Feature selection algorithms for very high dimensional data and mixed data

Tang, Wen Yin

2008

Tang, W. Y. (2008). Feature selection algorithms for very high dimensional data and mixed data. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/41404>

<https://doi.org/10.32657/10356/41404>

Feature Selection Algorithms for Very High Dimensional Data and Mixed Data

Tang Wenyin

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfilment of the requirement for the degree of
Doctorate of Philosophy

2008

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

31 Jan 2008
.....
Date

.....
Tang Wenyin

Acknowledgements

First and foremost I would like to thank my supervisor, Dr. Mao Kezhi, Associate Professor of School of Electrical and Electronic Engineering, Nanyang Technological University, who opened the door of scientific research for me. He gave me enormous guidance and help and offered me lots of chances and spaces to think independently. I appreciate the wisdom of his way of guidance, from which I would benefit in all my life.

I would like to thank Nanyang Technological University for the financial support during my Ph.D study. They also have kindly provided teaching assistantship positions and travel funds to attend overseas conference.

I thank Zhao Peng and Zhou Xin for sharing valuable experiences and providing helpful comments to me. Yang Xin, Yap and Mok deserve thanks for their technical assistance.

Special thanks must go to my parents, Liu Zhigang and Tang Meili, for their unconditional support and encouragement, which made me strong through my time in graduate school.

Table of contents

Acknowledgements	i
Summary	vi
List of Figures	ix
List of Tables	xiv
1 Introduction	1
1.1 Motivation	1
1.2 Objectives	3
1.3 Major contribution of the Thesis	4
1.4 Organization of the Thesis	6
2 Review of Pattern Classification Algorithms	8
2.1 An Overview of Pattern Recognition Systems	8
2.2 Classification Algorithms	10
2.2.1 Classifiers Based on the Concept of Similarity	11
2.2.2 Classifiers Based on Probabilistic Estimation	16

TABLE OF CONTENTS iii

2.2.3	Classifiers Based on Error Criteria	19
2.2.4	Classifiers Based on Decision Trees	29
2.3	Classification Accuracy Estimation	30
2.3.1	Error Estimation with A Single Validation Set	31
2.3.2	Validation Methods	33
3	Literature Review of Feature Selection	40
3.1	Why Feature Selection?	40
3.2	The Components of A Feature Selection Algorithm	42
3.2.1	Search Procedure	43
3.2.2	Evaluation Criterion	46
3.3	Existing Feature Selection Algorithms	47
3.3.1	Wrapper Methods	47
3.3.2	Filter Methods	50
3.3.3	Filter-Wrapper Hybrid Methods	59
3.4	Chapter Summary	61
4	Fast Feature selection Algorithm for High-Dimensional Data Based on Regularized Recursive Mahalanobis Measure	62
4.1	Introduction	62
4.2	Fast Feature Selection based on Regulated Recursive Mahalanobis Measure	64
4.2.1	Recursive Mahalanobis Measure	64
4.2.2	Forward Selection Based on Recursive Mahalanobis Measure .	69

TABLE OF CONTENTS

iv

4.2.3	Regularized Recursive Mahalanobis Measure	72
4.3	Experimental Studies	77
4.4	Chapter Summary	79
5	Fast Feature selection Algorithm for Very High Dimensional Data	
	Based on Regularized Correlation Criterion	85
5.1	Introduction	85
5.2	Regularized Correlation Measure for Feature Selection	87
5.2.1	A New Correlation-based Approach to Feature Evaluation . .	87
5.2.2	Regularized Correlation Measure	92
5.2.3	Generation of Z_m	96
5.2.4	Procedure of Regularized Correlation-based Sequential Feature Selection	99
5.3	Experimental studies	100
5.4	Chapter Summary	105
6	A Novel Feature Selection Algorithm For Mixed Data	110
6.1	Introduction	110
6.2	Error Probability Measure for Mixed Feature Evaluation	112
6.2.1	Error Probability Measure	112
6.2.2	Mixed Feature Space Decomposition	115
6.2.3	Error Probability Estimation	118
6.3	Search Algorithm for Mixed Data	119
6.4	Experimental Studies	122

TABLE OF CONTENTS	v
6.4.1 Artificial Data	122
6.4.2 Real-World Data	130
6.5 Chapter Summary	135
7 Conclusions and Future Work	149
7.1 Conclusions	149
7.2 Recommendations for Future Work	154
Author's Publications	158
Bibliography	160

Summary

Feature selection is an important issue in pattern recognition. The goal of feature selection algorithm is to identify a set of relevant features, based on which to construct a classifier for a pattern recognition problem. This thesis addresses the problem of feature selection for very high dimensional data and mixed data, which exist in many application domains of pattern recognition nowadays. The proposed feature selection algorithms aim to eliminate both irrelevant and redundant features while retaining major discriminating underlying data.

Firstly, a fast feature selection algorithm based on recursive Mahalanobis separability measure has been proposed for very high dimensional data. This algorithm combines the recursive Mahalanobis separability measure with a sequential forward selection algorithm. Instead of evaluating the Mahalanobis measure directly in high dimensional space, the recursive approach evaluates the measure through successive evaluations in 2-dimensional space, where a much lower computational complexity is needed in feature evaluation. In addition, the regularization technique has been introduced to recursive Mahalanobis measure to alleviate the overfitting problem caused by the relatively small sample size. Experimental studies on eight gene

expression problems show that the regularized recursive Mahalanobis separability measure substantially outperforms the non-regularized Mahalanobis measures and the benchmark recursive feature elimination (RFE) algorithm in seven out of the eight problems.

Secondly, a new set-based correlation criterion has also been developed for very high dimensional data through orthogonal decomposition. The new criterion is combined with a sequential forward selection algorithm. In each selection step, instead of orthogonalizing a candidate feature with each of the features in the feature subset selected in previous steps, the new approach orthogonalizes the candidate feature to the class label prediction built upon the previously selected feature subset. The merit of this approach is that it retains the redundancy handling capacity of orthogonal decomposition while requires less computation than conventional orthogonal decomposition. In this study, regularization technique is also introduced to feature selection and a regularized correlation criterion is derived to alleviate the overfitting problem caused by relatively small sample size in high dimensional feature space. Experimental studies on six gene expression problems show that the feature subsets selected by the regularized correlation criterion produce improved classification performance.

Thirdly, a feature selection algorithm has been proposed for mixed data with both continuous and nominal features. This method consists of a novel criterion based on error probability and a novel search algorithm called mixed forward search algorithm. For mixed data, the common practice is to transform the data into either continuous

TABLE OF CONTENTS

viii

data or nominal data through feature type transformation to fit the requirement of different learning algorithms. However, feature type transformation might cause loss of useful information. In order to make better use of different typed features, the proposed algorithm considers different feature types in both evaluation and search algorithm. The performance of the proposed method has been tested using both artificial mixed data sets and benchmark real-world mixed data sets. Experimental results show that our method provides improved performance compared with the commonly used feature selection algorithms for mixed data.

List of Figures

2.1	A Typical Pattern Recognition System. Black arrows show the “bottom up” data flow in pattern recognition system. Gray arrows show feedback from higher level to lower level, which may employed in some systems (Duda et al., 2001).	9
4.1	Computational complexity at different selection step	70
4.2	Comparison of RM and RRM with M and RFE in Leukaemia problem	80
4.3	Comparison of RM and RRM with M and RFE in Breast Cancer (ER) problem	80
4.4	Comparison of RM and RRM with M and RFE in Breast Cancer (LN) problem	81
4.5	Comparison of RM and RRM with M and RFE in Lung Cancer problem	81
4.6	Comparison of RM and RRM with M and RFE in CNS Tumor problem	82
4.7	Comparison of RM and RRM with M and RFE in Breast Cancer problem	82
4.8	Comparison of RM and RRM with M and RFE in Colon Cancer problem	83

LIST OF FIGURES

x

4.9	Comparison of RM and RRM with M and RFE in Prostate Cancer problem	83
5.1	Rotation angle versus gamma	96
5.2	Comparison of regularized correlation-based feature selection with different parameters in Leukaemia problem	102
5.3	Comparison of regularized correlation-based feature selection with different parameters in Breast Cancer (ER) problem	103
5.4	Comparison of regularized correlation-based feature selection with different parameters in Breast Cancer (LN) problem	103
5.5	Comparison of regularized correlation-based feature selection with different parameters in Lung Cancer problem	104
5.6	Comparison of regularized correlation-based feature selection with different parameters in CNS Tumor problem	104
5.7	Comparison in Breast Cancer problem	105
5.8	Comparison of regularized correlation (RC) measure and regularized recursive Mahalanobis (RRM) measure (proposed in Chapter 4) in Leukaemia problem	106
5.9	Comparison of regularized correlation (RC) measure and regularized recursive Mahalanobis (RRM) measure (proposed in Chapter 4) in Breast Cancer (ER) problem	106

LIST OF FIGURES

xi

5.10 Comparison of regularized correlation (RC) measure and regularized recursive Mahalanobis (RRM) measure (proposed in Chapter 4) in Breast Cancer (LN) problem	107
5.11 Comparison of regularized correlation (RC) measure and regularized recursive Mahalanobis (RRM) measure (proposed in Chapter 4) in Lung Cancer problem	108
5.12 Comparison of regularized correlation (RC) measure and regularized recursive Mahalanobis (RRM) measure (proposed in Chapter 4) in CNS Tumor problem	108
5.13 Comparison of regularized correlation (RC) measure and regularized recursive Mahalanobis (RRM) measure (proposed in Chapter 4) in Breast Cancer problem	109
6.1 Decomposition of the feature space with features Thal (nominal), Sex (nominal), Age (continuous) and ST-depression (continuous).	116
6.2 Block Diagram of Mixed Forward Selection (MFS).	120
6.3 The new criterion P_e^{KNN} , GMD, MD and UC in case 1, i.e. the mixed data set with relevant continuous and nominal features. (The vertical bars represents symmetric error bars of unit standard deviation for mean values.)	124

LIST OF FIGURES

xii

- 6.4 The new criterion P_e^{KNN} , GMD, MD and UC in case 2, i.e. the mixed data set with relevant nominal feature and irrelevant continuous feature. (The vertical bars represents symmetric error bars of unit standard deviation for mean values.) 126
- 6.5 The new criterion P_e^{KNN} , GMD, MD and UC in case 3, i.e. the mixed data set with relevant continuous feature and irrelevant nominal feature. (The vertical bars represents symmetric error bars of unit standard deviation for mean values.) 127
- 6.6 The Pearson's linear correlation between ranks assigned to the nominal features and the number of feature values vs. different ranges of numbers of feature values. 129
- 6.7 Learning curves for SFS[MD], SFS[UC] and MFS with Naive Bayes classifier (a) and k -nearest neighbor classifier (b) on dataset: cleveland. 137
- 6.8 Learning curves for SFS[MD], SFS[UC] and MFS with Naive Bayes classifier (a) and k -nearest neighbor classifier (b) on dataset: crx. . . 138
- 6.9 Learning curves for SFS[MD], SFS[UC] and MFS with Naive Bayes classifier (a) and k -nearest neighbor classifier (b) on dataset: bridges. 139
- 6.10 Learning curves for SFS[MD], SFS[UC] and MFS with Naive Bayes classifier (a) and k -nearest neighbor classifier (b) on dataset: hepatitis. 140
- 6.11 Learning curves for SFS[MD], SFS[UC] and MFS with Naive Bayes classifier (a) and k -nearest neighbor classifier (b) on dataset: horse-colic. 141

6.12 Learning curves for SFS[MD], SFS[UC] and MFS with Naive Bayes classifier (a) and k -nearest neighbor classifier (b) on dataset: long-beach-va.	142
6.13 Learning curves for RELIEF, CFS, SFS[GMD] and MFS with Naive Bayes classifier (a) and k -nearest neighbor classifier (b) on dataset: cleveland.	143
6.14 Learning curves for RELIEF, CFS, SFS[GMD] and MFS with Naive Bayes classifier (a) and k -nearest neighbor classifier (b) on dataset: crx.	144
6.15 Learning curves for RELIEF, CFS, SFS[GMD] and MFS with Naive Bayes classifier (a) and k -nearest neighbor classifier (b) on dataset: bridges.	145
6.16 Learning curves for RELIEF, CFS, SFS[GMD] and MFS with Naive Bayes classifier (a) and k -nearest neighbor classifier (b) on dataset: hepatitis.	146
6.17 Learning curves for RELIEF, CFS, SFS[GMD] and MFS with Naive Bayes classifier (a) and k -nearest neighbor classifier (b) on dataset: horse-colic.	147
6.18 Learning curves for RELIEF, CFS, SFS[GMD] and MFS with Naive Bayes classifier (a) and k -nearest neighbor classifier (b) on dataset: long-beach-va.	148

List of Tables

2.1	Characteristics of different classification algorithms	30
4.1	Datasets Description	78
5.1	Datasets Description	101
6.1	Variables and Their Types and Values	115
6.2	Distinct Values in Multi-Nominal Variable $\mathbf{z}$	115
6.3	Pseudocode of Search Algorithm: Mixed Forward Selection	121
6.4	Summary of datasets. All these datasets are obtained from UC Irvine Repository (Murphy and Aha, 1995). Datasets above the horizontal line are “balanced” over different classes and those below are “imbal- anced”.	131

Chapter 1

Introduction

1.1 Motivation

New application domains of pattern recognition bring new challenges. One challenge arises from the increasing dimension of data. In the past, the data dimensionality was considered to be large if the number of features was in the range of hundreds. Today, data collected from Internet, business and medical domains could contain thousands of features. Although more features imply greater discriminative power, pattern classification in a high-dimensional feature space might not necessarily perform well because the limited training samples are distributed sparsely in such a high-dimensional feature space. Such sparse feature space makes it very easy to find a decision function to perfectly classify the training samples. However, such a decision function might not generalize well to unseen data. In statistics, this phenomenon is commonly referred to as the “curse of dimensionality” (Bellman, 1961), which indicates the requirement of the exponential increase in training samples associated with the adding extra dimensions. With limited number of samples, the classification algorithms may break down in a high-dimensional feature space (Hastie

et al., 2001).

Usually, the curse of dimensionality is alleviated by reducing the number of features using feature selection algorithms. However, traditional feature selection algorithms may not be applicable to very high-dimensional data due to computational and statistical problems. Computationally, the complexity time is quadratic or higher about feature dimensionality. Statistically, to keep the accuracy of feature subset evaluation, the number of samples should be exponential with respect to feature dimensionality. For very high-dimensional data, feature selection algorithm that is able to select feature effectively and efficiently is worth to be developed.

Another challenging problem is mixed data problem. In real-world problems, the data collected are often mixed, with both continuous and nominal features. The distinction between these two feature types is that the continuous feature includes numeric values in a continuous range and there is always a certain distance relationship between any two numeric values; while the nominal feature includes several mutually exclusive categories and there is no definite distance or even order among these categories. For example, a feature COLOR has values of blue, yellow and green colors, but there is no certain distance or even order between blue, yellow and green colors.

Most existing feature selection algorithms are suitable for either continuous or nominal features, depending on the evaluation criteria used. For example, the distance criteria are suitable to the nature of continuous features while the information criteria are suitable to the nature of nominal features. These criteria cannot be

directly applied to mixed data. The common practice of dealing with mixed data is to transform continuous features into nominal ones or the inverse before feature selection. However, the type-transformation causes either loss of mutual information in nominal features or loss of order information in continuous features.

Classification accuracy can be applied to mixed data if the induced classifier is able to handle the mixed data, such as decision trees (Breiman et al., 1984; Quinlan, 1993). But these feature selection algorithms, called wrapper methods, are computationally expensive because they involve building a classifier hundreds of times to evaluate one feature subset, as described in Section 3.3.1.

Some criteria have been designed for mixed typed data in literature. For example, the hybrid distance criteria (Wilson and Martinez, 1997; de Leon and Carrere, 2003) combines two kinds of distance criteria, one for each feature type, in feature evaluation. This method avoids feature type transformation. However, the performance is not so satisfying. Feature selection algorithms for mixed data are worth to be explored.

1.2 Objectives

The objective of this thesis is to develop new, effective and efficient feature selection algorithms for very high dimensional data and mixed data. The proposed feature selection algorithms should improve the predictive accuracy, robustness and learning speed in pattern classification.

1.3 Major contribution of the Thesis

The follows are the main contributions of the thesis:

1. **Regularized recursive Mahalanobis measure for very high dimensional data**

Mahalanobis class separability measure provides an effective evaluation of the discriminative power of a feature subset, and is widely used in feature selection. However, this measure is computationally very intensive or even prohibitive when it is applied to very-high dimensional data set, such as gene expression data. In this thesis, a recursive approach to Mahalanobis measure evaluation is proposed, with the goal of reducing computational overhead. Instead of evaluating Mahalanobis measure directly in high dimensional space, the recursive approach evaluates the measure through successive evaluations in 2-dimensional space. Because of its recursive nature, this approach is extremely efficient when it is combined with a forward search procedure. In addition, it is noted that feature subsets selected by Mahalanobis measure tend to overfit training data and generalize unsatisfactorily on unseen test data, due to the relatively small sample size in high dimensional feature space. To alleviate the overfitting problem, a regularized recursive Mahalanobis measure is proposed in this study, and guidelines on determination of regularization parameters are provided.

2. **A regularized correlation criterion for efficient feature subset selec-**

tion

The correlation between an individual feature and class labels reflects the predictive power of the feature, and has been commonly used in feature selection for very high dimensional data because of its efficiency. However, the individual based feature selection can hardly identify redundant features. In order to alleviating redundancy of data, we proposed a new set-based correlation criterion for feature subset evaluation through orthogonal decomposition. This new criterion is combined with sequential forward selection. In each step, instead of orthogonalizing a candidate feature with each of the features in the feature subset selected in previous steps, the new approach orthogonalizes the candidate feature to the class label prediction built upon the feature subset. This approach is able to detect the redundancy while requiring less computation than conventional orthogonal decomposition. The regularization technique is also introduced to feature selection and a regularized correlation criterion is derived to alleviate the overfitting problem.

3. A new feature selection algorithm for mixed data

For mixed data analysis, the common practice is to transform the data into either continuous data or nominal data through feature type transformation according to the requirement of a particular evaluation criterion employed. This might either lose the order information in continuous features or the mutual information in nominal feature. In order to make better use of different typed features, an error probability-based criterion is developed for mixed

feature subset evaluation. This new criterion is combined with mixed forward search procedure and evaluates a mixed feature subset through integrating the error probabilities estimated in a series of continuous feature subspaces. These feature subspaces are obtained by space decomposition based on nominal features in the subset. The merits of this new criterion is that it can detect the association between features of different types.

In addition to the mixed evaluation criterion, a mixed forward search procedure is developed specially for mixed data, regarding different feature types in subset generation, evaluation and comparison. In mixed forward selection procedure, three different criteria are employed for continuous, nominal and mixed feature subsets respectively, but the direct comparison of different criteria is avoided. This new search procedure could improve the reliability of feature evaluation.

1.4 Organization of the Thesis

The rest of this thesis is organized as follows:

Chapter 2 briefly reviews the various classification algorithms and classification accuracy estimation methods. Chapter 3 is the literature review for feature selection. Components of a typical feature selection algorithm are described at first, including various evaluation criteria, search procedures and stopping criteria. Then, the existing feature selection algorithms are reviewed in three categories: filter method, wrapper method and filter-wrapper hybrid method. In Chapter 4 and Chapter 5,

two efficient feature selection algorithms are developed for very high dimensional data. Chapter 4 proposes a fast feature selection algorithm based on regularized recursive Mahalanobis separability measure. The recursive Mahalanobis separability measure, which is based on sequential forward selection, is first introduced. To alleviate the overfitting problem, a regularized recursive Mahalanobis distance measure is further developed. Two new proposed measures are compared and discussed in experimental study. Chapter 5 develops an efficient correlation-based criterion for feature subset selection. Regularized correlation based criterion is then proposed to alleviate the overfitting problem. The new criteria with and without regularization are compared in experimental study. In Chapter 6, a new feature selection algorithm is developed for mixed data. This includes a new feature evaluation criterion and a new search algorithm. In experimental studies, the performance of the proposed algorithm are tested by using both artificial and real world mixed data sets. Comparison is also performed with other bench mark methods. Conclusions of the thesis and suggestions for future work are presented in Chapter 7.

Chapter 2

Review of Pattern Classification

Algorithms

This section reviews methodologies of supervised learning in pattern recognition used in the thesis. An overview of pattern recognition systems is given in Section 2.1. Various classification algorithms and accuracy estimation methods are reviewed in Section 2.2 and Section 2.3, respectively.

2.1 An Overview of Pattern Recognition Systems

Pattern recognition, a sub-topic of machine learning, can be defined as *“the act of taking in raw data and taking an action based on the category of the data”* (Duda et al., 2001). As shown in Fig. 2.1, a typical pattern recognition system consists of sensors to collect the observations or signals; a feature extraction mechanism to compute numeric or symbolic information from the observations or signals; a feature selection mechanism to select a feature subset with a larger separability of classes; a classification to do the actual job of classifying or describing observations, relying on the selected features; and a post-processing to evaluate and optimize the

Figure 2.1: A Typical Pattern Recognition System. Black arrows show the “bottom up” data flow in pattern recognition system. Gray arrows show feedback from higher level to lower level, which may employed in some systems (Duda et al., 2001).

classification accuracy.

Two learning mechanisms, supervised learning and unsupervised learning, are usually used in pattern recognition. In supervised learning, the classification is usually based on a set of training samples, each of which is represented by a set of

features and a class label. Learning can also be unsupervised, in the case that the system is not given an a priori class labelling of patterns. The idea of unsupervised learning is to establish the classes/clusters based on the statistical regularities of the patterns instead. This thesis will focus on the supervised learning methods.

Typical applications of pattern recognition are automatic speech recognition, text classification (e.g. classify spam/non-spam email messages), the automatic recognition of handwritten (e.g. true/fake signatures), the fingerprint recognition, or the human face recognition. With the advance in information technology nowadays, a very huge amount of data could be collected from Internet, scientific and government domains and stored in huge databases. New application fields of pattern recognition have been expanded to automatic disease diagnose, cancer classification, gene microarray data analysis and business intelligence, etc.

2.2 Classification Algorithms

When confronting a pattern recognition problem, a suitable classifier is selected based on the characteristics of the problem, such as a priori information, data distribution, amount of training samples, feature types, etc. Various classifiers can be put in four categories:

- Classifiers based on the concept of similarity, e.g. the minimum mean distance classifier, nearest neighbor (Duda and Hart, 1973) and k-nearest neighbor classifier (Duda and Hart, 1973).
- Classifiers based on probabilistic estimation, e.g. Bayesian decision rule, Naive

Bayes classifier (Domingos and Pazzani, 1997), logistic regression classifier (Hosmer and Stanley, 1975).

- Classifiers to construct the decision boundary by optimizing some error criterion, e.g. least squares, linear/quadratic discriminant analysis, artificial neural networks (Haykin, 1999) and support vector machine (Vapnik, 1998).
- Classifiers based on decision trees, e.g. CART (Breiman et al., 1984), ID3 (Quinlan, 1986), C4.5 (Quinlan, 1993);

2.2.1 Classifiers Based on the Concept of Similarity

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

Minimum Mean Distance Classifier

Minimum mean distance classifier supposes that each class is represented by its mean vector μ_l ($l = 1, \dots, L$), estimated using the sample mean:

$$\hat{\mu}_l = \frac{1}{N_l} \sum_{\mathbf{x}_i \in \text{class } l} \mathbf{x}_i \quad (2.1)$$

where N_l is the number of samples in the l th class. Let $\mathbf{x}$ be an unknown input, a minimum mean distance classifier computes the distance between $\mathbf{x}$ and $\hat{\mu}_l$ for $l = 1, \dots, L$ and chooses the class for which this distance is minimum, as defined by

$$L(\mathbf{x}) = \arg \min_l \text{dist}(\mathbf{x}, \hat{\boldsymbol{\mu}}_l) \quad (2.2)$$

Here $L(\cdot)$ denoted as predict label of input vector and $\text{dist}(\mathbf{x}, \mathbf{z})$ defined as the distant between two vector $\mathbf{x}$ and $\mathbf{z}$. There is more than one way to define the distance metrics, corresponding to different ways of distance measuring. Three of the most common used metrics are Euclidean metric, Manhattan metric and Mahalanobis metric, as defined below.

