Classification and feature selection for biomedical and biological data using soft computing

Zhou, Nina

2009

Zhou, N. (2009). Classification and feature selection for biomedical and biological data using soft computing. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/46861>

<https://doi.org/10.32657/10356/46861>

Nanyang Technological University

Downloaded on 26 Apr 2023 17:37:51 SGT

423
2X
D/Aue

CLASSIFICATION AND FEATURE SELECTION FOR BIOMEDICAL AND BIOLOGICAL DATA USING SOFT COMPUTING

ZHOU NINA

School of Electrical and Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy

2009

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original re-
search and has not been submitted for a higher degree to any other universities or
institutions.

.....

Date

.....

Zhou Nina

Acknowledgements

I would like to express my sincere gratitude and appreciation to my supervisor, Professor Wang Lipo, who has given me great guidance and suggestions during this work, also continuous encouragement, great patience and effort for revisions of this dissertation; Thanks also go to Chu Feng for many discussions and suggestions for part of this work; Same thanks to Hu Guang'an and Xu Shuhua who give me great help in understanding the HapMap genotype data. Great thanks also should be given to Professor Ser Wee, Dr Yu Zhuliang and all other staffs in the Center for Signal Processing, who give me great support in writing this dissertation. Thanks all my friends in Singapore and China for their mental support and encouragement.

I also wish to extend my appreciation to the school of Electrical and Electronic Engineering of Nanyang Technological University for the financial support; The reviewers of the doctoral committee, especially Professor Pina Marziliano, for careful scrutiny of the manuscript of this dissertation.

Last and also the most important, I want to say thanks to my families, especially to my beloved father (Zhou Qing) and mother. If it is not because of their unselfish dedication and unconditional support within all the past years, it would be impossible for me to come to Singapore for study, to say nothing of finishing this dissertation.

Feb. 9, 2008

Zhou Nina

Abstract

This dissertation mainly discusses feature selection and classification problems in the context of bio-medical data. Feature selection and classification are two important tasks in processing bio-medical data. For example, in disease diagnosis, finding the sub-groups of genes or symptoms that are specific to a particular disease is a feature selection problem. Once these sub-groups of genes or symptoms are found, diagnosing the disease which is a classification problem, will become easy. However, because most of the bio-medical data has large numbers of features, feature selection and classification are hence particularly challenging.

Considering the fact that different feature subsets have different power in distinguishing different classes, in this dissertation, we are particularly interested in proposing a general hybrid approach to selecting *class-dependent* feature subsets. We argue that for a C -class classification problem, C 2-class classifiers, each of which discriminating one class from the other classes and having a characteristic input feature subset, should in general out-perform, or at least match the performance of, a C -class classifier with one single input feature subset. For each class, we select a desirable feature subset which leads to the lowest classification error rate for this class using a classifier for a given feature subset search algorithm. We use “one-against-all” to construct one model for each class by training the classifier with the class of desirable feature subset. To fairly compare different feature subsets in different feature subspaces, we propose a heuristic weight method for

the class-dependent classifier, i.e., assigning a weight to each model's output before the comparison is carried out. Here, each weight corresponds to the probability that a sample belongs to each class when classifying the sample with the union set of all class-dependent feature subsets. After training and during testing, each test pattern is tested on all models and the model with the maximum weighted output decides the class of the test pattern. The method's performance is evaluated on two artificial data sets and several real-world benchmark data sets, with the support vector machine (SVM) as the classifier, and with the RELIEF, class separability and minimal-redundancy-maximal-relevancy as attribute importance measures. We also compare the classification results with those obtained by class-independent feature selection methods, class-dependent feature extraction methods and class-dependent feature selection methods with the "one-against-one" strategy. Our results indicate that the *class-dependent* feature subsets selected by our approach can effectively remove irrelevant or redundant features, while maintaining or improving (sometimes substantially) the classification accuracy, in comparison with other feature selection methods.

Besides the general hybrid approach to class-dependent feature selection, we also propose a ranking measure and apply it on SNP data. After ranking all SNPs (i.e., features of the data), we determine a desirable feature subset (with the minimum size and most information) in a greedy manner with the help of the SVM classifier. The result shows that a very small number of important SNPs selected by the proposed method can make a good classification of three populations. The F-statistics measure is also applied to select SNPs for a comparison. The ranking result and consequent classification show that the modified t-test ranking method is more effective than the F-statistics measure. Most importantly, the proposed ranking measure is specially able to deal with features in the format of vectors.

