (eds.): Pattern Recognition, Proc. of 26th DAGM Symposium, Vol. 3175 of LNCS*, pages 212–219. Springer, 2004.
- C. H. Ooi and P. Tan. Genetic algorithms applied to multi-class prediction for the analysis of gene expression data. *Bioinformatics*, 19(1):37–44, 2003.
- C. H. Ooi, M. Chetty, and S. W. Teng. Differential prioritization between relevance and redundancy in correlation-based feature selection techniques for multiclass gene expression data. *BMC Bioinformatics*, 7:320, 2006. doi: 10.1186/1471-2105-7-320.
- M. Osl, S. Dreiseitl, F. Cerqueira, M. Netzer, B. Pfeifer, and C. Baumgartner. Demoting redundant features to improve the discriminatory ability in cancer data. *J Biomed Inform*, 42(4):721–725, Aug 2009. doi: 10.1016/j.jbi.2009.05.006.
- K. K. Paliwala and A. Sharma. Improved direct lda and its application to dna microarray gene expression data. *Pattern Recognition Letters*, 31(16):2489–2492, Dec 2010.
- H. H. Peng, F. H. Long, and C. Ding. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, Aug 2005. ISSN 0162-8828. doi: <http://doi.ieeecomputersociety.org/10.1109/TPAMI.2005.159>.

- S. L. Pomeroy, P. Tamayo, M. Gaasenbeek, L. M. Sturla, M. Angelo, M. E. McLaughlin, J. Y. Kim, L. C. Goumnerova, P. M. Black, C. Lau, J. C. Allen, D. Zagzag, J. M. Olson, T. Curran, C. Wetmore, J. A. Biegel, T. Poggio, S. Mukherjee, R. Rifkin, A. Califano, G. Stolovitzky, D. N. Louis, J. P. Mesirov, E. S. Lander, and T. R. Golub. Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*, 415(6870):436–442, January 2002. ISSN 0028-0836. doi: 10.1038/415436a.
- J. R. Quinlan. Induction of decision trees. *Mach. Learn.*, 1:81–106, March 1986. ISSN 0885-6125. doi: 10.1023/A:1022643204877.
- J. R. Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993. ISBN 1-55860-238-0.
- J. Rachlin, S. Kasif, S. Salzberg, and D. W. Aha. Towards a better understanding of memory-based reasoning systems. In *International Conference on Machine Learning*, pages 242–250, 1994.
- A. Rakotomamonjy. Variable selection using svm-based criteria. *Journal of Machine Learning Research*, 3:1357–1370, 2003.
- J. Reunanen. Overfitting in making comparisons between variable selection methods. *J. Mach. Learn. Res.*, 3:1371–1382, March 2003.
- I. Rish. An empirical study of the naive bayes classifier. In *IJCAI-01 workshop on "Empirical Methods in AI"*, 2001.
- M. Robnik-Sikonja and I. Kononenko. An adaptation of relief for attribute estimation in regression. In *Proceedings of the Fourteenth International Conference*

- on Machine Learning*, ICML '97, pages 296–304, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc. ISBN 1-55860-486-3.
- M. Robnik-Sikonja and I. Kononenko. Theoretical and empirical analysis of relief and rrelieff. *Machine Learning*, 53(1-2):23–69, 2003.
- R. Rosipal and N. Krämer. Overview and recent advances in partial least squares. In Craig Saunders, Marko Grobelnik, Steve Gunn, and John Shawe-Taylor, editors, *Subspace, Latent Structure and Feature Selection*, volume 3940 of *Lecture Notes in Computer Science*, chapter 2, pages 34–51. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2006.
- R. Ruiz, B. Pontes, R. Girddez, and J. AguilarRuiz. Gene ranking from microarray data for cancer classificationa machine learning approach. In Bogdan Gabrys, Robert Howlett, and Lakhmi Jain, editors, *Knowledge-Based Intelligent Information and Engineering Systems*, volume 4252 of *Lecture Notes in Computer Science*, pages 1272–1280. Springer Berlin / Heidelberg, 2006.
- Y. Saeys, I. Inza, and P. Larrañaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, September 2007. ISSN 1367-4803. doi: 10.1093/bioinformatics/btm344.
- Y. Saeys, T. Abeel, and Y. Peer. Robust feature selection using ensemble feature selection techniques. In *Proceedings of the European conference on Machine Learning and Knowledge Discovery in Databases - Part II*, ECML PKDD '08, pages 313–325, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 978-3-540-87480-5. doi: http://dx.doi.org/10.1007/978-3-540-87481-2_21.

- M. P. Sapat, A. C. Bovik, J. K. Aggarwal, and K. R. Castleman. Supervised parametric and non-parametric classification of chromosome images. *Pattern Recogn.*, 38:1209–1223, August 2005. ISSN 0031-3203.
- M. A. Shipp, K. N. Ross, P. Tamayo, A. P. Weng, J. L. Kutok, R. C. T. Aguiar, M. Gaasenbeek, M. Angelo, M. Reich, G. S. Pinkus, T. S. Ray, M. A. Koval, K. W. Last, A. Norton, T. A. Lister, J. Mesirov, D. S. Neuberg, E. S. Lander, J. C. Aster, and T. R. Golub. Diffuse large b-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature Medicine*, 8(1):68–74, 2002. doi: 10.1038/nm0102-68.
- D. Singh, G. P. Febbo, K. Ross, D. G. Jackson, J. Manola, C. Ladd, P. Tamayo, A. A. Renshaw, V. Anthony, D’Amico, J. P. Richie, E. S. Lander, M. Loda, P. W. Kantoff, T. R. Golub, and W. R. Sellers. Gene expression correlates of clinical prostate cancer behavior. *Nature*, 1(2):203–209, 2002.
- P. Somol and J. Novoviov. Evaluating the stability of feature selectors that optimize feature subset cardinality. In Niels da Vitoria Lobo, Takis Kasparis, Fabio Roli, James Kwok, Michael Georgiopoulos, Georgios Anagnostopoulos, and Marco Loog, editors, *Structural, Syntactic, and Statistical Pattern Recognition*, volume 5342 of *Lecture Notes in Computer Science*, pages 956–966. Springer Berlin / Heidelberg, 2008.
- Y. Song, J. Huang, D. Zhou, H. Zha, and C. L. Giles. Iknn: Informative k-nearest neighbor pattern classification. In *Principles of Data Mining and Knowledge Discovery*, pages 248–264, 2007. doi: 10.1007/978-3-540-74976-9_25.