Euclidean metric: $\text{dist}(\mathbf{x}, \mathbf{z}) = \sqrt{(x_1 - z_1)^2 + (x_2 - z_2)^2 + \cdots + (x_d - z_d)^2}$

Manhattan metric: $\text{dist}(\mathbf{x}, \mathbf{z}) = |x_1 - z_1| + |x_2 - z_2| + \cdots + |x_d - z_d|$

Mahalanobis metric: $\text{dist}(\mathbf{x}, \mathbf{z}) = \sqrt{(\mathbf{x} - \mathbf{z})^T \Sigma^{-1} (\mathbf{x} - \mathbf{z})}$

where Σ is the covariance matrix with $d \times d$ dimensions and each element $\Sigma(j, k)$ represents the covariance of features j and feature k , for $j, k = 1, 2, \dots, d$. Minimum mean distance is simple and robust for many situations, however, it is unsatisfactory for classifying some complex situations, such as curved classification boundary, severe overlapping between different classes and highly correlated features.

Nearest Neighbor Classifier

Nearest neighbor classifier is another classifier based on distance between pairs of samples. Compared with minimum mean distance classifier, nearest neighbor classifier is even simpler because 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.

$$L(\mathbf{x}) = L \left[\arg \min_{\mathbf{x}_i} \text{dist}(\mathbf{x}, \mathbf{x}_i) \right] \quad (2.3)$$

where $L(\cdot)$ is denoted as the class label or predict label of input vector and $\text{dist}(\mathbf{x}, \mathbf{z})$ is denoted as the distance between two samples $\mathbf{x}$ and $\mathbf{z}$. There is more than one way to define the distance metrics, based on what the natural distance is in the corresponding space. The most commonly used is Euclidean distance as given before.

Nearest neighbor classifier is also called instance-based learner (Aha et al., 1991; Aha, 1992; Domingos, 1995) or memory-based reasoning method (Stanfill and Waltz, 1986; Rachlin et al., 1994), for classifying unseen data based on samples (instances) that are stored in memory. Further more, since classification of 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 classifier (Cover and Hart, 1967; Hart, 1968) 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 $\mathcal{N}_{\mathbf{x}} = \{(\mathbf{x}_i, y_i)\}$ for $i = 1, \dots, k$. The parameter k is generally predefined as $k = 3, 5, \dots, 21$. Then, the input sample $\mathbf{x}$ is labelled by the majority class in the neighborhood $\mathcal{N}_{\mathbf{x}}$.

$$L(\mathbf{x}) = \arg \max_l \frac{k_l}{k} \quad (2.4)$$

where k_l is number of samples that belong to class l 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.

K -nearest neighbor classifier can also be applied to datasets with nominal features by employing suitable distance metrics, such as overlapping metric (Aha et al., 1991; Aha, 1992) and value difference metric (VDM) (Stanfill and Waltz, 1986) as defined below.

$$\begin{aligned} \text{Overlapping metric:} \quad dist(\mathbf{x}, \mathbf{z}) &= \sum_{j=1}^d overlap(x_j, z_j) \\ overlap(x_j, z_j) &= \begin{cases} 0, & \text{if } x_i = z_i \\ 1, & \text{otherwise} \end{cases} \end{aligned}$$

$$\begin{aligned} \text{Value difference metric:} \quad dist(\mathbf{x}, \mathbf{z}) &= \sum_{j=1}^d vdm(x_j, z_j) \\ vdm(x_j, z_j) &= \sum_{l=1}^L \left| \frac{N_{x_j, l}}{N_{x_j}} - \frac{N_{z_j, l}}{N_{z_j}} \right|^q \end{aligned}$$

where

- $N_{x_j, l}$ is number of training samples have value x_j and output class l ;

- N_{x_j} is number of training samples have value x_j ;
- q is a constant equal to 1 or 2;

To handle mixed datasets with both continuous and nominal features, heterogeneous distance functions was proposed by employing different distance metrics for different types of features (Aha et al., 1991; Aha, 1992; Wilson and Martinez, 1997). One combination way, called HEOM (heterogeneous Euclidean-overlapping metric) (Aha et al., 1991; Aha, 1992), is to use overlapping metric for nominal features and normalized Euclidean distance for continuous features.

$$HOEM(\mathbf{x}, \mathbf{z}) = \sqrt{\sum_{j=1}^d \delta^2(x_j, z_j)} \quad (2.5)$$

where

$$\delta(x_j, z_j) = \begin{cases} 1, & \text{if } x_j \text{ or } z_j \text{ is unknown, else} \\ \text{overlap}(x_j, z_j), & \text{if the } j^{th} \text{ feature is nominal} \\ x_j - z_j, & \text{otherwise} \end{cases}$$

Wilson and Martinez (1997) proposed another combination way called HVDM (heterogeneous value difference metric), using VDM for nominal feature and Euclidean distance metric for continuous features.

$$HVDM(\mathbf{x}, \mathbf{z}) = \sqrt{\sum_{j=1}^d \delta^2(x_j, z_j)} \quad (2.6)$$

where

$$\delta(x_j, z_j) = \begin{cases} 1, & \text{if } x_j \text{ or } z_j \text{ is unknown, else} \\ vdm(x_j, z_j), & \text{if the } j^{th} \text{ feature is nominal} \\ x_j - z_j, & \text{otherwise} \end{cases}$$

2.2.2 Classifiers Based on Probabilistic Estimation

The second category of classifiers is designed by estimating the probability distributions of different classes in some given feature space. Any unknown data that is located somewhere in the same feature space would be classified to the class with the maximum posterior probability.

Bayesian Decision Rule

Let $l = 1, 2, \dots, L$ be the set of L possible classes, $p(\mathbf{x}|l)$ be the conditional probability density function for $\mathbf{x}$, $p(l)$ be the priori probability of training samples in class l . The posterior probability $P(l|\mathbf{x})$ can be computed from $p(\mathbf{x}|l)$ according to Bayesian formula (Duda and Hart, 1973):

$$P(l|\mathbf{x}) = \frac{p(\mathbf{x}|l)p(l)}{p(\mathbf{x})} \quad (2.7)$$

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

$$p(\mathbf{x}) = \sum_{l=1}^L p(\mathbf{x}|l)p(l) \quad (2.8)$$

and the Bayesian decision rule is given as:

$$\text{Decide class } l : \text{ if } P(l|\mathbf{x}) > P(l'|\mathbf{x}), \text{ for all } l \neq l' \quad (2.9)$$

If the conditional probability density is known, the Bayes decision rule yields an optimal classifier with minimum risk of misclassification (Duda et al., 2001), in the case of the same loss function. The probability density estimation can be either parametric (e.g. Gaussian distributions for continuous features, multinomial 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 $p(\mathbf{x}|l)$ is difficult.

Naive Bayes Classifier

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

$$p(\mathbf{x}|l) = \prod_{j=1}^d p(x_j|l) \quad (2.10)$$

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 (Dougherty et al., 1995; Domingos and Pazzani, 1996; Rish, 2001), even there are

clear dependencies between features. Domingos and Pazzani (1996) reported that the key to these results lies in the distinction between classification and probability estimation. 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). Further more, NBC can handle the mixed data with both continuous and nominal features naturally by choosing different probabilistic estimators for different types of features separately.

Logistic Regression Classifier

Logistic regression (LR) classifier, which is based on the maximum likelihood approach, attempts to model the posterior probabilities of L classes by linear functions using a set of training samples $\{(\mathbf{x}_i, y_i)\}$, where $y_i \in \{1, 2, \dots, L\}$ for $i = 1, 2, \dots, N$. For a two-class problem, the classifier finds the parameter $\hat{\beta}$ by the maximum likelihood as below:

$$\arg \max_{\beta} \prod_{\mathbf{x}_i \in \text{class1}} p_1(\mathbf{x}_i; \beta) \prod_{\mathbf{x}_i \in \text{class2}} p_2(\mathbf{x}_i; \beta) \quad (2.11)$$

where $p_l(\mathbf{x}_i; \beta)$ is the posterior probability of class l ($l = 1, 2$) given the i th training sample and parameters β . Given linear discriminant functions $f(\mathbf{x}_i; \beta) = \beta^T \mathbf{x}_i$, for $i = 1, 2, \dots, N$, the posterior probabilities of class 1 and class 2 for the sample $\mathbf{x}_i$ can be approximated by logistic functions:

$$\begin{aligned}
 p_1(\mathbf{x}_i; \boldsymbol{\beta}) &= \frac{e^{-f(\mathbf{x}_i; \boldsymbol{\beta})}}{1 + e^{-f(\mathbf{x}_i; \boldsymbol{\beta})}} \\
 p_2(\mathbf{x}_i; \boldsymbol{\beta}) &= \frac{1}{1 + e^{-f(\mathbf{x}_i; \boldsymbol{\beta})}}
 \end{aligned} \tag{2.12}$$

Eq. (2.11) can be optimized by Newton-Raphson algorithm, which is an iterative procedure for optimization (see Hastie et al., 2001, chapter 4.4). After finding the parameter $\hat{\boldsymbol{\beta}}$, the class label of any input unknown sample $\mathbf{x}$ is determined by

$$L(\mathbf{x}) = \arg \max_l p_l(\mathbf{x}; \hat{\boldsymbol{\beta}}) \tag{2.13}$$

Compared with discriminant analysis that assumes multivariate normal distribution of data, the logistic regression classifier makes no distributional assumption. For mixed data with both continuous and nominal features, where the multivariate normal assumption is clearly violated, logistical regression classifier could be a good choice (Jain et al., 2000).

2.2.3 Classifiers Based on Error Criteria

This category of classifiers construct decision functions by minimizing classification error, based on different error criteria defined.

Least Squares

Least squares (LS) is one of the most popular methods in pattern classification due to its simplicity and robustness (Hastie et al., 2001). Suppose that a linear model of form

$$\hat{y} = \hat{\beta}_0 + \sum_{j=1}^d \hat{\beta}_j x_j = \hat{\boldsymbol{\beta}}^T \mathbf{x} \quad (2.14)$$

is to be fit to a set of training data $\{(\mathbf{x}_i, y_i)\}$ for $i = 1, \dots, N$, each of which consists d features and class label y_i , where $\boldsymbol{\beta} = [\beta_0, \beta_1, \dots, \beta_d]^T$. The least squares find parameters $\hat{\boldsymbol{\beta}}$ to minimize residual sum of squares (RSS) defined as:

$$RSS(\boldsymbol{\beta}) = \sum_{i=1}^N (y_i - \hat{y}_i)^2 = \|\mathbf{y} - \hat{\mathbf{y}}\|^2 \quad (2.15)$$

which is a quadratic function of the parameters $\hat{\boldsymbol{\beta}}$. To find the minimum value of RSS , we differentiate RSS with respect to $\hat{\boldsymbol{\beta}}$ and get the normal equations:

$$\mathbf{X}^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) = 0 \quad (2.16)$$

If $\mathbf{X}^T\mathbf{X}$ is nonsingular, then the unique solution of $\hat{\boldsymbol{\beta}}$ is given by:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \quad (2.17)$$

Eq. (2.17) is so-called mean-square-error (MSE) solutions. In the context of pattern classification, the predicted value $\hat{y}$ of any input unknown sample $\mathbf{x}$ given by Eq.

(2.14) is converted to class label by

$$L(\mathbf{x}) = \text{sign}(\hat{y}) = \text{sign}(\hat{\boldsymbol{\beta}}^T \mathbf{x}) \quad (2.18)$$

The advantage of the least squares classifier is that it is not only simple but also very efficient in training and classification. However, least squares classifier can only work well for 2-class problem, due to its rigid nature. For multi-class problem, the more general methods, linear/quadratic discriminant analysis, could be used as reviewed next.

Linear/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:

$$f_l(\mathbf{x}) = \mathbf{x}^T \Sigma^{-1} \boldsymbol{\mu}_l - \frac{1}{2} \boldsymbol{\mu}_l^T \Sigma^{-1} \boldsymbol{\mu}_l + \log p(l) \quad (2.19)$$

where $\boldsymbol{\mu}_l$ denoted as the mean vector of class l and Σ denoted as the covariance matrix. Eq. (2.19) is an equivalent description of decision rule, corresponding to posterior probability of class l with input vector $\mathbf{x}$. It outputs an optimal class label with the maximum posterior probability.

$$L(\mathbf{x}) = \arg \max_l f_l(\mathbf{x}) \quad (2.20)$$

If the covariance matrix of each class is not assumed to be the same, the linear

discriminant functions in Eq. (2.19) could be modified to quadratic linear discriminant functions as below:

$$f'_l(\mathbf{x}) = \mathbf{x}^T \Sigma_l^{-1} \boldsymbol{\mu}_l - \frac{1}{2} \boldsymbol{\mu}_l^T \Sigma_l^{-1} \boldsymbol{\mu}_l + \log p(l) \quad (2.21)$$

where Σ_l is covariance matrix of the l 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 $\Sigma(\alpha)$ is defined as

$$\Sigma_l(\alpha) = \alpha \Sigma_l + (1 - \alpha) \Sigma \quad (2.22)$$

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

Artificial Neural Network

Artificial neural network (ANN) (Haykin, 1999) is another example in this categorical. Single-layer perceptron, where the separating hyperplane (defined by weights) is iteratively updated as a function of distances from the misclassified patterns to this hyperplane. When the threshold function is sigmoid function, single-layer perceptron shows a similar behavior of a linear classifier. Non-linear separating boundary can be learnt by adding some hidden layers, as multi-layer perceptron. However,

this also raises the danger of overfitting, i.e. the classifier fits perfectly to training data but can not generalize to the unseen data.

Radial-Basis Function (RBF) networks (Buhmann and Ablowitz, 2003) use a non-linear kernel function to project the input data into a space where the learning problem can be solved using a linear model, found in *Cover's theorem* on the separability of patterns:

A complex pattern-classification problem cast in a high-dimensional space nonlinearly is more likely to be linearly separable than in a low-dimensional space.

Similar idea is also used in support vector machines. RBF networks have two layers of processing: In the first layer, input is mapped onto each RBF in the hidden layer. The RBF chosen is usually a Gaussian. In regression problems, the output layer is then a linear combination of hidden layer values representing mean predicted output. The interpretation of this output layer value is the same as a regression model in statistics. In classification problems the output layer is typically a sigmoid function of a linear combination of hidden layer values, representing a posterior probability.

To deal with the overfitting problem occurred in neural networks, regularization is often introduced to shrink the values of parameters learned and smooth the classification boundary output, which is analogous to ridge regression for linear discriminant analysis.

Support Vector Machines

The most interesting developments recently in classification are the introduction of support vector machines (SVM) (Vapnik, 1996, 1998). SVM separates data from two classes by an *optimal separating hyperplane* which maximizes the distance to the closet training samples from either class (Vapnik, 1996). By maximizing the margin between two classes on the training data, the optimal separating hyperplane leads to better classification performance on unseen data (Hastie et al., 2001).

Assume a set of training samples $(\mathbf{x}_i, y_i)$, $i = 1, \dots, N$, each of which is represented by d 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):

$$\mathbf{w}^T \mathbf{x} + b = 0 \quad (2.23)$$

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

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1, \quad i = 1, 2, \dots, N \quad (2.24)$$

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

$$\begin{aligned} & \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{subject to: } & y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1, \quad i = 1, 2, \dots, N \end{aligned} \quad (2.25)$$

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

$$\mathcal{L}(\mathbf{w}, b) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^N \alpha_i [y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1] \quad (2.26)$$

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

$$\mathbf{w} = \sum_{i=1}^N \alpha_i y_i \mathbf{x}_i \quad (2.27)$$

$$0 = \sum_{i=1}^N \alpha_i y_i \quad (2.28)$$

Substituting these into Eq. (2.26), we obtain the dual form,

$$\mathcal{L}_d(\boldsymbol{\alpha}) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \quad (2.29)$$

$$\text{subject to: } \alpha_i \geq 0, i = 1, 2, \dots, N$$

It is a quadratic programming (QP) problem. To find an optimal classifier, we have to find the parameters α^* that maximize Eq. (2.29). Substitute α^* to Eq. (2.27), we obtain $\mathbf{w}^*$. The parameters α can be explicitly determined using Eq. (2.29), but the bias b can not. The bias b is determined by using the Karush-Kuhn-Tucker (KKT) complementarity conditions (see Hastie et al., 2001, page 374). These conditions state that the optimal solutions α^* and b^* must satisfy

$$\alpha_i^* [y_i(\mathbf{w}^{*T} \mathbf{x}_i + b) - 1] = 0. \quad (2.30)$$

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

$$L(\mathbf{x}) = \text{sign} [\mathbf{w}^{*T} \mathbf{x} + b^*] = \text{sign} \left[\sum_{i=1}^N \alpha_i^* y_i \mathbf{x}_i^T \mathbf{x} + b^* \right] \quad (2.31)$$

Eq. (2.27) 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 nearest neighbor classifier because the decision boundary of nearest neighbor classifier is also determined by the patterns around boundary area.

In linearly non-separable case, the optimization problem as defined in Eq. (2.25) 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:

$$\begin{aligned}
& \min_{\mathbf{w}, b, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i \\
& \text{subject to:} \quad y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \\
& \quad \xi_i \geq 0, i = 1, 2, \dots, N
\end{aligned} \tag{2.32}$$

where the slack variable ξ_i is the regression error for training sample $\mathbf{x}_i$. 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

$$\mathcal{L}(\mathbf{w}, b) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \alpha_i [y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^N \beta_i \xi_i \tag{2.33}$$

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

$$\begin{aligned}
\mathcal{L}_d(\boldsymbol{\alpha}) &= \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \\
&\text{subject to: } 0 \leq \alpha_i \leq C, i = 1, 2, \dots, N
\end{aligned} \tag{2.34}$$

After finding the Lagrange multipliers α_i^* that maximize Eq. (2.34), we obtain the solution for optimization problem (2.32),

$$\mathbf{w}^* = \sum_{i=1}^N y_i \alpha_i^* \mathbf{x}_i \quad (2.35)$$

$$b^* = y_i - \mathbf{w}^{*T} \mathbf{x}_i, \text{ for any } i \text{ with } \alpha_i > 0 \quad (2.36)$$

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 $\mathbf{x}_i$:

$$f(\mathbf{x}) = \sum_{i=1}^N \alpha_i^* y_i K(\mathbf{x}_i, \mathbf{x}) + b^* \quad (2.37)$$

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:

Polynomial: $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j)^q$

Radial Basis Function: $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$, $\gamma > 0$

Sigmoid: $K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(k \mathbf{x}_i^T \mathbf{x}_j + c)$, for some $k > 0$ and $c < 0$

In Eq. (2.37), α_i^* can be obtained by solving the optimization problem,

$$\begin{aligned}
&\text{maximize:} && \mathcal{L}_d(\boldsymbol{\alpha}) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \\
&\text{subject to:} && \sum_{i=1}^N y_i \alpha_i = 0, \quad \forall i : 0 \leq \alpha_i \leq C, \quad i = 1, 2, \dots, N
\end{aligned} \tag{2.38}$$

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

2.2.4 Classifiers Based on Decision Trees

Decision trees are trained by iterative selection of individual features that is the most informative at each node of the tree. Each node, which corresponds to a feature subspace containing a subset of training patterns, is split to two or more children based on feature values. 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.

Decision trees are different depend on how to split, how to process continuous 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 to two children nodes by asking, for example, the *color* is *red* or not; the *age* is greater than 40 or not? Besides binary splitting, multiway 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. Multiway

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). The other improvement of C4.5 relies on how to integrate continuous features. ID3 discretizes the continuous feature before learning so that it does not take full advantage of order information. However, C4.5 discretizes continuous features in each node (locally), just like CART, which alleviates the loss of order information in continuous features.

Characteristics of classification algorithms reviewed in this section are summarized in Table 2.1.

Classification algorithm	Characteristic			
	natural feature type	linear vs. non-linear	local vs. global	able to deal with irrelevant features
Support Vector Machine	continuous	both	local	no
Decision Trees	mixed	non-linear	local	yes
k-Nearest Neighbor	continuous	non-linear	local	yes
Naive Bayes	mixed	non-linear	global	no
Linear Discriminant Analysis	continuous	linear	global	no
Logistic Regression	mixed	linear	global	no

Table 2.1: Characteristics of different classification algorithms

2.3 Classification Accuracy Estimation

Predicting the accuracy of learning algorithms for future data is a fundamental issue in supervised learning. A good accuracy estimation method is not only important for evaluating the usefulness of feature selection and classifier designing but also

useful for tuning the parameters in a learning algorithm.

Given a finite training dataset D drawn from the true unknown distribution $\mathcal{D}$, accuracy of the learning model based on D for the true distribution $\mathcal{D}$ could be simulated explicitly by *resampling* with such a assumption that D is similar to the true distribution $\mathcal{D}$ and a sample subset, called training dataset, is similar to original sample set D . Samples that do not appear in the training dataset, called test dataset, is hold from training process and then used to evaluate the learning model.

In this section, we first introduce the estimation of error rate index and how the quality of error rate estimate is measured by confidence interval. Then, we introduce how to estimate the generalization error based on *resampling* (Weiss and Kulikowski, 1991; Efron and Tibshirani, 1993, 1997; Shao and Tu, 1995). Good reviews for generalization error estimation can be found in Devijver and Kittler (1982); Kohavi and John (1997) and Duda et al. (2001).