Contents

Acknowledgements	iii
Abstract	iv
Table of Contents	x
List of Abbreviations and Symbols	x
List of Tables	xv
List of Figures	xviii
1 Introduction	1
1.1 Background and Foundation	2
1.1.1 KDD and Data Mining	2
1.1.2 Data Dimensionality Reduction	4
1.1.3 Soft Computing	5
1.1.4 Classification	5
1.1.5 Bioinformatics Background for Feature Selection and Clas- sification	7
1.2 Motivation of Feature Selection	10
1.3 Objectives of Feature Selection	11
1.4 Major Contributions of the Thesis	12
1.5 Organization of the Thesis	13

2	Literature Review	14
2.1	Feature Selection	14
2.1.1	Feature Relevance	16
2.1.2	Feature Redundancy	17
2.1.3	Optimality of Feature Subsets	18
2.2	Feature Selection Methods	18
2.2.1	Filter Approaches	20
2.2.1.1	Filter Approaches Based on Ranking Measures . .	22
2.2.1.2	Filter Approaches Based on Various Search Strategies	26
2.2.2	Wrapper Approaches	30
2.2.2.1	Wrapper Approaches Based on Various Search Strategies	31
2.2.3	Feature Selection Based on Class Information	34
2.3	Summary	35
3	Feature Selection: A General Hybrid Approach to Class-dependent Feature Selection	37
3.1	Hybrid Approaches between Filters and Wrappers	39
3.1.1	Filters for the Hybrid Method	40
3.1.2	A Feature Extraction Method–Kernel Fisher Discriminant Analysis	44
3.1.3	One-against-one Strategy for Class-dependent Feature Selection	45
3.1.4	Wrappers for the Hybrid Method	46
3.2	The Proposed General Hybrid Approach to Class-dependent Feature Selection	47

3.3	Applications	49
3.3.1	Data Sets	49
3.3.2	Results and Discussions	51
3.4	Summary	61
4	Classification of Class-dependent Feature Subsets and One of Its Applications	64
4.1	The Support Vector Machines	64
4.2	The Classifier with Class-dependent Feature Subsets	67
4.3	A novel application of our method on face recognition	78
4.3.1	Background and Method	79
4.3.2	Experiments and Results	81
4.4	Summary	85
5	Effective Feature Selection and Classification on the HapMap Genotype Data	88
5.1	Motivation	88
5.2	Background	89
5.3	Methods	91
5.3.1	Modified T-test	92
5.3.2	F-statistics	94
5.3.3	The Classifier	94
5.4	Experiment Preparation	95
5.4.1	Data Description	95
5.4.2	Data Preprocessing 1	96
5.4.3	Implementation 1	99
5.4.4	Results of Preprocessing 1 and Discussions	99

- 5.4.5 Data Preprocessing 2 106
 - 5.4.6 Implementation 2 106
 - 5.4.7 Results of Preprocessing 2 and Discussions 107
 - 5.5 Summary 110
- 6 Conclusions 112**
 - 6.1 Summary of Work 112
 - 6.1.1 Feature Importance Ranking Measures 113
 - 6.1.2 The Notion of “Hybrid” 114
 - 6.1.3 The Notion of “Class-dependent” 115
 - 6.1.4 Current Limitations and Future Works 116
- References 120**

List of Abbreviations and Symbols

Abbreviations

DDR	data dimensionality reduction
KDD	knowledge discovery in database
CSM	class separability measure
mRMR	minimal redundancy and maximal relevancy
ITRC	information theoretic ranking criterion
MLP	multi-layer perceptron
NN	neural network
KNNs	k-nearest neighbors
NB	naive bayes
GAs	genetic algorithms
GMM	gaussian mixture model
FFNN	feed-forward neural network
BP	back propagation
SVM	support vector machine
DAGSVM	directed acyclic graph SVM
RBF	radial basis function
B&B	branch and bound