- D. V. Sridhar, E. B. Bartlett, and R. C. Seagrave. Information theoretic subset selection for neural network models. *Computers & Chemical Engineering*, 22(4-5):613 – 626, 1998. ISSN 0098-1354. doi: DOI:10.1016/S0098-1354(97)00227-5.
- Y. Tang, Y. Q. Tang, and Z. Huang;. Development of two-stage svm-rfe gene selection strategy for microarray expression data analysis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 4(3):365 – 381, 2007.
- S. Theodoridis and K. Koutroumbas. *Pattern Recognition, Third Edition*. Academic Press, Inc., Orlando, FL, USA, 2006. ISBN 0123695317.
- C. Thomaz, D. Gillies, and R. Feitosa. Using mixture covariance matrices to improve face and facial expression recognitions. In Josef Bigun and Fabrizio Smeraldi, editors, *Audio- and Video-Based Biometric Person Authentication*, volume 2091 of *Lecture Notes in Computer Science*, pages 71–77. Springer Berlin / Heidelberg, 2001.
- K. Tumer and J. Ghosh. Theoretical foundations of linear and order statistics combiners for neural pattern classifiers. Technical report, IEEE Transactions on neural networks, 1996.
- P. Turney. Technical note: Bias and the quantification of stability. *Mach. Learn.*, 20:23–33, July 1995. ISSN 0885-6125.
- M. van Erp and L. Schomaker. Variants of the borda count method for combining ranked classifier hypotheses. In *Seventh International Workshop on Frontiers in Handwriting Recognition*, pages 443–452, 2000.
- V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

- V. N. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, New York, 1998.
- J. Wang, K. A. Do, S. Wen, S. Tsavachidis, T. J. McDonnell, C. J. Logothetis, and K. R. Coombes. Merging microarray data, robust feature selection, and predicting prognosis in prostate cancer. *Cancer Informatics*, 2:87–97, January 2006.
- L. L. Wang and J. W. Ma. A post-filtering gene selection algorithm based on redundancy and multi-gene analysis. *International Journal of Information Technology*, 11(8):36–44, 2005.
- A. R. Webb. *Statistical Pattern Recognition, 2nd Edition*. John Wiley & Sons, October 2002.
- J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, and V. Vapnik. Feature selection for SVMs. In *Advances in Neural Information Processing Systems 13*, volume 13, pages 668–674, 2000.
- J. Weston, A. Elisseeff, B. Schölkopf, and M. Tipping. Use of the zero norm with linear models and kernel methods. *J. Mach. Learn. Res.*, 3:1439–1461, March 2003. ISSN 1532-4435.
- M. C. Wu, L. S. Zhang Z. X. Wang, D. C. Christiani, and X. H. Lin. Sparse linear discriminant analysis for simultaneous testing for the significance of a gene set/pathway and gene selection. *Bioinformatics*, 25(9):1145–1151, Jan 2009. doi: 10.1093/bioinformatics/btp019.
- M. Xiong, X. Fang, and J. Zhao. Biomarker identification by feature wrappers. *Genome Research*, 11(11):1878–1887, 2001.

- W. Z. Yan. Fusion in multi-criterion feature ranking. In *2007 10th International Conference on Information Fusion*, pages 1–6, Quebec, Que., July 2007.
- F. Yang and K. Z. Mao. Robust feature selection for microarray data based on multi-criterion fusion. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 8(4):1080–1092, 2011. ISSN 1545-5963. doi: 10.1109/TCBB.2010.103.
- J. Ye, R. Janardan, Q. Li, and H. Park. Feature reduction via generalized uncorrelated linear discriminant analysis. *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1312–1322, Oct 2006.
- L. Yu and H. Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *in ICML*, pages 856–863, 2003.
- L. Yu and H. Liu. Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research*, 5:1205–1224, December 2004. ISSN 1532-4435.
- L. Yu, C. Ding, and S. Loscalzo. Stable feature selection via dense feature groups. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 803–811, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-193-4. doi: <http://doi.acm.org/10.1145/1401890.1401986>.
- L. Zhou, L. Wang, and C. Shen. Feature selection with redundancy-constrained class separability. *Trans. Neur. Netw.*, 21:853–858, May 2010. ISSN 1045-9227. doi: <http://dx.doi.org/10.1109/TNN.2010.2044189>.

X. Zhou and K. Z. Mao. Ls bound based gene selection for dna microarray data.

Bioinformatics, 21(8):1559–1564, 2005.

X. Zhou and K. Z. Mao. The ties problem resulting from counting-based error

estimators and its impact on gene section algorithms. *Bioinformatics*, 22(20):

2507–2515, 2006.

V. Zuber and K. Strimmer. Gene ranking and biomarker discovery under correlation.

Bioinformatics, 25:2700–2707, October 2009.