2.3.1 Error Estimation with A Single Validation Set

Suppose a known classifier $\mathcal{L}$ and a set of test data $D_t = \{\mathbf{x}_i, y_i\}_{i=1}^n$, where y_i is the true class label of $\mathbf{x}_i$. Let E be the actual error probability of classification we want to know. Let the random variable $\delta(\mathbf{x}_i)$ equal 0 if the class label assigned to $\mathbf{x}_i$ by classifier $\mathcal{L}$ is y_i and 1, otherwise. Then, for any $\mathbf{x}_i$, $\delta(\mathbf{x}_i)$ is a Bernoulli random variable that takes value 1 with probability E and value 0 with probability $1 - E$. Let $\tau = \sum_{i=1}^n \delta(\mathbf{x}_i)$ denote the number of misclassifications of the classifier $\mathcal{L}$, and

τ follows a binomial distribution (i.e. the sum of Bernoulli distribution):

$$P(\tau = \nu | E) = \binom{n}{\nu} E^\nu (1 - E)^{n-\nu} \quad (2.39)$$

The fraction $\frac{\tau}{n}$ is therefore the maximum likelihood estimate $\hat{E}$ of E and yields

$$\mu(\hat{E}) = E; \quad \sigma^2(\hat{E}) = \frac{E(1 - E)}{n} \quad (2.40)$$

where μ and σ^2 is the expectation and variance of the estimate $\hat{E}$. This estimate is unbiased and consistent. Generally, $\hat{E}$ is not exactly equal to E but is equal to some value around E . But if no other information is given, $\hat{E}$ is the best estimate for E . Fortunately, *confidence interval* (see Rice, 1995) provides us an appropriate tool to quantify the quality of single estimate $\hat{E}$. A $100\gamma\%$ confidence interval of E is an interval that includes E with probability $100\gamma\%$, where $0 \leq \gamma \leq 1$. As the estimate $\hat{E}$ follows a normal distribution $\mathcal{N}(E, \frac{E(1-E)}{n})$, the equation below is governed by *central limited theorem*:

$$P \left[-z_\gamma < \frac{\hat{E} - E}{\sqrt{\frac{E(1-E)}{n}}} < z_\gamma \right] = \gamma \quad (2.41)$$

where z_γ is the $[\frac{1-\gamma}{2}]^{th}$ quantile point of the standard normal distribution $\mathcal{N}(0, 1)$. z_γ is given by $\Phi(z_\gamma) - \Phi(-z_\gamma) = \gamma$ where $\Phi(z_\gamma)$ is cumulative standard normal distribution

$$\Phi(z_\gamma) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z_\gamma} e^{-\frac{x^2}{2}} dx \quad (2.42)$$

An approximate $100\gamma\%$ confidence interval for E is obtained by inverting the in-

equality inside Eq.(2.41), yielding

$$P \left[\hat{E} - z_\gamma \sqrt{\frac{\hat{E}(1 - \hat{E})}{n}} < E < \hat{E} + z_\gamma \sqrt{\frac{\hat{E}(1 - \hat{E})}{n}} \right] = \gamma \quad (2.43)$$

For example, the 95% (i.e. $\gamma = 0.95$) confidence interval of E , with $z_{0.95} = 1.96$ that we find in the table of cumulative standard normal distribution (Devijver and Kittler, 1982), is as follow.

$$P \left[\hat{E} - 1.96 \sqrt{\frac{\hat{E}(1 - \hat{E})}{n}} < E < \hat{E} + 1.96 \sqrt{\frac{\hat{E}(1 - \hat{E})}{n}} \right] = 0.95 \quad (2.44)$$

2.3.2 Validation Methods

In the previous section, we have introduced the estimation of error rate of a known classifier by an ideal testing set. Unfortunately, such an ideal testing set is not readily available in practice. Given a finite set of data D drawn from the underlying population $\mathcal{D}$, it is reasonable to repeatedly partition the data D into mutually exclusive training and test sets and return the mean of the test set errors for these trials. If this experiment is repeated M times, each of the M individual estimates of error rate, denoted as $\hat{E}_m$ ($m = 1, 2, \dots, M$), follows normal distribution as presented in the previous section. The mean error rate, denoted as $\bar{E}$, which is also called *sample mean*, is given as

$$\bar{E} = \frac{1}{M} \sum_{m=1}^M \hat{E}_m \quad (2.45)$$

$\bar{E}$ is the estimate of true error rate E . In this case, the *sample variance*, denoted as $s_{\bar{E}}^2$ can be estimated by

$$s_{\bar{E}}^2 = \frac{1}{M(M-1)} \sum_{m=1}^M (\hat{E}_m - \bar{E})^2 \quad (2.46)$$

Since each estimated error rate $\hat{E}_m$ follows a normal distribution, as presented in the previous section, and the sample variance follows the χ^2 distribution, $\bar{E}$ will follow a t -distribution and the approximate confidence interval for estimating the true error rate E using $\bar{E}$ is given by

$$P [\bar{E} - t_{\gamma, M-1} s_{\bar{E}} < E < \bar{E} + t_{\gamma, M-1} s_{\bar{E}}] = \gamma \quad (2.47)$$

where $t_{\gamma, M-1}$ is a constant that plays a role analogous to that of z_{γ} in the confidence interval expression for normal distribution in Eq. (2.43). Notice the constant $t_{\gamma, M-1}$ has two subscripts that specifies the desired confidence level, as it did for z_{γ} , and *degrees of freedom*, related to the number of independent random events for the random variable $\bar{E}$, respectively. Note that $t_{\gamma, M-1}$ approaches z_{γ} if $M \rightarrow \infty$. Actually, when M is greater than 30, t -distribution is very close to normal distribution (Rice, 1995).

Several resampling methods, such as *holdout*, *cross-validation* and *bootstrapping*, as we shall introduce below, are often used for choosing among various learning models or application domains. These methods are different based on the fashions of sampling, the sizes of training/test subsets defined and the definitions of returned estimates.

Holdout

The holdout method randomly partitions the given data D into two disjoint subsets: one subset, called training set, is used for designing the classifier and the other subset, called test set, is used to test this classifier. Normally $1/3$ of data is randomly sampled to form the test set while the rest $2/3$ of data is used as training set. If the test data is denoted as D_t that contains h samples, the training data is $D \setminus D_t$ (where $\setminus$ denotes operation of set subtraction). The holdout error rate $\hat{E}_{hold}$ is the ratio of the number of misclassifications on the test set to the number of samples of the test set.

$$\hat{E}_{hold} = \frac{1}{h} \sum_{(\mathbf{x}_i, y_i) \in D_t} \delta(\mathcal{L}_{D \setminus D_t}(\mathbf{x}_i), y_i) \quad (2.48)$$

where the notation $\mathcal{L}_D(\mathbf{x}_i)$ represents the class label assigned to sample $\mathbf{x}_i$ by the classifier trained upon training sample set D . $\delta(i, j) = 1$ if $i \neq j$ and 0 otherwise.

Since in holdout method, classifier is designed on only a portion of the data, it can be inferior to the classifier designed on the entire dataset (Devijver and Kittler, 1982; Weiss and Kulikowski, 1991). If we leave the more data for training, the less number of test data will lead to a wider confidence interval as shown in Eq. (2.43). Next method will make better use of the available data and reduce the bias of the error estimate.

Leave-One-Out Cross-Validation

Leave-one-out cross-validation (LOOCV) leaves one sample, say $(\mathbf{x}_i, y_i)$, from the given dataset D . Classifier is trained using $D \setminus (\mathbf{x}_i, y_i)$ (where $\setminus$ denotes operation of set subtraction) with $N - 1$ number of samples and tested using $(\mathbf{x}_i, y_i)$. This process is repeated for N times. The estimate of error rate $\hat{E}$ is the empirical frequency of classification error over N .

$$\hat{E}_{loocv} = \frac{1}{N} \sum_{i=1}^N \delta(\mathcal{L}_{D \setminus (\mathbf{x}_i, y_i)}(\mathbf{x}_i), y_i) \quad (2.49)$$

where the notation $\mathcal{L}_D(\mathbf{x}_i)$ represents the class label assigned to sample $\mathbf{x}_i$ by the classifier trained upon training sample set D . $\delta(i, j) = 1$ if $i \neq j$ and 0 otherwise. In this method, all the available samples are ultimately used for classifier training and test, although in each partition, the training and test sets are regarded as independent.

However, the disadvantage of LOOCV is that the variance of LOOCV is high, particularly for small-sized data. These drawbacks can be improved by K -fold cross-validation as introduced in next section.

K -Fold Cross-Validation

In K -fold cross-validation, the full dataset D is divided into K mutually exclusive subsets D_k ($k = 1, \dots, K$), of approximately equal size. The training and test procedure is performed K times. Each time the classifier is trained on $D \setminus D_k$ (where $\setminus$ denotes operation of set subtraction) and tested on D_k . The cross-validation error

rate is the ratio of overall misclassifications of K test subsets to the number of samples in D .

$$\hat{E}_{kcv} = \frac{1}{N} \sum_{k=1}^K \sum_{\mathbf{x}_i \in D_k} \delta(\mathcal{L}_{D \setminus D_k}(\mathbf{x}_i), y_i) \quad (2.50)$$

where the notation $\mathcal{L}_D(\mathbf{x}_i)$ represents the class label assigned to sample $\mathbf{x}_i$ by the classifier trained upon training sample set D . $\delta(i, j) = 1$ if $i \neq j$ and 0 otherwise. When K equals to the number of samples in the given dataset, K -fold cross-validation is reduced to leave-one-out cross-validation. In practice, 10-fold cross-validation is most commonly used for error estimation. A common rule of thumb is that we can use normal distribution when there are at least 30 samples (Mitchell, 1997). Therefore, we define the value of K in terms of the requirement of at least 30 samples in test set throughout the thesis.

Bootstrap

In bootstrap method, subsamples of data rather than subsets of data are repeatedly generated for error estimation. A bootstrap sample set D^* is created by sampling N samples uniformly from the original dataset D with replacement. Some samples will appear in bootstrap sample more than once, while others may not appear at all. The probability of any sample not being chosen after N samples is $(1 - 1/N)^N \approx e^{-1} = .368$, thus the probability of any sample appearing in bootstrap sample is .632.

In practice, the bootstrap error $\hat{E}_B$ is approximated overall B independent sam-

ple set D_b^* , $b = 1, \dots, B$, as given by:

$$\hat{E}_B = \frac{1}{B} \sum_{b=1}^B (\hat{E}_b) \quad (2.51)$$

where the error rate $\hat{E}_b$ is derived by using bootstrap sample set D_b^* for training and the samples that do not appear in bootstrap sample D_b^* for testing. The $\hat{E}_B$ is upward biased, and Efron and Tibshirani (1993) proposed the .632 bootstrap error estimator to correct the upward bias, as defined in Eq. (2.52).

$$\hat{E}_{B.632} = \frac{1}{B} \sum_{b=1}^B (.632 \cdot \hat{E}_b + .368 \cdot \overline{err}) \quad (2.52)$$

where $\overline{err}$ is the apparent error rate estimated by using the original dataset for both training and testing, as defined in Eq. (2.53).

$$\overline{err} = \frac{1}{N} \sum_{i=1}^N \delta(\mathcal{L}_D(\mathbf{x}_i), y_i) \quad (2.53)$$

Efron and Tibshirani (1997) proposed a less biased version of .632 bootstrap, called .632+ bootstrap error estimator, as defined by Eq. (2.54).

$$\begin{aligned} \hat{E}_{B.632+} &= \frac{1}{B} \sum_{b=1}^B (w_b \cdot \hat{E}_b + (1 - w_b) \cdot \overline{err}) \\ &\left(w_b = \frac{.632}{1 - .368 \hat{R}_b} \right) \end{aligned} \quad (2.54)$$

where $\hat{R}_b$ is *relative overfitting rate* calculated in b th round. Let $\hat{p}_{1b}$ be proportion of

testing data in b th round truly from class 1, and let $\hat{q}_{1b}$ be the proportion of testing data with a prediction of class 1. $\hat{R}_b$ can be defined as:

$$\hat{R}_b = \frac{\hat{E}_b - \overline{err}}{\hat{\gamma}_b - \overline{err}}$$

$$\hat{\gamma}_b = \hat{p}_{1b}(1 - \hat{p}_{1b}) + (1 - \hat{p}_{1b})\hat{q}_{1b} \quad (2.55)$$

The computational cost of bootstrap error estimator mainly depends on the number of bootstrap samples (i.e. B). In statistics, B is usually set to 200 or higher, which could be a big computational burden. One noticeable advantage of bootstrap error estimator is that the variability of bootstrap error estimator is relatively low, while leave-one-out or even K -fold cross-validation may exhibit excessive variability (Braga-Neto and Dougherty, 2004b). Therefore, for data with small sample size, where the variability of estimation could be intrinsically high, bootstrap error estimator could be a better choice (Zhou and Mao, 2005).

Chapter 3

Literature Review of Feature Selection

3.1 Why Feature Selection?

Given a pattern recognition task, the quality of data input is very essential to the success of the ultimate model learnt. The saying “garbage in, garbage out” is a suitable description in this case. Unfortunately, data collected from real-world always contains a mass of irrelevant information for the learning task. This is because data collected may be not solely for one particular task or one objective. When a data is prepared for multipurpose, the existence of *irrelevant* and *redundant* features is inevitable for one particular task. Even if the particular task is known, we have no precise idea about features, therefore we always introduce as more candidate features as possible. In this case, we also can hardly avoid collecting of irrelevant and redundant features.

Theoretically, more features imply more discriminant power and hence result in a more accurate classification/prediction. However, this is not tenable in practice

due to the limited number of samples available. Many classification algorithms can be regarded as making probability estimation for target classes given features, and will break down in high dimensional feature space. Hastie et al. (2001) examined the behavior of k -nearest neighbor in high dimensions and found that it could fail due to the low density of neighbors in neighborhood. Decision trees such as C4.5 (Quinlan, 1993) can overfit the training data to grow a very large tree with many irrelevant features. Naive Bayes classifier have been reported to be sensitive to redundant features (Langley and Sage, 1994). These phenomena are commonly referred as “overfitting problem”, i.e. classifier perfectly fits to the training data but generalize poorly to unseen data.

A simple but effective way of dealing with overfitting problem is to reduce the number of feature dimensions, by eliminating irrelevant and redundant features. Feature selection is just such a technique. By removing the irrelevant and redundant features, the simpler classifier that is learnt using only relevant feature subset could generalize better to unseen data and hence could lead to better classification accuracy. Feature selection also makes the classification more efficient and easier to be understood. Last but not the least, feature selection reduces the cost of data acquisition while the irrelevant and redundant features identified need not be collected again. Just like Occam’s Razor principle indicated, “all things being equal, the simplest solution tends to be the best one” (Gamberger and Lavrac, 1997).

As its importance to machine learning, feature selection has received considerable attentions in the past two decades. Excellent reviews for feature selection can be

referred to Devijver and Kittler (1982); Kohavi (1995b); Blum and Langley (1997); Hall (1998); Liu and Motoda (1998); Liu et al. (2002); Guyon and Elisseeff (2003); Liu and Yu (2005).

This chapter begins by introducing the basic components of feature selection algorithms in Section 3.2. Section 3.3 reviews important current approaches of feature selection, organized by filter methods, wrapper methods and hybrid methods of feature selection. Finally, we briefly summary this chapter in Section 3.4.

3.2 The Components of A Feature Selection Algorithm

Feature selection algorithms search through feature subspaces for optimal feature subset. As a result, the nature of a typical feature selection algorithm is dominated by three components: search procedure, evaluation criterion and stopping criterion. A *search procedure* generates candidate feature subsets, the goodness of which are evaluated by an *evaluation criterion*. This evaluation result is then feedback to search algorithm and used to update previous selected subset of features. This selection process is continued until a *stopping criterion* is met. Usually, selection procedure is stopped if the classification accuracy or the class separability is no longer increased by adding features or degraded by deleting features.

3.2.1 Search Procedure

Feature selection can be viewed as a search problem, aiming to search through feature spaces for an optimal one. For a data set with d features, there are all together 2^d different feature subsets. The direct exhaustive search, using forward selective direction for example, could generate $\binom{d}{0} + \binom{d}{1} + \cdots + \binom{d}{p} = \frac{d!}{(d-p)!p!}$ feature subsets for evaluation until an optimal subset of p features is found. Clearly, even for moderate values of p and d , exhaustive search is too expensive to be practical.

In practice, suboptimal search procedures are often used to find a good solution, if not the optimal one, at a reasonable computational cost. This suboptimality is often necessary because the trade-off of optimality for searching speed brings much faster speed and little loss of optimality. The commonly used suboptimal search procedures are reviewed below.

Heuristic Search

Heuristic search is a method that is first used in the context of Artificial Intelligence (AI). The heuristics, also called “rules of thumb”, could be used to generate good solutions without exhaustive search (Russell and Norvig, 1995). Among various heuristic search algorithm, the hill-climbing strategy is the most commonly used.

Hill-climbing strategy, such as the well-known Sequential Forward Selection (SFS) and Sequential Backward Selection (SBS), considers local changes and makes the best changes to improve the current state iteratively. SFS is a “bottom-up” method, starting with the empty set and adding one feature to the feature subset at each se-

lection step. At each step, the feature is selected from the candidate feature subset so that the new enlarged feature set yields an optimal value of the evaluation criterion. SBS is the “top-down” method, starting with an full feature set and discarding one feature at each step until the final feature subset is found. SBS may reflect feature relations in feature subset better (Devijver and Kittler, 1982). However, to delete redundant features would be problematic for SBS when many features are highly correlated. In this case, exclusion of any one will not improve the performance because the redundancy will still exist, while SFS will immediately detect a harmful redundant feature once it is added (Langley and Sage, 1994). For high dimensional data, SFS is the first choice and can go forward until the evaluation of generated feature subset is unaffordable.

The drawback of Sequential Forward/Backward Selection is that they may be stuck in local optimum, and thus the algorithm will halt even though this solution is far from satisfactory. Floating search method (Pudil et al., 1994), including SFFS (Sequential Floating Forward Selection) and SFBS (Sequential Floating Backward Selection), alleviates the local optimal problem in SFS/SBS by allowing backtracking in search processing. The basic idea of SFFS is after each forward selection step, applying a number of backward steps. In each backward step, the worst feature could be excluded if the exclusion could obtain a better performance (estimated by evaluation criterion) than the previous output subsets with the same number of features. The backward step is continued until no further exclusion could lead to a better feature subset. The same applies for SFBS procedure. By applying

backtracking, the floating search method is able to correct wrong inclusion/removal decisions, therefore, the local optimal problem can be alleviated. However, the floating search method has no theoretical bound on computational cost due to its nature. Experiments show that one can expect roughly 10 times more computational cost than SFS/SBS for datasets with a moderate number of features. To deal with datasets containing a large number of features, floating search method may be computationally prohibitive (Pudil et al., 1994).

Random Search

The competitive alternative of heuristic search is *random search*, where genetic algorithm (GA) is a typical example (Siedlecki and Sklansky, 1989). GA approach manipulates a set of feature subsets in “population”, each of which is represented by a binary string (“chromosome”) with fixed length d , i.e. the total number of features. The i th position in string represents the status of feature i , with 1 and 0 for presence and absence of feature i , respectively. Each chromosome is evaluated by a “fitness” function, which evaluates how likely the chromosome is to survive in the next generation. Offsprings are generated and added to the population by the process of (i) crossover that exchanges parts of two chromosomes and (ii) mutation that flips the status of randomly selected positions (from 1 to 0 or reverse). Since better feature subsets have a greater chance to be selected to generate offsprings through crossover and mutation, good subsets are searched over evaluation.

The advantage of random search is that it could alleviate the local optimum problem occurred in SFS and SBS. Ferri et al. (1994) compared the performance

of GA-based and sequential-based search algorithms for datasets with up to 360 dimensions. The experimental results reveal that the performance of GA degrades with the increasing dimensionality of data, where the floating search method yields a better result.

3.2.2 Evaluation Criterion

To identify a good feature subset, an evaluation criterion is needed to measure the goodness of feature subsets. The evaluation criteria in the literature of feature selection can be classified into two categories: classifier-dependent and classifier-independent.

The first category of criteria evaluate feature subsets based on the performance of a particular classifier. For each feature subset, the classifier is induced and the classification is performed on a separate validation sample set. The proportion of samples misclassified, i.e. error rate, is the typical measure in this category. This category of criteria finds feature subsets to fit a particular classifier. When a different ultimate classification algorithm is used, all the feature selection process need to be rerun.

The second category of evaluation criteria is the classifier-independent criteria, which evaluate the goodness of a feature subset based on the inherent property of the data, such as class separability measure. The feature subset with a large class separability is preferred because the larger the class separability, the more chance of successful classification. In the literature of feature selection, four kinds of measures

have been used to assess the class separability, i.e. distance measures, information measures, dependency measures, and consistency measures (Devijver and Kittler, 1982; Jain and Zonker, 1997; Liu and Motoda, 1998; Liu and Yu, 2005).

In fact, the feature selection algorithms that employ these two categories of criteria are referred to wrapper methods (Kohavi, 1995b; Kohavi and John, 1997) and filter methods (Liu and Motoda, 1998; Liu and Yu, 2005; Guyon and Elisseeff, 2003), respectively. The wrapper method often produces feature subsets leading to better classification result because it takes the bias of induction algorithm into account during feature evaluation (Kohavi, 1995b). However, the wrapper method is computationally more expensive since it runs induction algorithm repeatedly in feature evaluation. When dealing with large datasets, the filter method is often the first choice for its computational efficiency. Recently, filter-wrapper hybrid methods were proposed to integrate advantages of both filter and wrapper methods (Das, 2001). This method provides a more flexible way to handle data with a large number of features. Various feature selection algorithms, which can be classified into these three categories, will be reviewed in next section.

3.3 Existing Feature Selection Algorithms

3.3.1 Wrapper Methods

John et al. (1994) first used “wrapper” as a general category of feature selection in the context of machine learning. They used heuristic search algorithms such

as sequential forward selection and backward selection to search feature subspace and evaluated the subset using k -fold cross-validation ($k = 25$ in their work) on induction algorithms ID3 (Quinlan, 1986) and C4.5 (Quinlan, 1993). This ID3- and C4.5-wrapper are tested on nine datasets including three artificial data sets and six real-world datasets. The experimental results showed that, except one artificial domain, feature selection did not significantly change ID3 and C4.5's generalization performance and the main advantages of feature selection is to reduce the size of decision tree (Kohavi, 1995b).

Kohavi (1995a)'s IDTM algorithm (Inducer of Decision Table Majority) used wrapper method to find an optimal feature subset for building a decision table with a highest future classification accuracy (lowest error rate). IDTM employed forward selection as a search procedure. Experiences on both artificial and real-world data sets showed that the prediction accuracy of IDTM was comparable to and sometimes outperformed that of C4.5. One limitation of IDTM is it can only employed for categorical features. Therefore, continuous features in the dataset need be discretized into categorical ones before being evaluated by IDTM. Different from C4.5 that discretizes the continuous features in some specific tree node locally, IDTM conducts the discretization globally. This could lose a lot of information in the continuous features.

Langley and Sage (1994) proposed selective Naive Bayes using a wrapper method to delete redundant features, which may violate the independent hypothesis of Naive Bayes classifier. Sequential forward selection was employed as the search algorithm

because forward selection will detect a redundant feature once it is added in. To evaluate the classification accuracy of a feature subset, *hold-out* evaluation method was employed. The selective Naive Bayes was tested on six real-world domains and compared with C4.5 with pruning. The experimental results showed that selective Naive Bayes significantly outperformed plain Naive Bayes in the domains containing redundant features and it did as well as the plain Naive Bayes where the latter already outperformed decision-tree induction. Improvement for naive Bayes using wrapper-based feature selection has also been reported by Kohavi and Sommerfield (1995) and Kohavi and John (1997).

Singh and Provan (1995) and Singh (1996) have applied the wrapper to select features, upon which Bayesian networks are built. Their results showed that while feature selection did not improve accuracy over networks constructed from the full set of features, the networks constructed after feature selection were considerably smaller and faster to learn.

Although the relatively high accuracy of the wrapper method has been reported in many papers, the limitation of the wrapper method is also obvious, i.e. the high computational cost involved. For a dataset with a large number of features, the wrapper method may be impractical. In this case, filter methods, which is computationally more efficient in general, should be a better choice.

3.3.2 Filter Methods

In this section, we will review the existing filter feature selection algorithms in the context of supervised learning. The filter feature selection algorithms are introduced in chronological sequence to show the progress of algorithms development, which implies the developing challenges for feature selection.

Classifier-Based Filters

Some particular classifiers embed feature selection and have been employed as filter for another induction algorithm. The process of classifier training can be viewed as searching best feature subsets that help to distinguish different classes.

Separate-and-conquer methods for learning decision rules (Michalski, 1980; Clark and Niblett, 1989; Pagallo and Haussler, 1990) embed feature selection by using an evaluation function to discover feature that helps distinguish class l from others, then adding the best feature to a single conjunctive rule for class l . These methods repeat this process until there is no pattern that is misclassified to class l , then remove all the members of class l that the rule covers and go on this process for the remaining training patterns. Finally, the subset of features appears in all decision rules learnt is output as the selective feature subset.

Recursive partitioning-based classifier, including decision table such as Decision Table Majority and decision trees such as ID3 (Quinlan, 1986), C4.5 (Quinlan, 1993), CART (Breiman et al., 1984), oblivious decision tree (Kohavi and Li, 1995), also embed feature selection and have been explored for finding the feature subsets for

another classifier.

Pfahring (1995) proposed an algorithm to use decision table majority classifier to select features for C4.5 classifier. The decision table majority (DTM) classifier is trained by greedily searching the feature space using minimum description length (MDL) as the evaluation criterion. The empirical results showed that DTM classifier induced by MDL outperforms the same classifier induced by cross-validation (wrapper method) as introduced in (Kohavi, 1995a).

Cardie (1993) used C4.5 to identify the features to be used in k-nearest neighbor and the experimental results showed the hybrid approach outperformed both C4.5 or k-nearest neighbor when used alone. Singh and Provan (1996) used oblivious decision tree for feature selection to improve the performance of Bayesian network in a similar manner. Feature subsets selected by three oblivious decision trees, each employing a different information splitting criterion, were used to construct Bayesian network classifier. Experimental results showed that the hybrid approach outperformed Bayesian network without feature selection and Bayesian networks with wrapper-based features selection.

Consistency-Based Filters

Consistency-based filters employ consistency measures as the feature evaluation criteria and the “min-features bias” rule (Press et al., 1988), i.e. if two feature subsets are consistent with the training data, the smaller one is preferred. The goal of consistency-based filters is to find a minimum number of features that separate classes as consistently as the full set of features can.