PLNs	piecewise linear networks
SPR	statistical pattern recognition
SA	simulated annealing
KFDA	kernel fisher discriminant analysis
SOAPs	selection of attributes by projections
PCA	principal component analysis
FDA	fisher discriminant analysis
FLDA	fisher linear discriminant analysis
ICA	independent component analysis
GTDA	general tensor discriminant analysis
PDFs	probability density functions
LVF	Las Vegas filter
SFS	sequential forward selection
SBS	sequential backward selection
BDS	bi-directional search
BSE	backward stepwise elimination
std	standard deviation
CIDFS	class-independent feature selection
CDFS	class-dependent feature selection
DTs	decision trees
FNNs	fuzzy neural networks
SNP	single nucleotide polymorphism
NSCM	nearest shrunken centroid method
CEU	Utah residents with ancestry from northern and western Europe
YRI	Yoruba in Ibadan, Nigeria in west Africa
CHB	Han Chinese in Beijing

JPT	Japanese in Tokyo
MI	mutual information
LD	linkage disequilibrium
ORCA	optimal rate of correct assignment

Symbols

S	the sample or training set
X	one instance or sample
X_i	the i -th instance or sample
x	one random sample
x_i	the i -th feature of x
x_{ij}	the j -th feature of the i -th sample
y_i	the class label of the sample X_i
g	one random sample
g_i	the i -th feature of g
N	number of samples
n_i	the number of samples in the i -th class
d	the number of features or dimension of feature space
d'	the number of features
D	the dimension of one feature space
C	the number of classes
c_i	the class label of the i -th class
c	one of C classes
ϕ	mapping function
φ	mapping function

m_i	mean value of samples in the i -th class
Σ	the covariance matrix
Σ^{-1}	the inverse of Σ
S^d	the sample set in the d -dimensional space
$\mathbb{R}^d$	a real d -dimensional space
F	feature sets
F^d	feature sets with dimension d
f^d	one feature set with dimension d
t	target function
t'	approximation of target function
J	selection criterion
$E_f(t)$	expected value of function t on the feature set f
$\vec{w}$	a weight vector
$\mathbf{X}_h$	nearest hit of the instance $\mathbf{X}$
$\mathbf{X}_m$	nearest miss of the instance $\mathbf{X}$
x_{mi}	the i -th element of $\mathbf{X}_m$
x_{hi}	the i -th element of $\mathbf{X}_h$
S_w	within-class distance
S_b	between-class distance
P_c	the probability of the c -th class
$I(x_i; x_j)$	mutual information of two features
$\vec{\mathbf{X}}$	feature vector
t_i	the t-statistics value of the i -th feature
t_{ic}	the t-statistics value of the i -th feature in the c -th class
S_i	within-class standard deviation of the i -th feature
S_0	median value of all S_i

M	image matrix
T	transformation matrix
I_a	informativeness for assignment
M_B	between class scatter matrix
M_W	within class scatter matrix
p	frequency of one allele
q	frequency of one allele
p_c	frequency of one allele for the c -th population
$\bar{p}$	mean frequency of one allele
$\bar{q}$	mean frequency of one allele
Var	variance of one allele
F_{st}	the F-statistics ranking value

List of Tables

- 2.1 Categorization of feature selection methods. 21
- 2.2 Probabilistic measures. 24
- 2.3 Comparison of search methods. 33
- 3.1 Basic information of the data sets used in this thesis. 51
- 3.2 The total number of times that each of the ten features is selected in 10 simulations for artificial data set A, using the CSM ranking and one-against-all measures. 53
- 3.3 The total number of features selected in each of the 10 simulations, the average number and its standard deviation (std), using class separability measure (CSM) to rank features for the artificial data A. 53
- 3.4 Same as Table 3.2, for the artificial data set B, which has 10 classes and 20 features. 54
- 3.5 Same as Table 3.3, for the artificial data B. 54
- 3.6 Same as Table 3.3, with RELIEF and CSM to rank features, for the Breast cancer data set with 9 feautres. 56
- 3.7 Same as Table 3.3, with RELIEF and CSM to rank features, for the Ionosphere data set with 34 features. 56
- 3.8 Same as Table 3.3, with RELIEF and CSM to rank features, for the Thyroid data set with 5 features. 56