Baim (1988) described an algorithm, called VARSEL, to search feature subsets by using relevance measure from information theory, as used in ID3 (Quinlan, 1986). The “most relevant features” identified by VARSEL were then used to formulate rules by another classifier GEM (Michalski and Larson, 1978). VARSEL adopted *random search*, where small subsets of features were evaluated and the individual relevancy measure of each constitute feature was improved or degraded based on the performance of the selected subset as a whole. If the process converges, the most relevant features were then indicated by high relevancy measure.

One variation of VARSEL is to employ a hill-climbing search instead of random search, where features are added sequentially until a sufficiently discriminatory feature subset is found. This method is computationally more efficient, but it ignores interdependency among features.

Almuallim and Dietterich (1991) proposed a filter for boolean (binary) features called FOCUS by adopting “min-feature” bias. FOCUS begins with an empty set and exhaustively searches a higher-dimensional feature subspace until no inconsistency exists, i.e. no two patterns that agree on all the features selected but do not agree on the class. There are two weaknesses in FOCUS. Firstly, the exhaustive search is computationally prohibitive even for a small data set (as described in Section 3.2.1). Secondly, a strong bias towards consistency can be statistically unwarranted and may lead to overfitting of the training data because the algorithm will continue to add features to repair a single inconsistency. Authors proposed FOCUS-2 to address the problem of computational cost by introducing heuristic

search algorithm that traded optimality for computational efficiency (Almuallim and Dietterich, 1992, 1994).

Liu and Setiono (1996) described another consistency-based filter method, called LVF (Las Vegas algorithm for Feature selection). Like FOCUS, LVF is biased toward minimum feature subsets, but provides provision for tolerating noise if the level of noise is known. LVF randomly generates a different feature subset S in each round by using *Las Vegas Algorithm* (Bressard and Bratley, 1996). In each iteration, S is used to replace the current best feature subset if S is either a smaller subset satisfying a predefined inconsistent rate or equal-sized feature subset with a lower inconsistency rate than the current best one. This process is continued until the predefined maximum number of iterations is reached. To calculate the inconsistency rate (γ) for a dataset, all the patterns are first grouped based on their feature values, and in any group k , patterns match in their feature values. Then, the inconsistency rate can be calculated as follow:

$$\gamma = \frac{1}{N} \sum_k \left[N_k - \max_l N_{k,l} \right] \quad (3.1)$$

where N_k is number of matching patterns in the k th group, $N_{k,l}$ is number of these patterns with class label l .

Instance-Based Filters

RELIEF algorithm (Rendell and Kira, 1992) and its extensions are some of the most known filter methods for estimating the quality of both dependent and independent

features in classification problems. RELIEF was firstly introduced by Rendell and Kira (1992), and was then 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 (Liu et al., 2004).

As a typical instance-based filter, the idea behind the RELIEF algorithm is similar to the principle of k-nearest neighbors (KNN) classification: the closest sample of the same class, called *nearest hit* H , is expected to be closer than the closest sample of other classes, called *nearest miss* M , in dimensions of relevant features. For each pattern R_i , the distance between nearest miss M_i and nearest hit H_i in the dimension of X represents the discriminant ability of X on R_i . The overall score W of X is averaged over all samples $R_1, \dots, R_N$, as defined below.

$$W[X] = \sum_{i=1}^N [\text{diff}(X, R_i, M_i) - \text{diff}(X, R_i, H_i)] \quad (3.2)$$

where $\text{diff}(X, I_1, I_2)$ calculates the difference between the values of feature X for two instances I_1 and I_2 .

- For nominal feature:

$$\text{diff}(X, I_1, I_2) = \begin{cases} 0 & \text{value}(X, I_1) = \text{value}(X, I_2) \\ 1 & \text{otherwise} \end{cases} \quad (3.3)$$

- For continuous feature:

$$\text{diff}(X, I_1, I_2) = \frac{|\text{value}(X, I_1) - \text{value}(X, I_2)|}{\max(X) - \min(X)} \quad (3.4)$$

where, $value(X, I)$ denotes the value of the feature X in the instance I . Notice that the nearest miss and hit of each sample are found in the original feature space with whole set of features. This implies the best individual feature should preserve the most discriminating information in the full feature set.

RELIEF algorithm is computationally efficient, and therefore can be applied to datasets with a large number of features. However, RELIEF algorithm assigns a relevance score to each feature independently regardless of correlations in between, thus, those discriminating but redundant features could not be removed. Taking into account both advantages and disadvantages, RELIEF is a very good choice for feature pre-selection, by detecting irrelevant attributes (Florez-Lopez, 2000; Bins and Draper, 2001).

Domingos (1997) proposed another instance-based filter, called RC (Relevance in Context) for lazy learner (some times also called instance-based, case-based, memory-based or exemplar-based learner) on context-sensitive features. Context-sensitive features, the opposite concept of context-free features, are relevant only in some parts of the space; deleting them will hurt accuracy in those parts, while selecting them will hurt accuracy in the rest parts. To handle this kind of features, RC uses a cluster like approach to select different features for different instances. For each instance, RC finds the nearest instance from the same class (like nearest hit in RELIEF) and assumes that the features along which they differ are irrelevant. Remember that in consistency-based method like FOCUS, features are added if there is a inconsistency. On the contrary, RC firstly finds the nearest sample

with same class and then make them agree on features too. RC was tested on both real-world datasets and artificial datasets. The experimental results show that RC is indeed able to detect context dependency effects, which significantly improves the classification accuracy. One potential problem of RC is that it is unable to detect the redundant features because the values of redundant features in two samples could be very similar.

Correlation-Based Filters

Battiti (1994) proposed a feature selection algorithm, called MIFS (Mutual Information Feature Selection), to select an informative subset for a neural network classifier. This algorithm sequentially adds features by taking both the mutual information with respect to the class and with respect to the already-selected features into account with the goal of avoiding inclusion of redundant features. Given two random variables X and Y , their mutual information is defined in terms of their probabilistic density function $p(X)$, $p(Y)$ and $p(X, Y)$:

$$I(X, Y) = \int \int p(X, Y) \log \frac{p(X, Y)}{p(X)p(Y)} dX dY \quad (3.5)$$

Hall (1998) proposed a correlation-based feature selector (CFS), by using the hypothesis: *a good feature subset is one that contains 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

(Ghiselli, 1964):

$$J_{CFS}(S) = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \quad (3.6)$$

where $J_{CFS}(S)$ is the merit of feature subset S containing k features; $\overline{r_{cf}}$ and $\overline{r_{ff}}$ are averaged class-feature correlation and feature-feature correlation, respectively. This heuristic criterion can detect irrelevant features that are poorly correlated with class, and redundant features that are highly correlated with other features. CFS evaluates features independently, and therefore it can not identify strong associations among features. However, it can identify useful features in moderate level of association (Hall, 1998, 2000).

By using the similar hypothesis described above, Yu and Liu (2004) proposed an correlation-based feature selection algorithm, called FCBF (Fast Correlation-Based Filter). Different from CFS that handles irrelevant features by a heuristic criterion, FCBF introduced a new framework that decoupled the relevancy and redundancy analysis into two steps as follows.

Step1 select a set of relevant features by calculating the class-feature correlation value for each feature, selecting relevant features based on a predefined threshold δ and ordering them in a descending order according to their class-feature correlation values.

Step2 select predominant features from relevant ones by filtering the redundant features as follows:

- (a) let the reference feature F_j be the first element in ranking list.

- (b) for all features have lower ranks than F_j , remove F_i from ranking list if $Corr(F_i, F_j) \geq Corr(F_i, Y)$. Set the remaining feature right next to F_j as the new reference F_j .
- (c) repeat (b) until the ranking list is empty.

In FCBF, the correlation between two variables $Corr(X, Y)$ was estimated by symmetrical uncertainty (UC), as defined follow.

$$UC(X, Y) = 2.0 \times \left[\frac{H(Y) - H(Y|X)}{H(Y) + H(X)} \right] \quad (3.7)$$

where $H(Y)$ is the *entropy* of variable Y that measures the uncertainty of variable Y and $H(Y|X)$ is the *conditional entropy* of variable Y given X , as given below:

$$H(Y) = - \sum_{y \in \text{value}(Y)} p(y) \log_2 [p(y)] \quad (3.8)$$

$$H(Y|X) = - \sum_{x \in \text{value}(X)} p(x) \sum_{y \in Y} p(y|x) \log_2 [p(y|x)] \quad (3.9)$$

where the function $\text{value}(X)$ denotes all the possible values in the variable X .

FCBF is computationally very efficient, and can be applied to datasets with a huge number of features. However, the weakness of FCBF is that the number of features finally selected is uncertain due to the nature of search strategy and it may select insufficient number of features.

Peng et al. (2005) proposed a feature selection algorithm based on Mutual Information with a criterion of Max-Relevance and Min-Redundancy (mRMR).

$$J_{mRMR}(S) = D - R \quad (3.10)$$

with

$$D = \frac{1}{|S|} \sum_{X_j \in S} I(X_j, Y)$$

$$R = \frac{1}{|S|^2} \sum_{X_j, X_k \in S} I(X_j, X_k)$$

where D and R are average class-feature correlation and feature-feature correlation. $I(X, Y)$ is the mutual information of variables X and Y as defined in Eq. (3.5). mRMR criterion in Eq. (3.10) is quite similar with CFS's heuristic criterion in Eq. (3.6) except the ways of combination for relevance and redundancy. mRMR is very simple and computationally efficient, and thus can be used for very-high dimensional data. But it may not be the best way to evaluate the redundancy among features by simply averaging the pairwise feature-feature correlations, especially when the number of selected features is large.

3.3.3 Filter-Wrapper Hybrid Methods

The advantages and disadvantages of filter and wrapper methods for feature selection somewhat complement each other. Das (2001) examined the advantages and disadvantages of filter and wrapper methods for feature selection and proposed a hybrid method that begins with a filter method to decide one best subset for each cardinality and then use a wrapper method to select the final best subset over differ-

ent cardinalities. Empirical studies on six real-world problems show that the hybrid method is competitive with wrapper methods while being much faster and scales well to datasets with thousands of features.

Some other work combines filter and wrapper methods using a two-staged framework. In the first stage, a candidate subset with k features is preselected by a filter method, and in the second stage, the wrapper method is applied to select the final compact feature subset. This two-staged hybrid framework is popular in recent literatures of feature selection for very high-dimensional data such as gene expression data. Peng et al. (2005) proposed a two-staged feature selection method, which combined feature filter based on mRMR criterion (as described in Section 3.3.2) and wrapper method, to select a compact feature set with low computational cost. Peng et al. (2006) proposed another hybrid approach called *FR-Wrapper* by using both filter and wrapper method to balance the precision and computation cost.

The strength of hybrid methods are two folds. First of all, the hybrid methods combines the advantages of both filter and wrapper methods. Secondly, the trade-off between computational complexity of feature selection and the classification accuracy can be controlled explicitly by setting the number of candidate features to be selected by filter method. However, the problem here is how to choose the candidate feature size. Peng et al. (2006) chose only 50, 100 and 200 top-ranked features for candidate feature subset in their work. Peng et al. (2005) selected the k top-ranked genes (features) corresponding to the smallest consistent error (with both small mean and variance) in the inner cross-validation. Which one is better is still

not clear. This issue will be further discussed in Section 7.2.

3.4 Chapter Summary

In the context of pattern recognition, feature selection is a key step to improve the learning speed, the generalized classification accuracy and the interpretability of a classifier, by removing both *irrelevant* and *redundant* features. This section reviewed the basic components of feature selection algorithms and the various existing feature selection algorithms developed in past decades.

Practical feature selection algorithms always trade off the optimality for the computational efficiency because a little bit reduction in accuracy and a considerable reduction in computational cost is worthwhile. Recently, the efficiency of feature selection algorithms become more important because of the appearance of very high-dimensional data.

Chapter 4

Fast Feature selection Algorithm for High-Dimensional Data Based on Regularized Recursive Mahalanobis Measure

4.1 Introduction

Mahalanobis separability is a set-based classifier-independent measure, and has been frequently used in feature selection for conventional data (see Dash and Liu, 1997, for example). When applied to very high-dimensional data such as gene expression data, Mahalanobis separability measure is computationally very intensive or even prohibitive, due to the involvement of inverse operation of a huge number of very high-dimensional matrices. In this study, a recursive approach to Mahalanobis separability measure evaluation is proposed, with the goal of reducing computational overhead. Instead of evaluating Mahalanobis measure directly in high dimensional

space, the recursive approach evaluates the measure through a series of successive operations in 2-dimensional space, where matrix inverse can be obtained efficiently. Assume the d dimensions of the space are denoted by $X_1, X_2, \dots, X_d$. A new dimension Z_2 is first created by combining dimensions X_1 and X_2 . Z_2 is such a dimension that the major discriminative power underlying X_1 and X_2 is captured by it. The new dimension Z_2 is then combined with X_3 to create dimension Z_3 . This successive procedure is continued until dimension Z_d is created using Z_{d-1} and X_d . The class separation information underlying $X_1, X_2, \dots, X_d$ is captured by Z_d , and the separability measure on Z_d provides an approximation of the original Mahalanobis measure in the d -dimensional space. The merit of the recursive approach is that it reduces computational complexity from $O(d^3)$ to $O(d)$. This recursive approach is extremely efficient when it is combined with a forward search procedure. Take step j of a sequential forward feature selection procedure as an example. To identify the j th feature to be selected from a pool of $d - j + 1$ candidate features, the non-recursive method has to evaluate $d - j + 1$ Mahalanobis measures in j -dimensional space where matrix inverse has to resort to numerical methods such as Gaussian elimination. By contrast, the recursive approach evaluates $d + j - 1$ measures in 2-dimensional space, where matrix inverse is obtained analytically.

In addition, it is noted that Mahalanobis measure is saturated after a small number of features have been selected, if the small number of samples is small. However, the small number of features usually perform unsatisfactorily on unseen test data. This is because the feature subsets selected by Mahalanobis measure overfit the

4.2 Fast Feature Selection based on Regulated Recursive Mahalanobis Measure 64

small-sized training data. To alleviate over-fitting and to achieve the efficient selection of feature subsets with enhanced generalization capacity, a regularized recursive Mahalanobis measure is proposed, and guidelines on determination of regularization parameter are provided. Experimental studies reveal that the regularized recursive measure could produce feature subsets leading to substantially superior results to the non-regularized measure.

4.2 Fast Feature Selection based on Regulated Recursive Mahalanobis Measure

4.2.1 Recursive Mahalanobis Measure

Given a dataset D for a two-class problem with N samples,

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)$$

each of which contains d input features $x_{i,1}, x_{i,2}, \dots, x_{i,d}$ and one output class label y_i , where $y_i \in \{-1, +1\}$. We denote by X_j the j th input feature and denote by Y the output class label.

$$X_j = [x_{1,j}, x_{2,j}, \dots, x_{N,j}]^T$$

$$Y = [y_1, y_2, \dots, y_N]^T$$

4.2 Fast Feature Selection based on Regulated Recursive Mahalanobis Measure 65

where $x_{i,j}$ is the j th feature in i th sample.

If we assume that the distribution of samples in each class is Gaussian, the discriminative power underlying the d features can be evaluated using Mahalanobis class separability measure, defined as:

$$D = (\mathbf{m}_1 - \mathbf{m}_2)^T \Sigma^{-1} (\mathbf{m}_1 - \mathbf{m}_2) \quad (4.1)$$

where $\mathbf{m}_1$ and $\mathbf{m}_2$ are the mean vectors of class 1 and class 2, respectively. The mean vectors $\mathbf{m}_1$ and $\mathbf{m}_2$ can be estimated using sample means:

$$\mathbf{m}_l = \frac{1}{N_l} \sum_{\mathbf{x}_i \in \text{class } l} \mathbf{x}_i \quad (4.2)$$

where N_l is number of samples in class l ($l = 1, 2$). Σ is the covariance matrix of d features. Mahalanobis separability measure can be interpreted as the distance between points $\mathbf{m}_1$ and $\mathbf{m}_2$ in d -dimensional feature space, with correlations between d features removed. Mahalanobis measure can also be interpreted as a combination of three operators:

(i) finding a linear transform:

$$\mathbf{w} = \Sigma^{-1} (\mathbf{m}_1 - \mathbf{m}_2) \quad (4.3)$$

4.2 Fast Feature Selection based on Regulated Recursive Mahalanobis Measure 66

(ii) projecting data onto a new dimension Z (with $Z = [z_1, z_2, \dots, z_N]^T$):

$$z_i = \mathbf{x}_i^T \mathbf{w}$$

(iii) calculating difference of means of two classes in dimension Z :

$$\begin{aligned} D &= \frac{1}{N_1} \sum_{i \in \text{class } 1} z_i - \frac{1}{N_2} \sum_{i \in \text{class } 2} z_i \\ &= \frac{1}{N_1} \sum_{i \in \text{class } 1} \mathbf{x}_i^T \mathbf{w} - \frac{1}{N_2} \sum_{i \in \text{class } 2} \mathbf{x}_i^T \mathbf{w} \\ &= (\mathbf{m}_1 - \mathbf{m}_2)^T \mathbf{w} \\ &= (\mathbf{m}_1 - \mathbf{m}_2)^T \Sigma^{-1} (\mathbf{m}_1 - \mathbf{m}_2) \end{aligned}$$

The computational complexity involved in the Mahalanobis measure is mainly attributed to the linear transform $\mathbf{w}$. The transform in Eq. (4.1) is obtained directly in d -dimensional space and is a global optimal transform in the sense that it maximizes the ratio of inter-class difference to intra-class scatter.

In this chapter, we propose a recursive approach to find this transform through successive step-wise optimizations in 2-dimensional space. A new dimension, named Z_2 , is first created by optimizing the ratio of inter-class difference to intra-class scatter in the 2-dimensional space X_1 and X_2 . This new dimension captures major class separation information underlying X_1 and X_2 . Dimension Z_2 is then combined with X_3 to create new dimension Z_3 . This procedure is continued until Z_n is created.

Since Z_d retains major discriminative power underlying $X_1, X_2, \dots, X_d$, the

4.2 Fast Feature Selection based on Regulated Recursive Mahalanobis Measure 67

mean difference of two classes on Z_d provides a good measure of class separability therein. This procedure is summarized below:

- (i) Normalize each of the input features to zero mean and unit deviation.
- (ii) Define the first dimension X_1 as Z_1 , and set $j = 2$.
- (iii) Create new dimension Z_j based on Z_{j-1} and X_j using the following transform

$$Z_j = [Z_{j-1}, X_j] \mathbf{w}_j \quad (4.4)$$

Let δ_X be the mean differences of two classes in dimension X . Let Σ_j be the covariance matrix of Z_{j-1} and X_j . The transform $\mathbf{w}_j$ is defined as:

$$\mathbf{w}_j = \Sigma_j^{-1} [\delta_{Z_{j-1}}, \delta_{X_j}]^T \quad (4.5)$$

- (iv) Set $j = j + 1$, and repeat (iii) until new dimension Z_d is created.
- (v) Calculate difference of means of two classes on Z_d , yielding:

$$D_d = [\delta_{Z_{d-1}}, \delta_{X_d}] \Sigma_d^{-1} [\delta_{Z_{d-1}}, \delta_{X_d}]^T \quad (4.6)$$

Since $\mathbf{w}_d$ (Eq. (4.5) for $j = d$) is obtained through successive step-wise optimization, it is an suboptimal approximation of the optimal $\mathbf{w}$ shown in Eq. (4.3). However, suboptimality is not a major issue here because feature subsets selected by the optimal transform $\mathbf{w}$ often overfit small-sized training data and generalize

4.2 Fast Feature Selection based on Regulated Recursive Mahalanobis Measure 68

unsatisfactorily on unseen test data. The optimality of $\mathbf{w}$ is often deliberately degraded through regularization to enhance generalization capacity of the features selected (details refer to Section 4.2.3). The merit of the recursive approach is its computational efficiency as analyzed below.

If both dimensions of the data is normalized to zero mean and unit deviation, and the correlation between Z_{j-1} and X_j is denoted by ρ_j , we then have:

$$\Sigma_j^{-1} = \begin{bmatrix} 1 & \rho_j \\ \rho_j & 1 \end{bmatrix}^{-1} = \frac{1}{1 - \rho_j^2} \begin{bmatrix} 1 & -\rho_j \\ -\rho_j & 1 \end{bmatrix}$$

Thus, $\mathbf{w}_j$ can be written as:

$$\mathbf{w}_j = \frac{1}{1 - \rho_j^2} [\delta_{Z_{j-1}} - \rho_j \cdot \delta_{X_j}, \quad \delta_{X_j} - \rho_j \cdot \delta_{Z_{j-1}}]^T \quad (4.7)$$

The major computations involved at step j include new dimension creation, sample projection and normalization. These calculations demand $3N$, $3N$ and $2N$ basic operations, (*i.e.* addition, subtraction and multiplication etc) respectively, where N is the number of samples. Thus, the recursive evaluation of Mahalanobis measure in d -dimensional space demands approximately $8 \times N \times d$ operations. By contrast, non-recursive evaluation of Mahalanobis measure demands $(d^3 - d)/3 + N \times d^2 + 2 \times d^2$ operations, where $N \times d^2$, $(d^3 - d)/3$ and $2 \times d^2$ are the operations involved in covariance matrix Σ formation, inversion of matrix Σ by Gaussian elimination, and calculation of $\Sigma^{-1}(\mathbf{m}_1 - \mathbf{m}_2)$ respectively. It is apparent that the recursive approach achieves a massive reduction in the computational complexity, particularly when the

4.2 Fast Feature Selection based on Regulated Recursive Mahalanobis Measure 69

dimension d is much greater than the number of samples N , which is the case in gene expression data.

4.2.2 Forward Selection Based on Recursive Mahalanobis Measure

A sequential forward selection algorithm builds up the feature subset step by step, by adding one feature at a step. At step j , the objective is to identify the next, *i.e.* j^{th} , feature to be selected from a pool of $d - j + 1$ candidate features. Each of the $d - j + 1$ features in the candidate feature set is temporarily added to the feature subset selected at previous steps, and the corresponding separability measure is evaluated. This selection process evaluates $d - j + 1$ Mahalanobis measure in j -dimensional space, and therefore involves $(d - j + 1) \times (j^3 - j)/3 + N \times j^2 + 2 \times (d - j + 1) \times j^2$ operations. Note this complexity does not take into account the computations involved in forming matrix Σ_j , assuming correlations between features are calculated beforehand.

The complexity could be substantially reduced if the recursive approach is used to evaluate Mahalanobis measure. To select the j^{th} feature, each of the $d - j + 1$ features in the candidate feature subset is combined with Z_{j-1} and its significance is evaluated in 2-dimensional space.

The number of operations involved is approximately $8 \times N \times (d - j + 1)$, which is much lower than $(d - j + 1) \times (j^3 - j)/3 + 2 \times (d - j + 1) \times j^2$. For illustration, we take an gene expression data with 100 samples and 3000 features (genes) as an example.

4.2 Fast Feature Selection based on Regulated Recursive Mahalanobis Measure 70

The operations involved at different steps of a forward feature selection algorithm using non-recursive and recursive Mahalanobis measure respectively are shown in Figure 4.1. It is observed that the complexity of the non-recursive measure-based forward selection increases very rapidly as the selection progresses. However, the complexity of the recursive measure-based forward selection is almost fixed at all selection steps. The computation involved in the recursive approach is still moderate even if a large subset of features are to be selected from a huge feature set.

Figure 4.1: Computational complexity at different selection step

The procedure of forward selection of feature subset based on recursive Mahalanobis measure is summarized below:

(i) Initialization.

Set the selected feature subset S to empty and the candidate feature subset U to full set. Let $X_{(j)}$ denote the j th step optimal feature selected. Normalize

4.2 Fast Feature Selection based on Regulated Recursive Mahalanobis Measure 71

each of the input features to zero mean and unit deviation.

(ii) Select first feature.

Evaluate the mean difference between two classes in dimension of each feature $X_1, X_2, \dots, X_d$. Let $X_{(1)}$ be the feature with maximum mean difference. Add $X_{(1)}$ into feature subset S and delete $X_{(1)}$ from the candidate feature subset U . Let $Z_1 = X_{(1)}$ and $j = 2$.

(iii) Select the j th feature ($j > 1$).

Temporarily combine each of the features in candidate feature subset U with Z_{j-1} and calculate the corresponding Mahalanobis class separability measure using Eq. (4.8):

$$D_j(X_k) = \frac{\delta_{Z_{j-1}}^2 + \delta_{X_k}^2 - 2\rho_k \delta_{Z_{j-1}} \delta_{X_k}}{1 - \rho_k^2} \quad (4.8)$$

where X_k represents the k th feature in candidate feature subset U and ρ_k is denoted as the correlation coefficient between Z_{j-1} and X_k . Identify the feature with the maximum separability measure

$$X_{(j)} = \arg \max_k D_j(X_k) \quad (4.9)$$

Add $X_{(j)}$ to selected feature subset S and delete $X_{(j)}$ from candidate feature subset U . Create new dimension Z_j based on Z_{j-1} and $X_{(j)}$:

$$Z_j = [Z_{j-1}, X_{(j)}] \mathbf{w}_j \quad (4.10)$$

4.2 Fast Feature Selection based on Regulated Recursive Mahalanobis Measure 72

The transform $\mathbf{w}_j$ in this step is defined by

$$\mathbf{w}_j = \Sigma_j^{-1}[\delta_{Z_{j-1}}, \delta_{X_{(j)}}]^T \quad (4.11)$$

where Σ_j is the covariance matrix of variable Z_{j-1} and feature $X_{(j)}$. Normalize variable Z_j to zero mean and unit deviation. Let $j = j + 1$.

- (iv) repeat (iii) until the stopping criterion, say the number of features selected, is met.

4.2.3 Regularized Recursive Mahalanobis Measure

We found in experiments on gene expression problems that contain small number of samples, the original Mahalanobis measure and the recursive Mahalanobis measure become saturated after a small number of features have been selected, and the small number of features could classify training data perfectly well. When tested on unseen data, however, the small number of features often produce undesirable results. This is the typical phenomenon of overfitting. Overfitting is unlikely to occur in Mahalanobis measure-based feature selection for conventional data, and it therefore has not received many attentions. But Overfitting is indeed a severe problem in feature selection because of the very small sample size and very high dimensionality, as in gene expression data.

In classification and regression problems, regularization technique (Tikhonov and Arsenin, 1977) is often used to alleviate or prevent overfitting problem. Regular-

4.2 Fast Feature Selection based on Regulated Recursive Mahalanobis Measure 73

ization might also be used here to address the overfitting problem occurred in Mahalanobis measure-based feature selection. But how to apply it to Mahalanobis separability measure is the major issue.

As analyzed in Section 4.2.1, Mahalanobis distance separability measure can be interpreted as the distance between two mean vectors of two classes after being transformed by $\mathbf{w}$. Given Z_{j-1} and X_j , the transform Eq. (4.10) is optimal in the sense that it maximizes the ratio of inter-class difference to intra-class scatter. It is this optimality that causes the overfitting problem. We may alleviate the overfitting problem through deterring the optimality.

In classification and regression problems, regularization is often realized through limiting the amplitude of the parameters in models. This idea is adopted in the regularized recursive Mahalanobis measure in this study. Assume candidate feature $X_{(j)}$ is selected at step j , transform $\mathbf{w}_j$ is regularized as follows:

$$\mathbf{w}_j = \left[\frac{\delta_{Z_{j-1}} - \rho_j \times \delta_{X_{(j)}}}{1 - \rho_j^2 + \lambda_1}, \frac{\delta_{X_{(j)}} - \rho_j \times \delta_{Z_{j-1}}}{1 - \rho_j^2 + \lambda_1} \right]^T \quad (4.12)$$

where $\lambda_1 > 0$ is the regularization parameter. Obviously, a positive λ_1 could reduce the amplitude of $\mathbf{w}_j$. In the next step, *i.e.* $(j+1)$ th step, the regularized recursive Mahalanobis separability measure, D_{j+1} is given by:

$$D_{j+1}(X_k) = \frac{\delta_{Z_j}^2 + \delta_{X_k}^2 - 2\rho_k \delta_{Z_j} \delta_{X_k}}{1 - \rho_k^2 + \lambda_1} \quad (4.13)$$

Geometrically, transform vector $\mathbf{w}_j$ is a line in two-dimensional space, denoted

4.2 Fast Feature Selection based on Regulated Recursive Mahalanobis Measure 74

by its length and direction. The regularization parameter introduced in Eq. (4.12) could regularizes the length, *i.e.* the norm of $\mathbf{w}_j$, but it has no effect on the direction of $\mathbf{w}_j$.

As described in step (iii) of the feature selection procedure, a new dimension is created and samples are projected onto this new dimension after a feature has been selected at each step. Regularizing the length or norm of $\mathbf{w}_j$ could effectively degrade optimality of $\mathbf{w}_j$. However, regularization of the norm of $\mathbf{w}_j$ has no effect on the sample projections on the new dimension. This is because sample projections on the new dimension are normalized to unit standard deviation in the feature selection procedure, and this normalization offsets the effect of regularization of the norm of $\mathbf{w}_j$. It is the direction of $\mathbf{w}_j$ rather than its length that matters in creation of new dimension and generation of sample projections on it.

Regularization of direction of $\mathbf{w}_j$ can be realized through reducing the amplitude of w_{j1} while keeping w_{j2} remain unchanged. If $X_{(j)}$ is selected at step j , the new direction Z_j is created using the following regularized transform:

$$\mathbf{w}_j = \left[\frac{\delta_{Z_{j-1}} - \rho_j \times \delta_{X_{(j)}}}{1 - \rho_j^2 + \lambda_2}, \frac{\delta_{X_{(j)}} - \rho_j \times \delta_{Z_{j-1}}}{1 - \rho_j^2} \right]^T \quad (4.14)$$

where $\lambda_2 > 0$ is the regularization parameter. Taking Z_{j-1} as the reference axis, the angle of non-regularized $\mathbf{w}_j$ defined in Eq. (4.11) is given by:

$$\alpha = \arctan \left(\frac{\delta_{X_{(j)}} - \rho_j \times \delta_{Z_{j-1}}}{\delta_{Z_{j-1}} - \rho_j \times \delta_{X_{(j)}}} \right) \quad (4.15)$$

4.2 Fast Feature Selection based on Regulated Recursive Mahalanobis Measure 75

While the angle of the regularized $\mathbf{w}_j$ defined in Eq. (4.14) is given by:

$$\beta = \arctan \left(\frac{1 - \rho_j^2 + \lambda_2}{1 - \rho_j^2} \times \frac{\delta_{X(j)} - \rho_j \times \delta_{Z_{j-1}}}{\delta_{Z_{j-1}} - \rho_j \times \delta_{X(j)}} \right) \quad (4.16)$$

The angle becomes very small after a few features have been selected. Using the relationship $\arctan(\gamma) \approx \gamma$ for small γ , yields:

$$\beta \approx \left(1 + \frac{\lambda_2}{1 - \rho_j^2} \right) \alpha \quad (4.17)$$

By setting λ_2 to different values, different extends of regularization will be obtained.

Regularization technique is used twice in each step of the feature selection procedure. λ_1 regularizes the norm of $\mathbf{w}_j$ and is used in feature evaluation and selection, while λ_2 regularizes the direction of $\mathbf{w}_j$ and is employed in generation of sample projections on new dimension after a feature has been selected. Suitable values of λ_1 and λ_2 are problem dependent. As regularization parameters, suitable values of λ_1 and λ_2 could always be determined through cross validation, just like in other applications of the regularization technique. Next, we consider two special values of λ_1 and λ_2 .

If the regularization parameter λ_1 is set as:

$$\lambda_1 = \rho_j^2 \quad (4.18)$$

4.2 Fast Feature Selection based on Regulated Recursive Mahalanobis Measure 76

then the regularized recursive Mahalanobis measure in Eq. (4.13) is simplified to:

$$D_j(X_k) = \delta_{Z_{j-1}}^2 + \delta_{X_k}^2 - 2\rho_k \delta_{Z_{j-1}} \delta_{X_k} \quad (4.19)$$

If the regularization parameter λ_2 is set as:

$$\lambda_2 = 1 - \rho_j^2 \quad (4.20)$$

then the regularized transform has an angle of

$$\beta \approx 2\alpha \quad (4.21)$$

In other words, the regularization results in a rotation of angle α . Eq. (4.17) is an approximated result, and the actual angle of regularized $\mathbf{w}_j$ is slightly smaller. Thus, a slightly greater λ_2 should be used, in order to obtain an actual rotation of α :

$$\lambda_2 = 1.05 \times (1 - \rho_j^2) \quad (4.22)$$

Our experimental studies show that regularization parameters given in Eq. (4.18) and Eq. (4.22) produces excellent results. Throughout our experiments in next section, the two regularization parameters are set as in Eq. (4.18) and Eq. (4.22) respectively.

4.3 Experimental Studies

The performance of a feature selection algorithm is usually assessed in terms of its computational efficiency and classification accuracy. As analyzed in Section 4.2.2, the forward feature selection algorithm employing recursive Mahalanobis measure has a much lower complexity than that employing non-recursive Mahalanobis measure. The recursive Mahalanobis measure-based feature selection algorithm also has much lower complexity than algorithms that employ classifier-dependent measures such as the recursive feature elimination (RFE) algorithm (Guyon et al., 2002). Since the computational superiority of the recursive Mahalanobis measure is obvious, it is not demonstrated in the experiment. The proposed algorithm is tested in terms of classification accuracy only.

Since there are many feature selection algorithms in the literature, it is impractical to compare our work with all of them. In the experimental study, the recursive Mahalanobis (RM) measure and the regularized recursive Mahalanobis (RRM) measure are compared only with the original Mahalanobis (M) measure and the recursive feature elimination (RFE) algorithm (Guyon et al., 2002), which is often considered as a benchmark due to its good performance.

A few estimators may be used to estimate error rate. These include leave-one-out error, repeated K -fold cross validation error and bootstrap error etc. The study in Braga-Neto and Dougherty (2004a) revealed that error estimation based on cross validation including leave-one-out and repeated K -fold cross validation may exhibit excessive variability. In this study, .632+ bootstrapping (Efron and Tibshirani,

1997) was used. In the bootstrap testing, 200 replica were generated to estimate the error rate, and the splits of training and test data in the 200 replica were kept identical during the testing of M measure, RM measure, RRM measure and RFE.

Eight gene expression problems were used to test the performance of the proposed algorithm. The eight problems are summarized in Table 4.1. Every feature in each of these datasets was standardized as as to possess a zero mean and unit deviation. Since the dimensionality (*i.e.* the number of features) of gene expression data is very high, and most of these features are irrelevant to the discriminant task, a pre-selection procedure was employed to reduce the number of candidate features to 1000 based on Fisher's ratio, which is an individual feature ranking criterion. All the experiments and comparisons in this work were conducted on the pre-selected data.

Datasets	Original sources	Genes	Samples
Leukaemia	Golub et al. (1999)	7129	72
Breast cancer (ER)	West et al. (2001)	7129	49
Breast cancer (LN)	West et al. (2001)	7129	49
Lung cancer	Gordon et al. (2002)	12533	181
CNS tumour	Pomeroy et al. (2002)	7129	60
Prostate cancer	Singh et al. (2002)	12600	136
Breast cancer	van't Veer et al. (2002)	24481	97
Colon Cancer	Alon et al. (1999)	2000	62

Table 4.1: Datasets Description

Figures 4.2-4.9 are the results when the regularization parameters λ_1 and λ_2 in the RRM measure were set as in Eq. (4.18) and Eq. (4.22) in feature evaluation and in sample projection respectively. The weight on slack variable of the support vector machine in RFE was set to 0.05. In the experimental study, a wide range

of values, as small as 0.001 and as great as 100, were used for the weight, but the results were almost identical.

Figures 4.2-4.8 show that the RM measure and M measure produce comparable results, with M measures slightly outperforming in some of the problems, while RM measure slightly outperforming in others. However, the RRM measure outperforms both M measure and RM measure. This is because the regularization introduced in RRM alleviated the overfitting problem caused by the small sample size of feature expression data, and enhanced the generalization capacity of the feature subsets selected.

It was widely accepted that classifier-dependent measures outperform classifier-independent measures because classifier-dependent measures select features that are most favoured by the classifier. However, the experimental results shown in Figures 4.2-4.9 indicate this is not true in feature selection for gene expression data. Among the 8 problems, RFE algorithm produced inferior results to RRM in 7 problems. We believe this is because the feature subsets selected by the classifier-dependent RFE tend to overfit the small-sized training sample, consequently, the feature subsets selected generalize unsatisfactorily on unseen test data. This is similar to the overfitting problem occurred in the non-regularized Mahalanobis measure.

4.4 Chapter Summary

In this chapter, we have proposed a recursive approach to Mahalanobis separability measure evaluation. Analysis has shown the recursive Mahalanobis measure has a

Figure 4.2: Comparison of RM and RRM with M and RFE in Leukaemia problem

Figure 4.3: Comparison of RM and RRM with M and RFE in Breast Cancer (ER) problem

Figure 4.4: Comparison of RM and RRM with M and RFE in Breast Cancer (LN) problem

Figure 4.5: Comparison of RM and RRM with M and RFE in Lung Cancer problem

Figure 4.6: Comparison of RM and RRM with M and RFE in CNS Tumor problem

Figure 4.7: Comparison of RM and RRM with M and RFE in Breast Cancer problem

much lower complexity than non-recursive measure. In addition, the regularization technique has been introduced to recursive Mahalanobis measure to alleviate the

Figure 4.8: Comparison of RM and RRM with M and RFE in Colon Cancer problem

Figure 4.9: Comparison of RM and RRM with M and RFE in Prostate Cancer problem

overfitting problem caused by the small sample size of gene expression data. Experimental studies have shown that the regularized recursive Mahalanobis measure produces improved results.

Chapter 5

Fast Feature selection Algorithm for Very High Dimensional Data Based on Regularized Correlation Criterion

5.1 Introduction

The correlation between a feature and class labels reflects the predictive power of the feature, and has been commonly used for individual feature evaluation. Due to the redundancy problem inherent in individual-based feature evaluation, set-based correlation criterion is the focus of this study. One such set-based criterion is the correlation heuristic proposed by Hall (1998), as defined in Eq. (3.6), which evaluates the merit of a feature subset based on the ratio of the average feature-class correlation to the average feature-feature correlation. The average feature-class correlation reflects the relevancy of a feature subset while the average feature-feature correla-

tion reflects the redundancy existing in the feature subset. Thus, maximization of the correlation heuristic (Hall, 1998) actually maximizes the relevancy and minimizes redundancy of the feature subset. Similar ideas were also adopted in (Ding and Peng, 2003; Peng et al., 2005). In this chapter, we propose a new approach to correlation-based feature subset evaluation, which is combined with sequential forward selection. At each selection step, the new approach evaluates the significance of a candidate feature to be added by first isolating the component that complements the selected feature subset, and then evaluating the feature based on the correlation between class labels and the isolated complementary component. The isolation of complementary component is done through orthogonal decomposition. Instead of orthogonalizing a candidate feature with each of the features in the subset, the new approach orthogonalizes the candidate feature to the class label prediction built upon the feature subset. The merit of this approach is that it retains the redundancy handling capacity of orthogonal decomposition but demands less computation than conventional orthogonal decomposition.

Due to small sample size and high dimensionality of gene expression data, overfitting could easily occur in feature selection. The regularization technique is introduced and a regularized correlation criterion is derived. Experimental studies on seven gene expression problems show that the regularized correlation measure produces feature subsets leading to excellent classification performance.

5.2 Regularized Correlation Measure for Feature Selection

5.2.1 A New Correlation-based Approach to Feature Evaluation

Given a dataset D for a two-class problem with N samples,

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)$$

each of which contains d input features $x_{i,1}, x_{i,2}, \dots, x_{i,d}$ and one output class label y_i with $y_i \in \{-1, +1\}$. We denote by X_j the j th input feature and denote by Y the output class labels.

$$X_j = [x_{1,j}, x_{2,j}, \dots, x_{N,j}]^T$$

$$Y = [y_1, y_2, \dots, y_N]^T$$

The correlation between a feature and class labels reflects the predictive power of the feature. The Pearson correlation coefficient is estimated as:

$$R_{YX_j} = \frac{\sum_{i=1}^N (x_{i,j} - \bar{x}_j)(y_i - \bar{y})}{\sqrt{\sum_{i=1}^N (x_{i,j} - \bar{x}_j)^2 \sum_{i=1}^N (y_i - \bar{y})^2}} \quad (5.1)$$

where the bar notation stands for an average over the index i . After features and

class labels are normalized to zero mean and unit standard deviation, R_{YX_j} defined in Eq. (5.1) can be rewrite as:

$$R_{YX_j} = \frac{X_j^T Y}{N - 1} \quad (5.2)$$

This correlation coefficient evaluates features on an individual basis. It is generally acknowledged that a collection of individually predictive features does not necessarily produce satisfactory classification when they work collectively in a pattern classifier. This is because severe redundancy often exists in the top-ranked individually predictive features. For efficient pattern classification or class prediction, features should be evaluated and selected on a set basis rather than an individual basis.

Next, we propose a new approach to correlation-based feature subset evaluation based on sequential forward selection. It is assumed that a feature subset S_m consisting of m features has already been selected at previous steps of a forward feature selection procedure, where $m \geq 1$. The new approach first project the data from m -dimensional space to a new one-dimensional space corresponding to a variable Z_m , using the linear support vector machine (SVM), where Z_m is a column vector with N elements as given below.

$$Z_m = [z_{1,m}, z_{2,m}, \dots, z_{N,m}]^T$$

Since the linear SVM classifier captures the major discriminative power under-

lying S_m , $z_{i,m}$ can be regarded as the class label prediction of sample $\mathbf{x}_i$ built upon given feature subset S_m . To identify the next feature to be added to S_m , each of the features in the candidate feature set should be evaluated. For candidate feature X_j , the information underlying X_j is first decomposed into two parts:

$$X_j = U_j + \alpha Z_m \quad (5.3)$$

where α is the projection operator defined by

$$\alpha = \frac{X_j^T Z_m}{Z_m^T Z_m} \quad (5.4)$$

Eq. (5.3) is actually the Gram-Schmidt orthogonal decomposition (see, for example Trefethen and Bau, 1997), and it can be easily shown that U_j is orthogonal to Z_m :

$$\begin{aligned} U_j^T Z_m &= (X_j - \alpha Z_m)^T Z_m \\ &= X_j^T Z_m - Z_m^T Z_m \frac{X_j^T Z_m}{Z_m^T Z_m} \\ &= 0 \end{aligned} \quad (5.5)$$

The orthogonality between U_j and Z_m means that U_j is the part of information that is complement to Z_m . Since Z_m is the class label prediction built upon S_m , it should capture the major predictive power underlying feature subset S_m . Thus, U_j can be roughly considered as the part of information complementary to S_m . Thus, given a previously selected feature subset S_m , the significance of candidate feature X_j can

be evaluated in terms of the correlation between U_j and class label Y .

The new approach isolates the complementary component by orthogonalizing X_j to Z_m instead of each of the features in S_m . The strength of this decomposition is that it demands less computation than conventional orthogonal decomposition meanwhile it retains the redundancy handling capacity of orthogonal decomposition. To select the $(m+1)^{th}$ feature from a candidate feature set of $d-m$ features at step $m+1$, the computations involved just include training a linear classifier once and performing $d-m$ vector products in N -dimensional space, where N is the training sample size.

Assume X_j , Y and Z_m are all normalized to zero mean and unit standard deviation. The correlation between U_i and Y is given by:

$$\begin{aligned}
 R_{YU_j} &= \frac{1}{\sigma_{U_j}} \frac{U_j^T Y}{N-1} \\
 &= \frac{1}{\sigma_{U_j}} \left(\frac{X_j^T Y}{N-1} - \alpha \frac{Z_m^T Y}{N-1} \right) \\
 &= \frac{1}{\sigma_{U_j}} (R_{YU_j} - \alpha R_{YZ_m})
 \end{aligned} \tag{5.6}$$

where R_{YZ_m} denotes the correlation coefficient between Y and Z_m , and σ_{U_j} is the

standard deviation of U_j given by:

$$\begin{aligned}
 \sigma_{U_j} &= \sqrt{\frac{U_j^T U_j}{N-1}} \\
 &= \sqrt{\frac{1}{N-1} (X_j - \alpha Z_m)^T (X_j - \alpha Z_m)} \\
 &= \sqrt{\frac{X_j^T X_j}{N-1} - 2\alpha \frac{X_j^T Z_m}{N-1} + \alpha^2 \frac{Z_m^T Z_m}{N-1}} \\
 &= \sqrt{1 - 2\alpha R_{Z_m X_j} + \alpha^2} \tag{5.7}
 \end{aligned}$$

where $R_{Z_m X_j}$ denotes the correlation between X_j and Z_m . Since Z_m is normalized to unit standard deviation, the α in Eq. (5.4) can be written as:

$$\alpha = \frac{X_j^T Z_m}{N-1} = R_{Z_m X_j} \tag{5.8}$$

Substituting Eq. (5.8) into Eq. (5.7), yields:

$$\sigma_{U_j} = \sqrt{1 - R_{Z_m X_j}^2} \tag{5.9}$$

Thus, the correlation coefficient $R_{Y U_j}$ is obtained as:

$$R_{Y U_j} = \frac{R_{Y X_j} - R_{Z_m X_j} R_{Y Z_m}}{\sqrt{1 - R_{Z_m X_j}^2}} \tag{5.10}$$

Eq. (5.10) provides a significance measure of feature X_j subject to a previously selected feature subset S_m . $R_{Y U_j}$ can be considered as the semi-partial correlation between Y and X_j holding Z_m constant for Y . But note the meaning of this cor-

relation measure is different from that of the conventional semi-partial correlation because Z_m carries the predictive power of a set of variables rather than a single independent variable.

5.2.2 Regularized Correlation Measure

For very high dimensional data with related small sample size, such as gene expression data, overfitting could easily occur in feature selection. When overfitting happens, the feature subsets selected perform perfectly on training data but badly on test data. In pattern classification and regression, regularization technique is often employed to alleviate overfitting problem. It is expected that the regularization technique is also able to alleviate the overfitting problem in feature selection. But the issue here is how to introduce regularization to the correlation measure Eq. (5.10).

In classification and regression, regularization is often realized by adding an extra term to the cost function to penalize the complexity of the solution. The regularized solution is no longer optimal to the original optimization problem. In a sense, regularization induces sub-optimality to the solution obtained. But note this sub-optimality is with respect to training data. Regularization actually provides a balanced solution between training data and test data. In this sense, regularization can be interpreted as an operation that induces sub-optimality to the solution on training data.