3.9 Same as Table 3.3, with RELIEF and CSM to rank features, for the
Iris data set with 4 features. 57

3.10 Same as Table 3.3, with RELIEF and CSM to rank features, for the
Ecoli data set with 7 features. 57

3.11 Same as Table 3.3, with RELIEF and CSM to rank features, for the
Wine data set with 13 features. 58

3.12 Same as Table 3.3, with RELIEF and CSM to rank features, for the
Waveform21 data set with 21 features. 58

3.13 The average number of features ($\pm$ std) selected in 10 simulations ,
by class-independent and class-dependent methods with “one-against-
all” strategy, for the Waveform40, Segment and Vehicle data sets. . 60

3.14 Same as Table 3.13, with the CSM and mRMR ranking measures,
for the HDR, DNA and Scene data sets. (FS denotes feature selection) 61

4.1 Comparisons of mean classification accuracy with different measures
for different methods (10-fold cross validation). Original refers to
the classification accuracy without feature selection. 73

4.2 Comparisons of classification accuracies among two feature selection
methods with output probability not weighted and weighted, with
CSM measure and “one-against-all”, for the artificial data A and B. 74

4.3 Same as Table 4.2, for Waveform40, Segment, and Vehicle data sets.
The Table also presents the results by the RELIEF ranking measure
without using the weight approach for those data sets. 75

4.4 Total numbers of selected features and the classification accuracies
for SOAP and RELIEFF methods [144]. 76

4.5 Classification comparisons among different feature selection meth-
ods for the HDR, DNA and Scene data sets. 77

4.6 Feature masks of the first 5 classes 84

4.7 Classification accuracy for variant numbers of features using class-independent feature selection method (10 fold cross validation) . . 84

4.8 Comparisons of classification accuracies with existing methods . . . 84

5.1 Classification results for 3 populations and 4 populations, respectively. 102

5.2 Top ranked features whose appearance frequencies are greater than 83.33% (25/30) in 30 simulations, and their mean ranking values by the F-statistics ranking measure for 3 populations. 104

5.3 Top ranked features whose appearance frequencies are greater than 83.33% (25/30) in 30 simulations, and their mean ranking values by the modified t-test ranking measure for 3 populations. 105

5.4 Top 10 features' ranking lists for the F-statistics ranking measure, for 3 populations and 4 populations. 108

5.5 Same as Table 5.4, top 11 features' ranking list for the modified t-test ranking measure. 108

5.6 Classification Accuracy on the 3 populations and 4 populations, using different feature subsets formed from the ranking list (F-statistics) in Table 5.4 109

5.7 Same as Table 5.6, features subsets are formed form the ranking list (modified t-test) in Table 5.5 110

List of Figures

2.1	Filter approaches to feature selection	22
2.2	Wrapper approaches to feature selection	30
3.1	Schematic diagram for our general hybrid approach to class-dependent feature selection.	48
4.1	Architecture of the general classifier with class-dependent feature subsets (after training and during testing).	69
4.2	Number of features selected for each classes using the class-dependent feature selection method.	83
5.1	Example of haplotypes and genotypes.	97
5.2	Classification accuracy results obtained by the F-statistics measure for different feature subsets with different numbers of top ranked features (SNPs) in 30 simulations, on 3 and 4 populations, respectively.	100
5.3	Classification accuracy results obtained by the modified t-test measure for different feature subsets with different numbers of top ranked features (SNPs) in 30 simulations, on 3 and 4 populations, respectively.	103

Chapter 1

Introduction

In this dissertation, two important assignments, i.e., feature selection and classification, will be studied. The relationship between these two techniques in this dissertation is as follows: feature selection is mainly studied as the crucial pre-processing step of classification, and classification is indicated as the objective of feature selection [1]. Intuitively, feature selection has a strong effect on the classification performance and will be more emphasized in this dissertation.

In this chapter, some concepts related to feature selection like data mining, knowledge discovery in databases (KDD), data dimensionality reduction (DDR) and soft computing are firstly roughly described. Some general definitions and characteristics of feature selection are also presented. Subsequently, the concept of classification is discussed, followed by an example of biomedical data. Thirdly, the motivation and objective of our work are presented. Fourthly, the proposed solution for the problems in this dissertation are presented and the major contributions of the work are also listed. Finally, the organization of the remaining chapters is briefly described.