For the decomposition Eq. (5.3), the α given in Eq. (5.4) can be regarded as the

optimal solution in the sense that it produces orthogonal vector to Z_m . Any other values that do not produce orthogonal vectors should be regarded as sub-optimal solutions. Considering orthogonality as optimality in Eq. (5.3) and regularization as inducing sub-optimality, we propose to introduce regularization to the correlation measure Eq. (5.10) through deliberately deteriorating the orthogonality between U_j and Z_m as follows:

$$\begin{aligned} V_j &= X_j - (\alpha + \beta)Z_m \\ &= U_j - \beta Z_m \end{aligned} \quad (5.11)$$

where β is the regularization parameter. The regularized correlation measure is given by:

$$\begin{aligned} R_{YV_j} &= \frac{1}{\sigma_{V_j}} \frac{V_j^T Y}{N-1} \\ &= \frac{1}{\sigma_{V_j}} \left(\frac{U_j^T Y}{N-1} - \beta \frac{Z_m^T Y}{N-1} \right) \\ &= \frac{1}{\sigma_{V_j}} (\sigma_{U_j} R_{YU_j} - \beta R_{YZ_m}) \\ &= \frac{1}{\sigma_{V_j}} [R_{YX_j} - (\beta + R_{Z_m X_j}) R_{YZ_m}] \end{aligned} \quad (5.12)$$

where σ_{V_j} is the standard deviation of V_j given by:

$$\begin{aligned}
 \sigma_{V_j} &= \sqrt{\frac{V_j^T V_j}{N-1}} \\
 &= \sqrt{\frac{1}{N-1} (U_j - \beta Z_m)^T (U_j - \beta Z_m)} \\
 &= \sqrt{\frac{U_j^T U_j}{N-1} + \beta^2 \frac{Z_m^T Z_m}{N-1}} \\
 &= \sqrt{1 - R_{Z_m X_j}^2 + \beta^2}
 \end{aligned} \tag{5.13}$$

Substituting Eq. (5.13) into Eq. (5.12), yields:

$$R_{Y V_i} = \frac{R_{Y X_i} - (\beta + R_{Z_m X_i}) R_{Y Z_m}}{\sqrt{1 - R_{Z_m X_i}^2 + \beta^2}} \tag{5.14}$$

Eq. (5.14) is the regularized set-based correlation measure. Suitable values of the regularization parameter β are problem dependent. As a regularization parameter, β could always be determined through cross validation, just like in other applications of the regularization technique. If β is set as follows:

$$\beta = R_{Z_m X_j} \tag{5.15}$$

the regularized correlation measure becomes:

$$R_{Y V_j} = R_{Y X_j} - 2R_{Z_m X_j} R_{Y Z_m} \tag{5.16}$$

Apparently, the setting of regularization parameter in Eq. (5.15) simplifies the

regularized correlation measure to a very concise form. The effect of regularization is the creation of non-orthogonal vector V_j . Geometrically, the effect of regularization can be interpreted as rotating vector U_j to V_j , and the extent of regularization can be assessed in terms of the rotation angle. Assume the angle between V_j and U_j is denoted by θ , we have:

$$\begin{aligned}
 \cos(\theta) &= \frac{U_j^T V_j}{\|U_j\| \times \|V_j\|} \\
 &= \frac{U_j^T U_j}{\|U_j\| \times \|V_j\|} \\
 &= \frac{\|U_j\|}{\|V_j\|} \\
 &= \frac{\sigma_{U_j}}{\sigma_{V_j}} \\
 &= \sqrt{\frac{1 - R_{Z_m X_j}^2}{1 - R_{Z_m X_j}^2 + \beta^2}} \\
 &= \sqrt{1 - \frac{\beta^2}{1 - R_{Z_m X_j}^2 + \beta^2}}
 \end{aligned} \tag{5.17}$$

If β is set as in Eq. (5.15), the angle is given by:

$$\theta = \arccos \left(\sqrt{1 - R_{Z_m X_i}^2} \right) \tag{5.18}$$

To illustrate the effect of the regularization parameter, we plot the following function:

$$\theta = \arccos \left(\sqrt{1 - \gamma^2} \right) \tag{5.19}$$

where $\gamma > 0$ is a real number.

This means that feature with higher correlation with class labels will be regu-

Figure 5.1: Rotation angle versus gamma

larized at higher extent. In other words, the regularized correlation measure is in favor of features with high relevancy. Experimental studies show that the regularization parameter setting in Eq. (5.15) produces feature subsets leading to excellent classification results.

5.2.3 Generation of Z_m

At the beginning of step $m + 1$, Z_m has to be produced based on S_m first. $Z_m = [z_{1,m}, \dots, z_{N,m}]^T$ can be obtained using various classifiers such as support vector machines (SVMs), Fisher's linear discriminant analysis (FLD) and linear least mean square (LMS) estimator etc built upon S_m . Due to small sample size and high dimensionality of gene expression data, however, it is noted that $z_{i,m+1}$ ($i = 1, \dots, N$) is almost equal to the class label y_i after a small number of features are selected, no

matter what classifier is used. This can be considered as an overfitting problem. To alleviate this problem, two actions are taken in this study.

First, $z_{i,m}$ is produced using $z_{i,m-1}$ and the newly selected feature $x_{i,j}$, instead of S_m , as shown below:

$$z_{i,m} = w_1 z_{i,m-1} + w_2 x_{i,j} \quad (5.20)$$

where w_1 and w_2 are the coefficients. Eq. (5.20) is just a sub-optimal prediction of y_i compared with the prediction based on S_m . But this sub-optimality is intended here to alleviate the overfitting.

Second, a regularization parameter is introduced to the class label predictor Eq. (5.20). The linear predictor can be interpreted as a projection that maps data points in the two-dimensional space of X_j and Z_m onto a line whose direction is determined by coefficients w_1 and w_2 . The class label prediction can be regularized by rotating the line. Taking Z_m as a reference, the angle of the line before rotation, denoted by ϕ , is given by:

$$\phi = \arctan\left(\frac{w_2}{w_1}\right) \quad (5.21)$$

Rotation of the line can be realized by reducing the value of w_1 while keeping w_2 remain unchanged:

$$w_1' = \frac{w_1}{\lambda} \quad (5.22)$$

$$w_2' = w_2 \quad (5.23)$$

Thus, the angle of the line after rotation is:

$$\phi' = \arctan \left(\lambda \frac{w_2}{w_1} \right) \quad (5.24)$$

After a few features are selected, $z_{i,m}$ are very close to class label y_i , and the value of w_1 has a much greater amplitude than w_2 . Thus,

$$\phi \approx \frac{w_2}{w_1} \quad (5.25)$$

$$\phi' \approx \lambda \frac{w_2}{w_1} \quad (5.26)$$

Suitable values for the regularization parameter λ are problem dependent, and could be found through cross validation. If λ is set as:

$$\lambda = 2 \quad (5.27)$$

The angle of the regularized line is twice of the original angle. Experimental studies show that the setting Eq. (5.27) produces satisfactory results. Thus, the regularized class label prediction is given by:

$$z_{i,m} = \frac{1}{2} w_1 z_{i,m-1} + w_2 x_{i,j} \quad (5.28)$$

$z_{i,m}$ is based on $z_{i,m-1}$ and $x_{i,j}$. Besides its capacity to help alleviate the overfitting problem, this method is recursive in nature. Calculations in 2-dimensional space

is computationally much efficient than in high-dimensional space. In addition, the selection of regularization parameter in 2-dimensional space is much easier than in high-dimensional space.

5.2.4 Procedure of Regularized Correlation-based Sequential Feature Selection

The procedure of sequential forward feature selection based on regularized semi-partial correlation measure is summarized below:

(i) Initialization.

Set the selected feature subset S to empty and the candidate feature subset U to full set. Denote $X_{(j)}$ as the j th step optimal feature selected. Normalize input features to zero mean and unit standard deviation.

(ii) Select first feature ($j = 1$).

Pearson correlation between class label and each of the d features in the candidate features set: U . Identify the feature that has the maximum correlation measure, i.e. $X_{(1)}$, add it to the selected feature subset $S_1 = \{X_{(1)}\}$ and remove it from the candidate feature subset U . Let $Z_1 = X_{(1)}$ and $j = j + 1$.

(iii) Select the j th feature ($j > 1$).

Evaluate the regularized correlation criterion between Y and each of the $n - j + 1$ features in the candidate feature set using Eq. (5.16). Identify the feature having the maximum measure, denoted as $X_{(j)}$, add it to the feature subset

$S_j = \{S_{j-1}, X_{(j)}\}$, and remove it from the candidate feature set U .

Update the parameters w_1 and w_2 based on Z_{j-1} and the newly selected feature X_j using SVMs, FLD or LMS estimator. Regularize w_1 and w_2 using Equations (5.22) - (5.23), produce Z_j and normalize Z_j to zero mean and unit standard deviation. Let $j = j + 1$.

- (iv) Repeat step (iii) until a stopping criterion, say the number of feature selected, is satisfied.

In this procedure, the major computations for selecting the j^{th} feature include evaluating $(d - j + 1)$ candidate features one by one in two-dimensional feature space, creating and normalizing the new dimension. For a dataset with N samples, the computations at j^{th} step involved are $6N \times (d - j + 1)$, $3N$, $2N$ and N operations (i.e. addition, subtraction and multiplication) for evaluation, projection and normalization, respectively. In traditional feature evaluation, those candidate features could be temporally added in and evaluated in j -dimensional feature space, the computations of which could be much higher (see, Section 4.2.1).

5.3 Experimental studies

In the experiment, the performance of the proposed correlation heuristic was evaluated in terms of classification error rate. To estimate error rate, several error estimators are available including leave-one-out cross validation, repeated K -fold cross validation and bootstrap etc. The study in (Braga-Neto and Dougherty, 2004b)

revealed that error estimation based on cross validation including leave-one-out and repeated K -fold cross validation may exhibit excessive variability. In this study, .632+ bootstrapping (Efron and Tibshirani, 1997) was used. The error rate was estimated using the average over 200 replica and the splits of training and test data in the 200 replica were kept identical during the testing of the feature selection algorithms.

Six gene expression datasets were used to test the performance of the proposed algorithm. Each dataset contains more than seven thousands of features (genes), as described in Table 5.1.

Datasets	Original sources	Genes	Samples
Leukaemia	Golub et al. (1999)	7129	72
Breast cancer (ER)	West et al. (2001)	7129	49
Breast cancer (LN)	West et al. (2001)	7129	49
Lung cancer	Gordon et al. (2002)	12533	181
CNS tumour	Pomeroy et al. (2002)	7129	60
Breast cancer	van't Veer et al. (2002)	24481	97

Table 5.1: Datasets Description

Every feature of each of these datasets was standardized to be zero mean and unit deviation. Again, since the dimensionality (*i.e.* the number of genes) of gene expression data is very high, and most of these features are irrelevant to the discriminant task, top-1000 candidate features (with highest values of Fisher's ratio) was pre-selected and all the experiments and comparisons in this work were conducted on the pre-selected data.

Two regularization parameters, β and λ are introduced to the correlation-based feature selection algorithm. The former is introduced to regularize the correlation

measure, and the latter is introduced to regularize class label prediction. To study the effects of the two regularization parameters, we investigated the following 4 cases. (i) $\beta = \alpha$, $\lambda = 2$; (ii) $\beta = \alpha$, $\lambda = 0$; (iii) $\beta = 0$, $\lambda = 2$; (iv) $\beta = 0$, $\lambda = 0$, where α is defined in Eq. (5.8). The experimental results on the 6 datasets are shown in Fig. 5.2 to Fig. 5.7, respectively.

Across the 6 problems, the regularization technique produces improved results. In addition, it is noted from Figures 5.2-5.7 that the regularization parameter β play a more critical role than parameter λ .

Figure 5.2: Comparison of regularized correlation-based feature selection with different parameters in Leukaemia problem

The experimental results of the proposed methods in the previous Chapter and this Chapter are compared in Fig. 5.8 to Fig. 5.13. The comparison results show the performance of both methods are similar for the data problems (in Table 5.1). Both proposed methods employ the sequential forward selection algorithm and evaluate

Figure 5.3: Comparison of regularized correlation-based feature selection with different parameters in Breast Cancer (ER) problem

Figure 5.4: Comparison of regularized correlation-based feature selection with different parameters in Breast Cancer (LN) problem

Figure 5.5: Comparison of regularized correlation-based feature selection with different parameters in Lung Cancer problem

Figure 5.6: Comparison of regularized correlation-based feature selection with different parameters in CNS Tumor problem

Figure 5.7: Comparison in Breast Cancer problem

the goodness of feature subsets in a recursive fashion. The major difference between them is that the different evaluation criteria are used. The similar performance may show that the performance of Mahalanobis distance (in proposed method of Chapter 4) is consistent with that of the correlation criterion.

5.4 Chapter Summary

In this chapter, we have proposed a regularized correlation criterion for efficient feature subset selection. The regularized correlation criterion has been tested using 6 gene expression datasets. The experimental results show that regularization substantially improves the performance of the correlation-based feature selection algorithm for gene datasets. This reveals that regularization is very necessary in feature selection for gene expression data, which is usually with small sample size

Figure 5.8: Comparison of regularized correlation (RC) measure and regularized recursive Mahalanobis (RRM) measure (proposed in Chapter 4) in Leukaemia problem

Figure 5.9: Comparison of regularized correlation (RC) measure and regularized recursive Mahalanobis (RRM) measure (proposed in Chapter 4) in Breast Cancer (ER) problem

Figure 5.10: Comparison of regularized correlation (RC) measure and regularized recursive Mahalanobis (RRM) measure (proposed in Chapter 4) in Breast Cancer (LN) problem

and very high dimensionality.

Figure 5.11: Comparison of regularized correlation (RC) measure and regularized recursive Mahalanobis (RRM) measure (proposed in Chapter 4) in Lung Cancer problem

Figure 5.12: Comparison of regularized correlation (RC) measure and regularized recursive Mahalanobis (RRM) measure (proposed in Chapter 4) in CNS Tumor problem

Figure 5.13: Comparison of regularized correlation (RC) measure and regularized recursive Mahalanobis (RRM) measure (proposed in Chapter 4) in Breast Cancer problem

Chapter 6

A Novel Feature Selection

Algorithm For Mixed Data

6.1 Introduction

In real-world pattern recognition problems, data collected are usually contains features/attributes with different feature types. The feature with numeric values in a certain range is called continuous feature, such as AGE of customers that can be any value from 0 to 100 (years of age). While the feature with several possible values, each of which represents a mutually exclusive category, is called nominal features, such as MARRIAGE STATUS that has only three categories, i.e. *single*, *married* or *divorced*. It is easy to determine the certain distance between two patterns using a continuous feature, for example, one customer A that is 40 years' old is 5 years older than another customer B that is 35 years' old. However, differences or even order between *single*, *married* and *divorced* can not certainly be defined.

Due to the different natures of continuous and nominal features, how to make good use of information in both types of features are key to the success in machine

learning. In the literature, two categories of methods have been developed to evaluate subsets with mixed features: those adopting an early integration scheme and those adopting an intermediate integration scheme (Pavlidis et al., 2001).

Methods using the early integration scheme, such as the correlation-based feature selection (CFS) (Hall, 1998) and decision tree ID3 (Quinlan, 1986), preprocess the mixed data into single-typed data before learning, through either mapping nominal features to numeric scores or discretizing the continuous features into nominal ones. The feature-type transformation might cause loss of mutual information in nominal features or loss of order information in continuous features.

Methods adopting the intermediate integration scheme, such as hybrid criteria (Aha, 1992; Bar-Hen and Daudin, 1995; Wilson and Martinez, 1997) measure different types of features of a mixed feature subset using different metrics, whose resulting values are then combined to give an overall evaluation of the mixed feature set. Advantage of this category of methods is to avoid feature type transformation, but how to combine different metrics is a crucial problem because those measures from different metrics can hardly be unified.

Evaluating a subset with both continuous and nominal features are not the only difficulty in feature selection for mixed data. If we regard feature type in search algorithm, subsets generated in a certain selection step might be various, including subsets with only continuous features, subsets with only nominal features and/or subsets with both continuous and nominal features. Although comparing various subsets is a difficult task, it is unavoidable for the best feature subset of mixed data

should not be limited in subsets with both continuous and nominal features; it can also be subset with single-typed features, based on the particular machine learning problems.

These problems in feature selection for mixed data motivate our work in this chapter. Major contributions are in two folds. First, we propose a error probability measure for subset evaluation. This criterion that evaluates subset with both continuous and nominal features can detect the dependencies among features, without using feature type transformation. Secondly, we propose a search algorithm for mixed data using a filter-wrapper hybrid fashion, where subsets with same types are firstly evaluated by filter-liked criterion, and then the best subsets of different types are then compared by a wrapper-liked criterion to lead to an overall best subset.

6.2 Error Probability Measure for Mixed Feature Evaluation

6.2.1 Error Probability Measure

In pattern recognition, the ultimate objective is to learn a classifier which will classify unseen data with a low classification error rate. Thus, the error probability could serve as a measure in feature evaluation (see Devijver and Kittler, 1982, chapter 5).

In the feature space defined by a set of features $\mathbf{X}$, which could belong to one of C class y_c , $c = 1, 2, \dots, C$, the *error probability*, denoted as $P_e(\mathbf{X})$ is defined as:

$$P_e(\mathbf{X}) = \int_{\mathbf{X}} \left[1 - \max_c P(y_c | \mathbf{X}) \right] p(\mathbf{X}) d\mathbf{X} \quad (6.1)$$

In the feature space defined by a mixed feature set with both continuous and nominal features, i.e. $[\mathbf{X}, \mathbf{Z}]$, the error probability, $P_e(\mathbf{X}, \mathbf{Z})$, can be written as:

$$P_e(\mathbf{X}, \mathbf{Z}) = \sum_{\mathbf{Z}} \int_{\mathbf{X}} \left[1 - \max_c P(y_c | \mathbf{X}, \mathbf{Z}) \right] p(\mathbf{X}, \mathbf{Z}) d\mathbf{X} \quad (6.2)$$

Rearranging Eq.(6.2), yields:

$$\begin{aligned} P_e([\mathbf{X}, \mathbf{Z}]) &= 1 - \sum_{\mathbf{Z}} \int_{\mathbf{X}} \max_c p(y_c, \mathbf{X}, \mathbf{Z}) d\mathbf{X} \\ &= \sum_{\mathbf{Z}} p(\mathbf{Z}) - \sum_{\mathbf{Z}} \int_{\mathbf{X}} \max_c p(y_c, \mathbf{X} | \mathbf{Z}) p(\mathbf{Z}) d\mathbf{X} \\ &= \sum_{\mathbf{Z}} p(\mathbf{Z}) \left[1 - \int_{\mathbf{X}} \max_c p(y_c, \mathbf{X} | \mathbf{Z}) d\mathbf{X} \right] \\ &= \sum_{\mathbf{Z}} p(\mathbf{Z}) \int_{\mathbf{X}} \left[1 - \max_c P(y_c | \mathbf{X}, \mathbf{Z}) \right] p(\mathbf{X}) d\mathbf{X} \\ &= \sum_{\mathbf{Z}} p(\mathbf{Z}) P_e(\mathbf{X} | \mathbf{Z}) \end{aligned} \quad (6.3)$$

Eq.(6.3) shows that the error probability of a mixed feature set is the weighted sum of conditional error probabilities of continuous features given nominal features. For ease of representation, we define a multi-nominal variable $\mathbf{z}$ that contains M possible distinct values: $\{z_1, z_2, \dots, z_M\}$, each of which represents a value-combination of nominal features $\mathbf{Z}$. Thus, Eq.(6.3) can be rewritten as:

$$P_e([\mathbf{X}, \mathbf{Z}]) = \sum_{m=1}^M p(z_m) P_e(\mathbf{X}|z_m) \quad (6.4)$$

Given a training sample set with N samples, the maximum likelihood estimate of $p(z_m)$ is the ratio of the frequency of samples given the value-combination z_m , denoted as N_m , to the number of samples N , i.e.

$$\hat{p}(z_m) = \frac{N_m}{N} \quad (6.5)$$

Substituting Eq. (6.5) into Eq. (6.4), yields the error probability criterion for given training data set as below:

$$\widehat{P}_e([\mathbf{X}, \mathbf{Z}]) = \sum_m^M \frac{N_m}{N} \widehat{P}_e(\mathbf{X}|z_m) \quad (6.6)$$

Eq.(6.6) reveals that, to obtain $\widehat{P}_e([\mathbf{X}, \mathbf{Z}])$, the feature space could be firstly decomposed into a set of mutually exclusive feature subspaces based on the multi-nominal variable $\mathbf{z}$, and the conditional error probability $\widehat{P}_e(\mathbf{X}|z_m)$ could then be measured using the continuous features $\mathbf{X}$ in the corresponding feature subspace. The overall error probability for the entire mixed feature set could finally be obtained by linearly combining the conditional error probability of each subspace weighted by $\hat{p}(z_m)$. This evaluation method requires no transformation of feature types and takes the dependency among subsets of features into account.

Next, we will illustrate the decomposition of feature space and introduce the estimation of conditional error probability $P_e(\mathbf{X}|z_m)$.

6.2.2 Mixed Feature Space Decomposition

The principle of space decomposition could be illustrated using an example such as the heart disease prediction problem (Murphy and Aha, 1995). In this problem, the objective is to evaluate the merit of four features to discriminate between the presence and absence of heart disease. The variables and their values are listed in Table 6.1.

Variable	Type	Values
Thal	nominal	normal, fixed defect, reversible defect
Sex	nominal	male, female
Age	continuous	29 ~ 77
ST-depression	continuous	0 ~ 6.2

Table 6.1: Variables and Their Types and Values

In this example, the multi-nominal variable $\mathbf{z}$ contains all distinct value-combination of the nominal features appeared in the training data set as shown in Table 6.2.

values in $\mathbf{z}$ (Thal $\times$ Sex)	
z_1	normal AND female
z_2	normal AND male
z_3	fixed defect AND female
z_4	fixed defect AND male
z_5	reversible defect AND female
z_6	reversible defect AND male

Table 6.2: Distinct Values in Multi-Nominal Variable $\mathbf{z}$

The space is firstly decomposed along these distinct value-combinations of $\mathbf{z}$ and the merit of the four features is then evaluated in each subspace in terms of the continuous features “Age” and “ST-depression” as illustrated in Fig. 6.1, where the estimation for $P_e(\mathbf{X}|z)$ (in the last column of Fig. 6.1) will be introduced in the

6.2 Error Probability Measure for Mixed Feature Evaluation

116

next section. Finally, the evaluation criterion of the four features is calculated using Eq. (6.4).

<i>Thal</i>	<i>Sex</i>	<i>ST-depression & Age</i>	$P(z)$	$Pe(X z)$
Normal	Female		0.2741	0.1554
	Male		0.2889	0.3718
Fixed defect	Female	N.A.	N.A.	N.A.
	Male		0.0519	0.6786
Reversalble detect	Female		0.0481	0.2692
	Male		0.3370	0.3846

Figure 6.1: Decomposition of the feature space with features Thal (nominal), Sex (nominal), Age (continuous) and ST-depression (continuous).

When the number of the nominal features of a mixed feature subset is large, the number of samples in each subspace divided by nominal features could be small, and this in turn results in inaccurate probability estimation due to the low sample

density. To avoid fragmentation of feature space, in the heuristic search algorithm of this work as we will described in next section, each subspace generated in previous steps (defined by previously selected nominal features) is decomposed along the values of the “best” chosen nominal feature in the current step, denoted as Z_{best} , based on following rules:

1. if all the samples in the given subspace have the same class labels, no decomposition is performed by Z_{best} ;
2. if the number of samples in the given feature space is too small, no decomposition is performed by Z_{best} (The default value of small sample size is set to 20 in our experiments);
3. otherwise, the given subspace is decomposed along all possible values of Z_{best} , so that the error probabilities of all the subspaces generated could be measured. The subspaces generated which do not lead to lower error probability than that of the given subspace before decomposition are merged together. This rule assumes that the decomposition along the value of Z_{best} is redundant if it could not reduce error probability.

When the “best” chosen feature is continuous feature rather than nominal feature, the feature subspaces keep unchanged.

6.2.3 Error Probability Estimation

After the space decomposition, the error probability $P_e(\mathbf{X}|z_m)$ will be measured in each subspace defined by the continuous features $\mathbf{X}$. In the subspace corresponding to z_m , which contains a subset of patterns $\mathcal{D}_{z_m} = \{\mathbf{x}_i, y_i\}_{i=1}^{N_m}$, the error probability $\widehat{P}_e(\mathbf{X}|z_m)$ of hypothesis h could be the average of error probabilities over all the data points in this subspace as below:

$$\widehat{P}_e(\mathbf{X}|z_m) = \frac{1}{N_m} \sum_{i=1}^{N_m} \hat{P}[h(\mathbf{x}_i) \neq y_i|\mathbf{x}_i] = 1 - \frac{1}{N_m} \sum_{i=1}^{N_m} \hat{P}(y_i|\mathbf{x}_i) \quad (6.7)$$

To estimate the probability, k-nearest-neighbor (KNN) is employed in this study because of its simplicity and effectiveness. In KNN, the posterior probability $\hat{P}(y_i|\mathbf{x}_i)$ in Eq. (6.7) is defined as the ratio of the number of patterns that belong to y_i (the true class of the pattern $\mathbf{x}_i$) within the k nearest neighbors of the pattern $\mathbf{x}_i$, k_{y_i} , to the number of neighbors k .

$$\hat{P}^{KNN}(y_i|\mathbf{x}_i) = \frac{k_{y_i}}{k} \quad (6.8)$$

Substituting Eq. (6.8) to Eq. (6.7) yields Eq. (6.9), i.e. the error probability estimated by KNN, denoted as $\hat{P}_e^{KNN}(\mathbf{X}|z_m)$.

$$\hat{P}_e^{KNN}(\mathbf{X}|z_m) = 1 - \frac{1}{N_m} \sum_{i=1}^{N_m} \frac{k_{y_i}}{k} \quad (6.9)$$

Substitute Eq. (6.9) into Eq. (6.6) yields:

$$\widehat{P_e^{KNN}}([\mathbf{X}, \mathbf{Z}]) = \sum_m^M \frac{N_m}{N} \left[1 - \frac{1}{N_m} \sum_{i=1}^{N_m} \frac{k_{y_i}}{k} \right] \quad (6.10)$$

6.3 Search Algorithm for Mixed Data

To our best knowledge, no existing search algorithm takes the different feature types into account during the searching process. If different feature types are taken into account in search algorithm, subsets generated in a certain selection step might be various, including subsets with only continuous features, subsets with only nominal features and/or subsets with both continuous and nominal features. Comparing various subsets is a difficult task because different criteria employed for subsets with different feature types may not be comparable. However, comparison among subsets with different types is unavoidable since the best feature subset of mixed data should not be limited in subsets with both continuous and nominal features; it can also be subset with single-typed features, based on the particular machine learning problems.

In this section, we propose a search algorithm, called Mixed Forward Selection (MFS), for mixed data. In MFS, different feature types are taken into account in subset generation, evaluation and comparison using a filter-wrapper hybrid framework. Just like SFS (Sequential Forward Selection), MFS starts from an empty set and adds one feature at each step. Unlike SFS, each step MFS select the step best feature by integrating both filter and wrapper methods in sequential.

Figure 6.2: Block Diagram of Mixed Forward Selection (MFS).

The block diagram and pseudocode of MFS is shown in Fig. 6.2 and Table 6.3, respectively. Initially, the candidate features U are firstly divided into candidate continuous and nominal subsets, denoted as U_x and U_z respectively, as shown in Fig. 6.2. In the first stage, MFS searches the optimal nominal and continuous feature separately. When searching U_x , the search algorithm adds each of the candidate continuous features in U_x into the feature subset S selected at preceding step. If S is a purely continuous feature subset, the subsets generated would be continuous too, then the continuous feature criterion (J_c) should be used. Otherwise, all the feature subsets generated are mixed-typed and the mixed feature criterion (J_m) should be used. Finally, the best continuous feature, denoted as X' , is identified. Similarly, the best candidate nominal feature, denoted as Z' , is identified. In the second stage, the step-optimal feature from the two candidate features identified at the first stage by comparing the classification accuracies led by subsets $S \cup X'$ and $S \cup Z'$. The subset with higher classification accuracy evaluated by inner cross-validation of some predefined classifier is the final step-optimal feature.

This two-stage MFS carefully considers feature type in feature subsets generation

Algorithm: Mixed Forward Selection (MFS)

Input: U (the full feature set),
 p (the predefined number of feature to be selected),
 $J_c(\cdot)$, $J_n(\cdot)$ and $J_m(\cdot)$ (evaluation criteria for subsets with continuous, nominal and mixed features, respectively),
 $Acc(\cdot)$ (classification accuracy evaluation criterion)
Output: S (the subset of selected features)

Initialize:

set S to the empty set; // S is the subset of selected features
 set U_X and U_Z to the candidate feature subsets
 with continuous and nominal features, respectively;
 set X' and Z' to variables; // X' and Z' are the type optimal features

while $|S| < p$ **do**

 /*filter stage to select one best feature for each type*/

if S is empty

$X' = \arg \max_j J_c \{S \cup U_X(j)\};$

$Z' = \arg \max_j J_n \{S \cup U_Z(j)\};$

if S is subset with only continuous features

$X' = \arg \max_j J_c \{S \cup U_X(j)\};$

$Z' = \arg \max_j J_m \{S \cup U_Z(j)\};$

if S is subset with only nominal features

$X' = \arg \max_j J_m \{S \cup U_X(j)\};$

$Z' = \arg \max_j J_n \{S \cup U_Z(j)\};$

if S is subset with mixed features

$X' = \arg \max_j J_m \{S \cup U_X(j)\};$

$Z' = \arg \max_j J_m \{S \cup U_Z(j)\};$

 /*wrapper stage to select best feature*/

if $Acc(S \cup X') > Acc(S \cup Z')$

 //if X_{opt} results in a higher classification accuracy

$S = S \cup X'$; // X' is added into S

$U_X = U_X - X'$; // and deleted from U_X

else

$S = S \cup Z'$;

$U_Z = U_Z - Z'$;

Return S ;

Table 6.3: Pseudocode of Search Algorithm: Mixed Forward Selection

and comparison. Three different criteria are employed for continuous, nominal and mixed feature subsets respectively, but comparison of different criteria is avoided. Various evaluation criteria could be used for continuous and nominal features in our algorithm. In our experimental study, Mahalanobis distance and symmetrical uncertainty (see, Eq. (3.7)) are employed as the evaluation criteria for continuous and nominal features respectively. The stopping criterion in the MFS algorithm is either a predefined feature subset size or the cross-validation error rate.

6.4 Experimental Studies

In this section, the performance of MFS is tested and compared with other well-known methods on both artificial and real-world mixed data. In the first part, we test the robustness of our methods by creating a series of artificial datasets by varying number of values in nominal features and the sample size. In the second part, we continued to evaluate MFS by using six real-world datasets from UCI machine learning data repository (Murphy and Aha, 1995). The performance of MFS is compared with other methods that are often applied to mixed data.

6.4.1 Artificial Data

In this section, artificial mixed data is used to test whether our method is robust under various number of samples and drastically different number of distinct values in nominal features.

The artificial data generation program used in NIPS 2001 is employed to generate

the continuous feature set with uniform distribution and any proportion of relevant features. To form the mixed data set, we generate the nominal features by discretizing some of the continuous features. Given a continuous feature X_j for discretization and a predefined number of values V , the possible value set $\mathcal{V} = \{1, 2, \dots, V\}$ is firstly randomly grouped into two mutually exclusive subsets: $\mathcal{V}_1$ and $\mathcal{V}_2$ with approximately equal sizes. Then, the particular value of each element of the nominal feature Z_j , say z_{ij} , is selected from either $\mathcal{V}_1$ or $\mathcal{V}_2$ based on the corresponding value x_{ij} of the continuous feature while the selection of particular value inside the subset is random from the uniform distribution, i.e.

$$z_{ij} = \begin{cases} \text{any value from } \mathcal{V}_1, & \text{if } x_{ij} \leq \bar{X}_j \\ \text{any value from } \mathcal{V}_2, & \text{if } x_{ij} > \bar{X}_j \end{cases} \quad (6.11)$$

Bias of Criteria to Multivalued Nominal Features

To test the bias of our method and other methods to multivalued nominal features, we generated a simple feature subset with one continuous feature and one nominal feature in three different cases:

case 1 relevant continuous AND relevant nominal features

case 2 relevant continuous AND irrelevant nominal features

case 3 relevant nominal AND irrelevant continuous features

In each situation, different nominal features were made equally informative with different number of values V : 2, 5, 10 and 20 respectively. The new criterion

(P_e^{KNN}) and three other criteria including generalized Mahalanobis distance (GMD), Mahalanobis distance (MD), symmetric uncertainty (UC) were employed. Since MD and UC were defined for continuous and nominal features respectively, the nominal feature in the mixed feature subset was regarded as continuous one when MD was employed, while the continuous feature was discretized into nominal one using recursive entropy minimization heuristic-based discretization method (Fayyad and Irani, 1993) when UC was employed.

Figure 6.3: The new criterion P_e^{KNN} , GMD, MD and UC in case 1, i.e. the mixed data set with relevant continuous and nominal features. (The vertical bars represents symmetric error bars of unit standard deviation for mean values.)

Fig. 6.3 shows the criteria in case 1, i.e. the mixed data set with a relevant continuous and a nominal features. For P_e^{KNN} , higher error probability is observed when the number of feature values V is large and the number of samples n is small. This implies that P_e^{KNN} might underestimate the nominal feature with more number of values especially when the number of samples is small.

GMD could be larger when the number of nominal feature values is large and the number of samples is small. This bias is actually opposite to the previous one since it would make GMD in favor of the nominal feature with more number of values, especially when the number of patterns is small.

The curves of MD are little influenced by the number of samples n . These curves decrease from $V = 2$ to $V = 5$ but remain unchanged when V is greater than 5, so they also have little to do with the large V . One possible explanation is that the nominal feature with 2 values still retains the order information that could be measured by MD while the information of the nominal feature with more values could hardly be measured. The similar tendency of curves appears in Fig. 6.4 when the feature subset has relevant continuous and irrelevant nominal features. But when the nominal feature is irrelevant, as shown in Fig. 6.5, the curves are nearly unchanged with the number of nominal feature values because no information of nominal feature could be ignored.

For UC, the curves as shown in Fig. 6.3 tend to decrease at first and then increase with the increasing number of nominal feature values V . This tendency is clearer when the number of samples n is small and similar results could be observed in other

Figure 6.4: The new criterion P_e^{KNN} , GMD, MD and UC in case 2, i.e. the mixed data set with relevant nominal feature and irrelevant continuous feature. (The vertical bars represents symmetric error bars of unit standard deviation for mean values.)

two situations as shown in Fig. 6.4 and Fig. 6.5. When evaluating nominal features, UC is often employed instead of information gain which tends to overestimate the goodness of multi-valued nominal features. However, this experimental results show that UC still has some limitations especially when the number of samples is small.

Figure 6.5: The new criterion P_e^{KNN} , GMD, MD and UC in case 3, i.e. the mixed data set with relevant continuous feature and irrelevant nominal feature. (The vertical bars represents symmetric error bars of unit standard deviation for mean values.)

Correlation Between Selective Ranks and Number of Feature Values

In the previous section, we have tested the evaluation criteria on the simple mixed feature subsets with only one nominal feature. Next, we will test feature selection algorithms on the mixed feature subset with many nominal features. Since the heuristic feature selection algorithms aim to find the best subset based on the feature ranks, the correlation between the number of values of the nominal features and their

corresponding ranks reveals how the selective results are affected by the number of values of the nominal features.

We generated a mixed data set of 600 patterns represented by 20 features, out of which 2 were continuous and 18 were nominal. All the nominal features were equally informative with different number of values, which were random integers from $[2, 3, \dots, V_{max}]$. To investigate whether the ranks were correlated with the values of the nominal features, V_{max} was increased from 3 to 20, where a larger V_{max} resulted in more drastically different number of feature values. The correlation between the ranks and the number of values was measured by Pearson's linear correlation coefficient as below.

$$R_{YX} = \frac{\sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})}{(N-1)s_x s_y} \quad (6.12)$$

where $\bar{x}$ and s_x is sample mean and sample standard deviation of variable X with N elements. The correlation is 1 in the case of an increasing linear relationship, -1 in the case of a decreasing linear relationship, and some value in between in all other cases, indicating the degree of linear dependence between the variables. The closer the coefficient is to either -1 or 1 , the stronger the correlation between the variables.

Fig. 6.6 shows the correlation between the number of feature values and the ranks output by $\text{MFS}[P_e^{KNN}]$ and $\text{SFS}[\text{GMD}]$ with the increasing V_{max} . (Feature selection employing evaluation criteria, such as MD and UC, are not included in comparison because feature-type transformation could also affect the ranks.) For

Figure 6.6: The Pearson's linear correlation between ranks assigned to the nominal features and the number of feature values vs. different ranges of numbers of feature values.

$MFS[P_e^{KNN}]$, a positive correlation tends upward slightly with the larger V_{max} . This result implies that $MFS[P_e^{KNN}]$ would favor the nominal feature with fewer values.

This result is consistent with the characteristics of the new criterion as we have shown in the previous section. For $SFS[GMD]$, a negative correlation is obtained with an upward tendency, which reveals that $SFS[GMD]$ might favor the nominal feature with more values. The comparison of the tendencies of both methods show that the curve of our method is flatter than that of $SFS[GMD]$.

6.4.2 Real-World Data

This section describes the results of testing the proposed MFS method on six real-world data with mixed features. Since the relevant features are unknown in advance for these datasets, the performance of learning algorithms with enlarging feature subsets selected by feature selection algorithms is regarded as the indication of usefulness of MFS.

The machine learning algorithms using here are naive Bayes (Domingos and Pazani, 1997) and k -nearest neighbors (Duda and Hart, 1973) with $k = 3$ (both methods are described in Section 2.2.2). Naive Bayes classifier can be applied for mixed data by estimating the probabilities for nominal features using counts and those for continuous features using Parzen Window density estimation method (Duda et al., 2001).

To estimate the “true error rate” of classification, K -fold cross-validation is employed in our experiment. In K -fold cross-validation, given data are divided into K subsets of approximately equal size. We train the classifier K times, each time leaving out one of the subsets from training, but using only the validation set to test the generalization performance of learnt classifier. We set K to 5 throughout this experimental study since a common rule of thumb is that we can use Normal distribution when there are at least 30 samples for testing (see Mitchell, 1997, chapter 5). The averaged result of 50 replicas of 5-fold cross-validation is reported. The missing values in the continuous and nominal feature are replaced by the mean and mode respectively.

Data Description

Six real-world mixed datasets used in this experiment are summarized in Table 6.4. All these datasets are obtained from UC Irvine Repository (Murphy and Aha, 1995), including 5 2-class problems and 1 multi-class problem. Half of datasets are balanced in class distribution, while the other half of them are imbalanced. Details of datasets are described as follows.

no.	Dataset	Features			no. classes	Samples		
		all	nominal	cont		all	class1	class2
1	cleveland	13	7	6	2	303	164	139
2	crx	15	9	6	2	690	383	307
3	bridges	11	7	4	6	108	multi-class	
4	hepatitis	19	13	6	2	155	32	123
5	horse-colic	23	16	7	2	301	99	202
6	long-beach-va	12	6	6	2	200	51	149

Table 6.4: Summary of datasets. All these datasets are obtained from UC Irvine Repository (Murphy and Aha, 1995). Datasets above the horizontal line are “balanced” over different classes and those below are “imbalanced”.

Cleveland heart disease dataset (cleveland) contains 303 recodes of heart disease, collected from Cleveland Clinic Foundation. The task is to distinguish the presence or absence of heart disease in patients. There are 7 nominal features and 6 continuous features. The features include: age, sex, chest pain type, cholesterol, fasting blood sugar, resting ECG, max heart rate, etc.

Australian credit screening dataset (crx) contains 690 samples from an Australian credit company. The task is to determine whether a credit card should be given to an applicant. The dataset contains 6 continuous features and 9 nominal features, all of which have been coded to preserve confidentiality. This dataset is

interesting because it is a good mixture of features and contains nominal features with both small numbers of values (2 values) and large number of values (maximum: 14 values).

Pittsburgh bridges dataset (bridges) contains 108 bridges (samples) belonging to 6 “types” (classes). 12 features are available in this data set, in which the first feature “identifier” is no use for classification. Therefore, in this experiment, we omit feature “identifier” and predict the “type” of bridges using the remaining 11 features.

Hepatitis dataset contains 155 instances in this dataset created by G.Gong (Carnegie-Mellon University). This task is to determine whether a hepatitis patient *die* or *live*. The dataset contains 13 nominal features that are all binary ones and 6 continuous features. The class distribution is 32 die vs. 123 live. Obviously, it is a imbalanced class distribution. Missing values are presented in 15 out of 19 features.

Horse-colic dataset (horse-colic) contains 368 instances in this dataset created by Mary McLeish and Matt Cecile from University of Guelph. This task is to determine whether a lesion is surgical. The dataset contains seven continuous and fifteen nominal features, where the number of nominal features is much bigger than that of continuous features and the number of distinct values in any nominal feature is no more than 6.

Long-beach-va heart disease (long-beach-va) contains 200 samples of heart disease, collected from V.A. Medical Center, Long Beach, CA. The task is same as cleveland dataset as described above, *i.e.* to distinguish the presence or absence of

heart disease in patients. There are 6 nominal features and 6 continuous features.

Evaluating MFS vs. SFS with 2 ML Algorithms

In this section, we test MFS and compare it with other two feature selection algorithms SFS[MD] and SFS[UC]. Figures 6.7 to 6.12. show the learning curves of three feature selection algorithms in terms of increasing number of features selected, based on two learning algorithms Naive Bayes classifier in graph (a) and 3-nearest neighbor classifier in graph (b).

SFS[MD] and SFS[UC] using SFS as search algorithm but different evaluation criteria. When MD (Mahalanobis distance) is used as evaluation criterion, all the nominal features are used as continuous ones, and when UC (symmetrical uncertainty) is used for subset evaluation, all the continuous features are discretized into nominal ones in advance. These two methods will either lose the mutual information in nominal features or order information in continuous features due to the type transformation. MFS avoids the type transformation by integrating suitable criteria for subsets with various types, so that is expected to take better use of information in different types of features.

The experimental results are summarized as follows:

- (i) These figures show that MFS outperforms both SFS[MD] and SFS[UC] on 5 out of 6 datasets. One exception is cleveland dataset (as shown in Fig. 6.7), where SFS[MD] obtains the best result. A possible reason is that the nominal features in cleveland dataset are actually ordinal ones, which can be evaluated

by MD effectively. However, SFS[MD] can not perform well if the values of nominal feature are unordered, especially when those unordered nominal features are very important for discrimination, as we will analyzed next.

- (ii) The good performance can be obtained by either SFS[MD] or SFS[UC] in one datasets, but cannot be obtained by both. SFS[MD] outperforms SFS[UC] in 3 datasets including cleveland (Fig. 6.7), crx (Fig. 6.8), long-beach-va (Fig. 6.12) while SFS[UC] outperforms SFS[MD] in the other 2 dataset including bridges (Fig. 6.9) and horse-colic (Fig. 6.11). This result implies that the useful information can be lost in type transformation and the usage of criterion suitable for a particular feature type is limited in mixed data.

Evaluating MFS vs. Other FS algorithms with 2 ML Algorithms

Next we compare MFS with other well-known feature selection algorithms including RELIEF (Rendell and Kira, 1992; Kononenko, 1994), CFS (Correlation-based Feature Selection) (Hall, 1998) and SFS[GMS] (Generalized Mahalanobis distanced based SFS) (Bar-Hen and Daudin, 1995). RELIEF, as a feature ranking method, can be viewed as baseline in comparison.

Figures 6.13 to 6.18 show the experimental results of MFS and other feature selection algorithms on 6 real-world data using Naive Bayes classifier as in graph (a) and 3-NN as in graph (b).

The experimental results can be summarized as follows:

- (i) MFS outperforms RELIEF in all six real-world datasets.

- (ii) MFS outperforms SFS[GMD] in 5 out of 6 datasets. Again, cleveland dataset is the exception, as shown in Fig. 6.13.
- (iii) MFS obtains a comparable performance with CFS. In datasets cleveland and bridges (multi-class problem), CFS obtains the best performance; while in other 3 datasets including crx, horse-colic and long-beach-va, MFS obtains the best performance. In hepatitis datasets, the difference between MFS and CFS is insignificant.
- (iv) When the number of features selected is small, say less than 5, MFS is always best compared with all other feature selection algorithms. When the size of best feature subset is small, MFS find a subset leading lowest error rate, such crx datasets as shown in Fig. 6.14 and horse-colic datasets as shown in Fig. 6.17.

6.5 Chapter Summary

In this chapter, we have presented a mixed feature selection algorithm, which includes a new mixed feature evaluation criterion and a new search algorithm. The performance of our method has been tested using both artificial mixed data sets and benchmark real world mixed data sets. Experimental results show that our method provides improved performance compared with some commonly used algorithms for mixed feature selection.

Our method is readily applicable to data sets with moderate number of features.

When the number of features is huge, a direct application of our method could be computationally prohibitive. In such a case, a pre-selection procedure of individual feature ranking could be employed to reduce the size of feature set and then set-based feature selection algorithm is applied to the feature set with reduced-size. How to rank mixed feature is our future work.

(a) Using Naive Bayes as classification algorithm

(b) Using k -Nearest Neighbor as classification algorithm

Figure 6.7: Learning curves for SFS[MD], SFS[UC] and MFS with Naive Bayes classifier (a) and k -nearest neighbor classifier (b) on dataset: cleveland.

(a) Using Naive Bayes as classification algorithm

(b) Using k -Nearest Neighbor as classification algorithmFigure 6.8: Learning curves for SFS[MD], SFS[UC] and MFS with Naive Bayes classifier (a) and k -nearest neighbor classifier (b) on dataset: crx.

(a) Using Naive Bayes as classification algorithm

(b) Using k -Nearest Neighbor as classification algorithm

Figure 6.9: Learning curves for SFS[MD], SFS[UC] and MFS with Naive Bayes classifier (a) and k -nearest neighbor classifier (b) on dataset: bridges.

(a) Using Naive Bayes as classification algorithm

(b) Using k -Nearest Neighbor as classification algorithm

Figure 6.10: Learning curves for SFS[MD], SFS[UC] and MFS with Naive Bayes classifier (a) and k -nearest neighbor classifier (b) on dataset: hepatitis.

(a) Using Naive Bayes as classification algorithm

(b) Using k -Nearest Neighbor as classification algorithm

Figure 6.11: Learning curves for SFS[MD], SFS[UC] and MFS with Naive Bayes classifier (a) and k -nearest neighbor classifier (b) on dataset: horse-colic.

(a) Using Naive Bayes as classification algorithm

(b) Using k -Nearest Neighbor as classification algorithmFigure 6.12: Learning curves for SFS[MD], SFS[UC] and MFS with Naive Bayes classifier (a) and k -nearest neighbor classifier (b) on dataset: long-beach-va.

(a) Using Naive Bayes as classification algorithm

(b) Using k -Nearest Neighbor as classification algorithm

Figure 6.13: Learning curves for RELIEF, CFS, SFS[GMD] and MFS with Naive Bayes classifier (a) and k -nearest neighbor classifier (b) on dataset: cleveland.

(a) Using Naive Bayes as classification algorithm

(b) Using k -Nearest Neighbor as classification algorithmFigure 6.14: Learning curves for RELIEF, CFS, SFS[GMD] and MFS with Naive Bayes classifier (a) and k -nearest neighbor classifier (b) on dataset: crx.

(a) Using Naive Bayes as classification algorithm

(b) Using k-Nearest Neighbor as classification algorithm

Figure 6.15: Learning curves for RELIEF, CFS, SFS[GMD] and MFS with Naive Bayes classifier (a) and k -nearest neighbor classifier (b) on dataset: bridges.

(a) Using Naive Bayes as classification algorithm

(b) Using k -Nearest Neighbor as classification algorithm

Figure 6.16: Learning curves for RELIEF, CFS, SFS[GMD] and MFS with Naive Bayes classifier (a) and k -nearest neighbor classifier (b) on dataset: hepatitis.

(a) Using Naive Bayes as classification algorithm

(b) Using k -Nearest Neighbor as classification algorithm

Figure 6.17: Learning curves for RELIEF, CFS, SFS[GMD] and MFS with Naive Bayes classifier (a) and k -nearest neighbor classifier (b) on dataset: horse-colic.

(a) Using Naive Bayes as classification algorithm

(b) Using k -Nearest Neighbor as classification algorithm

Figure 6.18: Learning curves for RELIEF, CFS, SFS[GMD] and MFS with Naive Bayes classifier (a) and k -nearest neighbor classifier (b) on dataset: long-beach-va.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

Feature selection is the key to the success of pattern recognition system. Therefore, it has been a very active area in the past decades. As the application of pattern recognition spreads in other areas, new challenges come as well. This thesis focuses on feature selection and aims at addressing some current challenging problems including feature selection for very-high dimensional data and mixed data with both continuous and nominal features. In this thesis, three feature selection algorithms have been proposed in Chapter 4, Chapter 5 and Chapter 6, respectively.