1.1 Background and Foundation

With the success of the digital revolution and the advancement of internet technologies, various databases including information from text, image, audio, video, to biometrics and biomedical data are easily available online. In order to make use of these data, especially those with huge size, many intelligent tools based on computers, such as machine learning, KDD [2], [3] or data mining [2], [4], [5] are prompted to analyze or process these data. Considering the fact that many of these original data are rarely of direct interest to users, we think it should be our present principle assignment to intelligently “mine” useful information hidden in the tremendous raw data. In this case, the intelligent technologies like KDD and data mining will be mainly concerned in the following part, since they have the ability to discover and mine desirable knowledge for users. The technique of data dimensionality reduction (DDR) is discussed correspondingly. Besides, the backgrounds of soft computing and classification are discussed.

1.1.1 KDD and Data Mining

The term KDD refers to *the process of discovering novel and potentially useful knowledge from large volumes of data* [2]. Data mining denotes *the process of uncovering meaningful and previously unknown information from a mass of data* [4]. KDD is stressed to be an overall process including many techniques, such as data storage, data acquisition, data preparation, data integration, data reduction, data mining, interpretation and visualization of results [2]. At this point, data mining is viewed as a particular step of KDD by some people [2]. Also there are many people who think data mining and KDD are a pair of synonyms [2]. In this dissertation, we think of KDD as the same technique as data mining due to the same objective, i.e., deriving important and useful information from data.

CHAPTER 1. INTRODUCTION

Data mining tasks [5], [6] can be understood in two directions, i.e., knowledge discovery (discovering interesting patterns or features from data) and prediction (predicting or classifying unknown patterns based on the model learning from available data). If combining these two directions, the goal of data mining can be generalized as discovering relationships in data and predicting patterns. With the advent of automatic tools and sophisticated algorithms, data mining has been successfully applied in many fields, such as financial marketing, signal processing, biometrics and bioinformatics, for objectives of clustering, pattern recognition or classification [7] and so on. In this dissertation, we will mainly talk about the application of data mining in classification, a brief introduction of which will be given later in this chapter and more details will be mentioned in Chapter 4. According to [2], the first step of KDD is to understand the data's relevant prior knowledge (e.g., data size and data types) and objectives of the application (e.g., classification). Therefore, before adopting data mining, we need to learn about data examples used in the experiment. Referring to the numbers and the dimension of the data, we group data examples that arouse our interest into the following three types:

- (i) data sets with small size and low dimension, such as the Iris, Breast cancer, Vehicle, and Segment data sets from the UCI machine learning repository database [8].
- (ii) data sets with medium size and dimension, such as the handwritten digital data set (HDR), DNA and Scene-classification data sets from the UCI machine learning repository databases [8].
- (iii) data sets with small size and huge dimension: biometrics data sets like the fingerprint data, voice data and the face image [9], bioinformatics data sets

like DNA sequence, microarray data and the human HapMap genotype data set [10] with only several hundred samples but millions of features.

With the objective of classification on those data examples, our research emphasis is more on developing algorithms to *reduce the dimension of the data*, so that efficient computation and improved classification performance can be obtained. Data dimensionality reduction algorithms are used to deal with the problem of high dimension.

1.1.2 Data Dimensionality Reduction

Data dimensionality reduction [11], [12] is the process of reducing the number of random variables or features (attributes) under consideration. The term “variable” is often used in the field of statistics, while “feature” and “attributes” are two terms which are alternatively used in the fields of computer science and machine learning. From the viewpoint of mathematics, the process of data dimensionality reduction can be stated as follows: given a random variable $x = (x_1, x_2, \dots, x_d)$ with dimension d , we hope to find a representation of x , e.g., $g = (g_1, g_2, \dots, g_{d'})$. We hope that the dimension of g (i.e., d') is less than d . Moreover, g keeps the major contents hidden inside the original data. If there is no new features generated in the process of DDR, i.e., $g_i = x_j$ ($i = 1, 2, \dots, d', j = 1, 2, \dots, d$), this DDR technique is called as feature selection [12], [11], [13]. If g_i is a feature newly generated from original features x_i ($i = 1, 2, \dots, d$), e.g., $g_i = \phi_i(x_1, x_2, \dots, x_d)$ (ϕ_i is a mapping function), this technique of DDR is called feature extraction [13], [11], [12]. Feature selection and feature extraction are two widely used DDR techniques [14], [15]. The survey of these two techniques will be given in Chapter 2.

1.1.3 Soft Computing

Soft computing is a wide topic including neural networks, support vector machine, genetic algorithms, fuzzy sets, rough sets, chaos theory and also their hybridizations [16], [2]. Each soft computing technique has its own way to solve problems in its own domain, however, all those techniques are complementary rather than competitive [16], [2]. The major characteristic of soft computing, i.e., the difference between soft computing and hard computing, is that soft computing has strong tolerance for imprecision, approximate reasoning and uncertainty, and can provide tractable, analytic and low cost solutions for ambiguous problems [2]. All those soft computing techniques have been widely used to deal with data mining problems. For example, fuzzy sets are used in clustering, discovering association rule, time series analysis and so on, to deal with incomplete and imprecise data. Neural networks are robust in various data and have good performance for assignments such as extracting hidden knowledge, clustering and classification. Support vector machines (SVMs) are a very popular and promising tool for bioinformatics data mining, web mining, text mining, classification and regression.

1.1.4 Classification

Classification involves a learning process and predicting process. It is first to learn or train a model based on some known information and then utilize this model to predict or determine the class of unknown data samples. Those known information is mainly known as features (variables) that describe samples or their class labels. In statistics, classification problems can be described as follows: given a set of training pairs $(X_1, y_1), (X_2, y_2), \dots, (X_N, y_N)$, where N is the number of training samples and X_i ($i = 1, 2, \dots, N$) is one training sample, $y_i \in \{1, 2, \dots, C\}$ is the class label of X_i . Training samples are used to generate a classifying model through

CHAPTER 1. INTRODUCTION

a mapping function φ , i.e., $y_i = \varphi(X_i) = \varphi(x_{i1}, x_{i2}, \dots, x_{id})$, where φ denotes the function of the trained model, $x_{i1}, \dots, x_{id}$ denote d features of the sample X_i , namely inputs of the classifying model. Class label y_i is the output of the model.

There are two learning methods to train such a model. If the class information is used in the learning process, it is known as *supervised learning*. Alternatively, if the class information is not used, it is called as *unsupervised learning*, which is specifically fit for the situation where the class information is unknown.

Besides, *prediction and deviation analysis* [4] also can be grouped into two types of classification, both of which have been used in our real life. *Prediction* is often used in time series analysis to determine the future trends according to previously known characteristics of patterns. For example, *regression* has been used to predict weather (e.g., rain or not), temperature or stock trends. *Deviation analysis* is used to make a judgment by analyzing or detecting the change of objects [4]. For the example of credit card fraud detection, deviation analysis is to judge if the credit card is stolen or not by detecting the change of the customer's buying behavior or the change of signature [4]. Some other applications include detecting whether there are suspicious cells in blood samples, determining security systems' fingerprints match or not, identifying army tanks friendly or not from a set of images [4] and so on.

Common classifiers include various neural networks (NNs) (e.g., feed-forward NN (FFNN), back propagation (BP), multi-layer perception (MLP), etc.), support vector machines (SVMs), k-nearest neighbors (KNNs), Gaussian mixture model (GMM), Naive Bayes (NB) classifier, decision tree [4] and so on. The performance of a classifier, to a great degree, depends on the characteristics of the data. Therefore, there is no one classifier which always produces best result on all given data. However, based on existed empirical tests or suggestions, it is possible to choose

CHAPTER 1. INTRODUCTION

a comparative powerful classifier. For example, SVM is very popular in many areas because of its strong performance and successful experience in dealing with classification and regression problems.