To deal with a very high dimensional data, the computational efficiency of the feature selection algorithm is the primary concern. Therefore, feature ranking has been commonly applied here due to its computational efficiency. The sophisticated feature selection algorithms, which work well for the traditional datasets, may not be scalable to a very high dimensional data. A widely used criterion in feature selection such as Mahalanobis measure provides an effective evaluation of the discriminative power of a feature subset. However, it is computationally very expensive

for the data with a very high dimension. To reduce the computational cost of the Mahalanobis measure, a recursive approach has been proposed to approximate the original Mahalanobis measure for feature subset evaluation, i.e. recursive Mahalanobis (RM) measure and regularized recursive Mahalanobis (RRM) measure have been developed.

- Instead of evaluating the Mahalanobis measure directly in high dimensional space, the recursive approach evaluates the measure through successive evaluations in 2-dimensional space. Because of its recursive nature, this approach is extremely efficient when it is combined with a forward search procedure.
 - To evaluate a feature subset with n features in sequential forward selection, the original Mahalanobis measure involves calculating the inverse of a $n \times n$ covariance matrix, where the computational complexity for the inverse of $n \times n$ matrix by the numeric method is $O(n^3)$. The recursive Mahalanobis measure requires to calculate the inverse of a 2×2 covariance matrix, where the computational complexity is $O(n)$.
 - Both the original Mahalanobis measure and the recursive Mahalanobis measure have been tested using 8 gene expression datasets. The experiments showed classification error rates based on the recursive Mahalanobis measure were comparable to the original Mahalanobis measure.
- We also found that both the Mahalanobis measure and recursive Mahalanobis measure tended to overfit the training data and performed unsatisfactorily on

the unseen data, due to the relatively small sample size compared with the high dimension of features. To alleviate the overfitting problem, a regularized recursive Mahalanobis measure has been proposed. Some guidelines on the determination of the regularization parameters have also been provided.

- The regularized recursive Mahalanobis measure has been tested by the same 8 gene expression datasets and compared with the benchmark method in gene (feature) selection: SVM-RFE. The experimental results showed that the regularized recursive Mahalanobis measure obtained the lower error rates in 7 out of 8 datasets than the SVM-RFE did. The reductions in error rates in the above 7 datasets are in the range of 0.4% to 7%.

For a very high dimensional data, correlation-based individual feature ranking has been widely used due to its computational efficiency. The correlation between an individual feature and class labels reflects the predictive power of the feature. However, the individual based feature ranking can not eliminate the relevant but redundant features. To eliminate the redundancy among features, a new set-based correlation measure has been proposed for feature subset evaluation through orthogonal decomposition.

- This new criterion is combined with the sequential forward selection. In each step, instead of orthogonalizing a candidate feature with each of the features in the feature subset selected in previous steps, the new approach orthogonalizes the candidate feature to the class label prediction built upon the feature subset

selected in previous steps. Therefore, the redundancy of the candidate feature could be identified directly.

- The overfitting problem has also been investigated and alleviated through introducing regularization.
 - The experimental results showed that introducing regularization was helpful to alleviate the overfitting problem. The reduced classification error rates have been obtained in all 6 gene datasets when the regularization parameters were set to non-zero.
 - We also compared the classification error rate of the regularized correlation criterion with the regularized recursive Mahalanobis measure. This comparison showed that these two proposed methods obtained similar classification error rates in these 6 gene expression datasets. The regularized recursive Mahalanobis measure is more faster than the regularized correlation measure since the regularized correlation measure involved training the linear SVM classifier once when a new feature was added into the selected feature subset.
 - Compared with the benchmark method SVM-RFE, the reductions in error rates in these 6 datasets ranged from 0.4% to 8%.

For mixed data analysis, the common practice was to transform the data into either continuous data or nominal data through a feature type transformation. This transformation might either lose the order information in continuous features or

the mutual information in nominal feature. To make better use of different typed features, a novel feature selection algorithm, including a new criterion based on error probability and a new search algorithm for the data with mixed features, has been proposed.

- First, an error probability-based criterion has been developed for the mixed feature subset evaluation. This new criterion has been combined with the mixed forward search procedure and evaluate a mixed feature subset through integrating the error probabilities estimated in a series of continuous feature subspaces. These feature subspaces were obtained by space decomposition based on the nominal features in the subset. The merits of this new criterion is that it can detect the association between features of different types.
- In addition to the mixed evaluation criterion, a mixed forward search procedure has been developed specially for mixed data, considering different feature types in subset generation, evaluation and comparison. In mixed forward selection procedure, three different criteria have been employed for continuous, nominal and mixed feature subsets respectively, but the direct comparison of different criteria was avoided. This new search procedure could improve the reliability of feature evaluation.
 - The proposed method, mixed forward selection (MFS), has been tested using 6 real-world mixed datasets. MFS was first compared with its counterpart: sequential forward selection (SFS), which involved a feature type transformation. The classification error rates obtained by MFS were

lower than that obtained by SFS in 5 out of 6 datasets. This results implied that avoiding the feature type transformation in a mixed feature selection problem can be helpful to obtain the better feature subsets and hence lead to the better classification performance.

- We also compared MFS with other well-know feature selection algorithms including RELIEF, correation-based feature selection (CFS) and generalized Mahalanobis distance based SFS (SFS[GMD]). MFS outperforms RELIEF in all 6 mixed datasets and outperforms SFS[GMD] in 5 out of 6 datasets, while MFS obtained a comparable performance with CFS. When the number of features selected was less than 5, MFS was always best compared with other feature selection algorithms.

The proposed method is readily applicable to datasets with moderate number of features. For very high dimensional data, a direct application of our method could be computationally expensive. In this case, a pre-selection procedure of individual feature ranking can be employed. The method to rank mixed features by their importance will be our future work.

7.2 Recommendations for Future Work

These are some possible areas of future research, as follows.

- **Determining the size of initial candidate feature subset for a very-high dimensional data**

In this thesis, regularization has been introduced to the set-based feature evaluation to overcome the overfitting problem in very-high dimensional data. The other practical method to alleviate the overfitting problem in a very-high dimensional data is to find a initial candidate feature subset with enough but manageable number of features by feature ranking first. To do this, the a very-high dimensional data is first reduced to a relatively low dimensional data which can then be handled by some sophisticated set-based feature selection algorithm. This method, sometimes called two-staged feature selection algorithm, has received many attentions in a very high dimensional data analysis. One problem of two staged feature selection algorithm is the choice of the size of initial candidate feature subset. Determining the size of the initial candidate subset controls the tradeoff among the computational complexity, classification accuracy and overfitting problem.

In literature, the size of candidate feature set has been set to 50, 100, 200 and 1000 for gene expression data. But it is still unclear which choice is better. The suitable size of the initial candidate feature set should be problem dependent and will impact the final classification performance. How to determine a suitable size of initial candidate feature set worths more investigations in future.

- **The impact of imbalanced problem to feature selection in gene expression data**

In this thesis, gene expression data is used to test the proposed feature selection

algorithms for a very-high dimensional data. In our methods, two typical characteristics of gene expression data have been noticed, including (i) very-high dimensional feature space and (ii) small number of samples available. In fact, the imbalanced problem is also pervasive in gene expression data. The imbalanced problem means the number of samples in different class is (highly) imbalanced. For example, the leukemia dataset (Golub et al., 1999) contains 72 patients with either lymphoblastic leukemia (ALL, 47 samples) or acute myeloid leukemia (AML, 25 samples) and the number of samples in ALL is nearly two times of that in AML. In pattern classification, it has been well understood the effect of imbalanced problem on the classifier's behavior: noisy or otherwise unreliable samples from the majority class can overwhelm the minority class, although in practice, the minority class is often the valuable research objective. In feature selection, the imbalanced problem might mislead the feature selection and deteriorate the performance of classification.

In future research, the impact of imbalanced problem to feature selection for very-high dimensional data needs more investigation. Enhanced feature selection algorithm with the capability of handling imbalanced problem for gene expression data is a worthy development for future.

- **Efficient feature selection algorithm for mixed data**

In this thesis, mixed forward selection procedure has been proposed for mixed data. This method achieves a competitive and robust performance through a filter-wrapper hybrid framework. However, this method introduces more

computation which may limit the usage of the method for high dimensional mixed data with thousands of features. Efficient feature selection algorithm is worth further exploration. The most efficient method, feature ranking, might be extended to mixed data if some univariate criterion could be developed for different types of features.

Although finding more powerful feature selection algorithms are always the main aims, one thing is clear: no algorithm can outperform others in all problems. In users' viewpoint, a more important problem is how to choose a suitable feature selection algorithm for a particular problem. Systematical guidance for choosing the algorithm in a particular problem also needs further research.

Author's Publications

Journal Papers

1. **Wenyin Tang** and K. Z. Mao, Feature Selection Algorithm for Mixed Data with Both Nominal and Continuous Features, *Pattern Recognition Letters*, Volume 28, Issue 5, 1 April 2007, pp. 563-571.
2. K. Z. Mao, **Wenyin Tang** and Jun Yang, Gene Selection Based on Regulated Recursive Mahalanobis Separability Measure, submitted to *BMC, Bioinformatics*, 2007.
3. **Wenyin Tang** and K. Z. Mao, Impact of Candidate Gene Size for Gene Subset Selection, in preparation for *Bioinformatics*, 2007.

Conference Papers

1. K. Z. Mao and **Wenyin Tang**, Correlation-based Relevancy and Redundancy Measures for Efficient Gene Selection, *the 2nd IAPR Workshop on Pattern Recognition in Bioinformatics (PRIB 2007)*, accepted.
2. **Wenyin Tang** and K. Z. Mao, Feature Selection Algorithm for Mixed Data with Both Nominal and Continuous Features, *PAKDD 05, Lecture Notes in*

Computer Science, vol. 3518, pp. 683-688, Berlin Heidelberg, 2005.

Bibliography

David Aha. Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms. *International Journal of Man-Machine Studies*, 36(2):267–287, 1992.

David Aha, Dennis Kibler, and Marc Albert. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.

Hussein Almuallim and Thomas G. Dietterich. Learning with many irrelevant features. In *Ninth National Conference on Artificial Intelligence*, pages 547–552. MIT Press, 1991.

Hussein Almuallim and Thomas G. Dietterich. Efficient algorithms for identifying relevant features. In *Proceedings of the Ninth Canadian Conference on Artificial Intelligence*, pages 38–45. Morgan Kaufmann, 1992.

Hussein Almuallim and Thomas G. Dietterich. Learning boolean concepts in the presence of many irrelevant features. *Artificial Intelligence*, 69(1-2):279–306, 1994.

U Alon, N Barkai, D Notterman, K Gish, S Ybarra, D Mack, and AJ 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.
- Paul W. Baim. A method for attribute selection in inductive learning systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):888 – 896, 1988.
- A. Bar-Hen and J. J. Daudin. Generalization of mahalanobis distance in mixed case. *Journal of Multivariate Analysis*, 53:332–342, 1995.
- R. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5(4):537–550, 1994.
- R.E. Bellman. *Adaptive Control Processes*. Princeton University Press, Princeton, NJ, 1961.
- Jose Bins and Bruce Draper. Feature selection from huge feature sets. In *Proceedings. Eighth IEEE International Conference on Computer Vision*, volume 2, pages 159–165, 2001.
- Avrim Blum and Pat Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1-2):245–271, 1997.
- B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *5th Annual ACM Workshop on COLT*, pages 144–152, Pittsburgh, PA, 1992. ACM Press.

- M. Braga-Neto and Edward R. Dougherty. Is cross-validation valid for small-sample microarray classification? *Bioinformatics*, 20(3):374–380, 2004a.
- Ulisses Braga-Neto and Edward R. Dougherty. Bolstered error estimation. *Pattern Recognition*, 37(6):1267–1281, 2004b.
- Leo Breiman, Jerome Friedman, Charles J. Stone, and R. A. Olshen. *Classification and Regression Trees*. Chapman and Hall/CRC; New Ed edition (January 1, 1984), 1984.
- G. Bressard and P. Bratley. *Fundamentals of Algorithms*. Prentice Hall, 1996.
- Martin D. Buhmann and M. J. Ablowitz. *Radial Basis Functions : Theory and Implementations*. Cambridge University Press, 2003.
- Claire Cardie. Using decision trees to improve case-based learning. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 25–32. Morgan Kaufmann Publishers, Inc., 1993.
- Peter Clark and Tim Niblett. The cn2 induction algorithm. *Machine Learning*, 3: 261–283, 1989.
- T. M. Cover and P. E. Hart. Nearest neighbor pattern recognition. *IEEE transactions on Information Theory*, 13(1):21 – 27, 1967.
- Sanmay Das. Filters, wrappers and a boosting-based hybrid for feature selection. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 74 – 81, 2001.

- M. Dash and H. Liu. Feature selection for classification. *Intelligent Data Analysis*, 1:131–156, 1997.
- A. R. de Leon and K. C. Carrere. A generalized mahalanobis distance for mixed data. *Journal of Multivariate Analysis archive*, 92(1):174 – 185, 2003.
- P. A. Devijver and J. Kittler. *Pattern Recognition: A Statistical Approach*. Prentice/Hall International, 1982.
- Chris Ding and Hanchuan Peng. Minimum redundancy feature selection for gene expression data. In *IEEE Computer Society Bioinformatics Conference (CSB 2003)*, pages 523–529, Stanford, CA, 2003.
- Pedro Domingos. Rule induction and instance-based learning: A unified approach. In *IJCAI*, pages 1226–1232, 1995.
- Pedro Domingos. Contextsensitive feature selection for lazy learners. *Artificial Intelligence Review 11:227–253.*, 11:227–253, 1997.
- Pedro Domingos and Michael J. Pazzani. Beyond independence: Conditions for the optimality of the simple bayesian classifier. In *International Conference on Machine Learning*, pages 105–112, 1996.
- Pedro Domingos and Michael J. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29(2-3):103–130, 1997.
- James Dougherty, Ron Kohavi, and Mehran Sahami. Supervised and unsupervised

- discretization of continuous features. In *Machine Learning: Proceedings of the Twelfth International Conference*, 1995.
- Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, 1973.
- Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern classification (2nd edition)*. Wiley, New York, 2001.
- B. Efron and R.J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, 1993.
- B. Efron and R.J. Tibshirani. Improvements on cross-validation: The .632+ bootstrap method. *Journal of the American Statistical Association*, 92:548–560, 1997.
- U. M. Fayyad and K. B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning,. In *Proceedings of 13th International Joint Conference on Artificial Intelligence*, pages 1022–1027, 1993.
- F. Ferri, P. Pudil, M. Hatef, and J. Kittler. Comparative study of techniques for large-scale feature selection. In *Pattern Recognition in Practice IV, Multiple Paradigms, Comparative Studies and Hybrid Systems*,, pages 403–413, 1994.
- R. Florez-Lopez. Reviewing relief and its extensions: a new approach for estimating attributes considering high-correlated features. In *Proceedings. 2002 IEEE International Conference on Data Mining*, pages 605– 608, 2000.

- J. Friedman. Regularized discriminant analysis. *Journal of the American Statistical Association*, 84(165-175), 1989.
- D. Gamberger and N. Lavrac. Conditions for occam's razor applicability and noise elimination. In *Proceedings of Ninth European Conference on Machine Learning*, 1997.
- E. E. Ghiselli. *Theory of Psychological Measurement*. GcGraw-Hill, 1964.
- T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. R. 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*, 286:531–537, 1999.
- Gavin J Gordon, Roderick V Jensen, Li-Li Hsiao, Steven R Gullans, Joshua E Blumenstock, Sridhar Ramaswamy, William G Richards, David J Sugarbaker, and Raphael Bueno. Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma. *Cancer Research*, 62, 2002.
- Isabelle Guyon and Andre Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157 – 1182, 2003.
- Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.

- Mark A. Hall. Correlation-based feature selection for discrete and continuous class machine learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*. Morgan Kaufmann, San Francisco, CA, 2000.
- Mark A. Hall. Correlation-based feature selection for machine learning. *Ph.D diss.* Hamilton, NZ: Waikato University, Department of Computer Science., 1998.
- P.E. Hart. The condensed nearest neighbor role. *IEEE Transactions on Information Theory*, 14:515–516, 1968.
- Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning, Data Mining, Inference, and Prediction*. Springer, Verlag, 2001.
- S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1999.
- D. W. Hosmer and L. Stanley. *Applied Logistic Regression*. Chichester, Wiley, New York, 1975.
- Anil Jain and Douglas Zonker. Feature selection: Evaluation, application, and small sample performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):153–158, 1997.
- Anil K. Jain, Rober P. W. Duin, and Jianchang Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.
- George H. John, Ron Kohavi, and Karl Pfleger. Irrelevant features and the subset

- selection problem. In *International Conference on Machine Learning*, pages 121–129, 1994.
- R. Kohavi and G. H John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.
- Ron Kohavi. The power of decision tables. In Nada Lavrac and Stefan Wrobel, editors, *Proceedings of the European Conference on Machine Learning*, Lecture Notes in Artificial Intelligence 914, pages 174–189, Berlin, Heidelberg, New York, 1995a. Springer Verlag.
- Ron Kohavi. Wrappers for performance enhancement and oblivious decision graphs. *A Dissertation Submitted to the Department of Computer Science of Stanford University*, 1995b.
- Ron Kohavi and Chia Hsin Li. Oblivious decision trees, graphs and top-down pruning. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1071–1077, 1995.
- Ron Kohavi and Dan Sommerfield. Feature subset selection using the wrapper model: Overfitting and dynamic search space topology. In *The First International Conference on Knowledge Discovery and Data Mining*, pages 192–197, 1995.
- Igor Kononenko. Estimating attributes: Analysis and extensions of RELIEF. In *European Conference on Machine Learning*, pages 171–182, 1994.
- Pat Langley and Stephanie Sage. Induction of selective bayesian classifiers. In

- Proceedings of Tenth Conference on Uncertainty in Artificial Intelligence*, pages 399–406. Morgan Kaufmann, 1994.
- Huan Liu and Hiroshi Motoda. *Feature Selection For Knowledge Discovery And Data Mining*. Kluwer Academic Publisher, Doston/Dordrecht/London, 1998.
- Huan Liu and Rudy Setiono. A probabilistic approach to feature selection - a filter solution. In *International Conference on Machine Learning*, pages 319–327, 1996.
- Huan Liu and Lei Yu. Toward integrating feature selection algorithms for classification and clustering. *Knowledge and Data Engineering, IEEE Transactions on*, 17(4):491– 502, 2005.
- Huan Liu, Farhad Hussain, Chew Lim Tan, and Manoranjan Dash. Discretization: An enabling technique. *Machine Learning*, 6:393–423, 2002.
- Huan Liu, Hiroshi Motoda, and Lei Yu. A selective sampling approach to active feature selection. *Artificial Intelligence*, 159(1-2):49 – 74, 2004.
- R. S. Michalski. Pattern recognition as rule-guided inductive inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2:349–372, 1980.
- R. S. Michalski and J. B. Larson. Selection of most representative training examples and incremental generation of vl_1 hypothesis. *Dep. Comput. Sci., Univ. Illionis, Rep.687*, 1978.
- Tom M. Mitchell. *Machine Learning*. The McGraw-Hill Companies, Inc., New York, 1997.

P. M. Murphy and D. W. Aha. Uci repository of machine learning databases. *Irvine, CA: University of California*, 1995.

G. Pagallo and D Haussler. Boolean feature discovery in empirical learning. *Machine Learning*, 5:71–99, 1990.

P. Pavlidis, J. Weston, J. Cai, and W. N. Grundy. Gene functional classification from heterogeneous data. In *Proceedings of the Fifth Annual International Conference on Computational Biology*, pages 249 – 255, 2001.

Hanchuan Peng, Fuhui Long, and Chris 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, 2005.

Yanxiong Peng, Wenyan Li, and Ying Liu. A hybrid approach for biomarker discovery from microarray gene expression data. *Cancer Informatics*, 6:301–311, 2006.

Bernhard Pfahringer. Compression-based feature subset selection. pages 109–119, 1995.

Scott L. Pomeroy, Pablo Tamayo, Michelle Gaasenbeek, Lisa M. Sturla, Michael Angelo, Margaret E. McLaughlin, John Y. H. Kim, Liliana C. Goumnerova, Peter M. Black, Ching Lau, Jeffrey C. Allen, David Zagzag, James M. Olson, Tom Curran, Cynthia Wetmore, Jaclyn A. Biegel, Tomaso Poggio, Shayan Mukherjee, Ryan Rifkin, Andrea Califano, Gustavo Stolovitzky, David N. Louis, Jill P.

- Mesirov, Eric S. Lander, and Todd R. Golub. Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*, 415:436–442, 2002.
- W. H. Press, B. P. Flannery, and S. A. Teukolsky abd W. T. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 1988.
- P. Pudil, J. Novovicova, and J. Kittler. Floating search methods in feature-selection. *Pattern Recognition Letters*, 15(11):1119–1125, 1994.
- J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:88–106, 1986.
- J. R. Quinlan. *C4.5: Programmes for Machine Learning*. Morgan Kaufmann, 1993.
- John Rachlin, Simon Kasif, Steven Salzberg, and David W. Aha. Towards a better understanding of memory-based reasoning systems. In *International Conference on Machine Learning*, pages 242–250, 1994.
- L. A. Rendell and K. Kira. A practical approach to feature selection. In *International Conference on machine learning*, pages 249–256, 1992.
- John A. Rice. *Mathematical Statistics and Data Analysis (Second Edition)*. Duxbury Press, Belmont, California, 1995.
- Irina Rish. An empirical study of the naive bayes classifier. In *IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence*, 2001.
- Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Inc, 1995.

J. Shao and D. Tu. *The Jackknife and Bootstrap*. Springer-Verlag, New York, 1995.

W. Siedlecki and J. Sklansky. A note on genetic algorithms for large-scale feature-selection. *Pattern Recognition Letters*, 10:335–347, 1989.

Dinesh Singh, Phillip G. Febbo, Kenneth Ross, Donald G. Jackson, Judith Manola, Christine Ladd, Pablo Tamayo, Andrew A. Renshaw, Anthony V. D'Amico, Jerome P. Richie, Eric S. Lander, Massimo Loda, Philip W. Kantoff, Todd R. Golub, and William R. Sellers. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 1:203–209, 2002.

Moninder Singh. Induction of selective bayesian networks from data. In *AAAI/IAAI*, Vol. 2, page 1371, 1996.

Moninder Singh and Gregory M. Provan. A comparison of induction algorithms for selective and non-selective bayesian classifiers. In *International Conference on Machine Learning*, pages 497–505, 1995.

Moninder Singh and Gregory M. Provan. Efficient learning of selective bayesian network classifiers. In *International Conference on Machine Learning*, pages 453–461, 1996.

Craig Stanfill and David Waltz. Toward memory-based reasoning. *Communications of the ACM*, 29(12):1213 – 1228, 1986.

A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-posed problems*. W. H. Winston, Washington D.C., 1977.

- Lloyd N. Trefethen and David Bau. *Numerical Linear Algebra*, Society for Industrial and Applied Mathematics. 1997.
- van't Veer, H Dai, van de Vijver, Y D He, A A Hart, M Mao, H L Peterse, van der Kooy, M J Marton, A T Witteveen, G J Schreiber, R M Kerkhoven, C Roberts, P S Linsley, R Bernards, and S H Friend. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415, 2002.
- V. N Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1996.
- V. N Vapnik. *Statistical Learning Theory*. John Wiley and Sons, New York, 1998.
- S.M. Weiss and C.A. Kulikowski. *Computer Systems That Learn*. Morgan Kaufmann, 1991.
- M West, C Blanchette, H Dressman, E Huang, S Ishida, R Spang, H Zuzan, JA Jr Olson, JR Marks, and JR Nevins. Predicting the clinical status of human breast cancer by using gene expression profiles. *Proc Natl Acad Sci USA*, 98(20), 2001.
- D. Randall Wilson and Tony R. Martinez. Improved heterogeneous distance function. *Machine Learning*, 6, 1997.
- Lei Yu and Huan Liu. Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research*, 5:1205–1224, 2004.
- Xin Zhou and K. Z. Mao. Ls bound based gene selection for dna microarray data. *Bioinformatics*, 21(8):1559–1564, 2005.