1.1.5 Bioinformatics Background for Feature Selection and Classification

Bioinformatics is a technological term, which indicates applying mathematical tools based on computer technologies to solve biological problems, therefore, sometimes it is also called as computational biology [17]. The basic steps of bioinformatics include the determination of an algorithm, the preprocessing of biological data and the realization of the algorithm [18]. Common bioinformatics problems include gene expression analysis, gene finding, genome annotation, DNA sequencing, haplotype/genotype analysis, protein structure prediction, cancer classification and so on. Before dealing with those problems, we will provide some basic biological backgrounds like some common biological terms and some examples of biological problems in the following for better understanding.

From the point of view of biology, *DNA sequence* is a string of four types of nucleotide subunits: adenosine (A), cytidine (C), guanosine (G), and thymidine (T), e.g., ATCTGGCC. Different DNA segments have different functions and those DNA segments which carry genetic information are called *genes*. The total set of genes in a living organism is called the organism's *genome*, which may be stored on one or more *chromosomes*. A *chromosome* is formed from a very long DNA helix on which thousands of genes are encoded. In the human body, there are 23 pairs of chromosomes (i.e., 46 chromosomes), among which 22 pairs look the same for both males and females and the 23rd pair are sex chromosomes, i.e., for females consisting of two X chromosomes and for males consisting of one X chromosome and one Y chromosome. In natural evolution, it is hoped that parents' exact genetic

CHAPTER 1. INTRODUCTION

information is passed on from generation to generation. Whereas, rules have been changed due to the happening of chromosome mutation or genetic mutation, which usually produces fatal or undesirable effects on the new generation. Chromosome mutation or genetic mutation includes four types of basic changes, i.e., the deletion of genes, the duplication of genes, the inversion of genes and the translocation of genes, some of which may change the new organisms' genotype and phenotype in the harmful direction. For the example of an offspring's born, if the mutation changes a very crucial part of the offspring's DNA, abortion may happen. In addition, there are also mutations referring to the variations of nucleotides on a DNA sequence, e.g., the deletion, inversion or reversion of a nucleotide, all of which will change the DNA sequence and thus affect proteins' coding and the like related to the DNA sequence. For example, removing certain nucleotide from a DNA sequence may alter the coding of proteins that use the DNA sequence. So do the insertion of a nucleotide into a genetic sequence. As to the reversion of a particular nucleotide on a DNA sequence, this effect will not be so serious as the previous mutation, partly because only a small portion of the sequence has been affected by the reversion at large. When certain nucleotide is replaced by another one on one DNA sequence, which is called the substitution of a nucleotide, the synthesizing of any amino acid based on this DNA sequence will be affected. Especially if the gene is important, e.g., for coding hemoglobin, then the mutation will lead to serious results, i.e., organisms in this instance suffering from sickle cell anemia. In fact, most of the genetic mutations produce a harmful effect, such as the inheritance of diseases.

Furthermore, these mutations can be used to interpret the difference between different human beings, such as different skin colors, different responses to medicines, different body types and different health status. For example, Daly [19], Gabriel

CHAPTER 1. INTRODUCTION

[20] and Bafna et al. [21] have conducted research on finding genes related to common diseases. They have found causative genes of some common diseases, such as cancer, heart disease and diabetes, which makes it possible for medical researchers to acquire highly effective diagnoses and new drugs' discovery. Among various approaches to find genes of diseases in human populations, a popular one is to statistically study the genetic association between common diseases and DNA variations. Since single nucleotide polymorphisms (SNPs) occupy 80% of human genetic variations, SNPs have been a major research object in the work of genetic association study. Accordingly, haplotypes, as the set of closely linked SNPs on a chromosome, and genotypes, as the set of closely linked SNPs on two chromosomes, have been widely used for the genetic association study. Besides, population studies, another research based on genetic variations, have been developed by Rosenberg et al. [22], [23] and Park et al. [24] to identify populations based on SNPs.

Since data sets of genetic variations usually have the number on the order of 10^6 or above, to carry out research related to these data sets is one of the greatest challenges for us. Bioinformatics data mining has been an efficient tool to reduce the dimension and thereby attain a more efficient calculation. Similarly, soft computing techniques have been successfully used to solve classification or prediction problems. For example, neural networks or SVMs have been used for cancer classification [25], gene expression data analysis [2], [26] or protein structure prediction. Genetic algorithms (GAs) have been used to predict tertiary protein structure [27].

1.2 Motivation of Feature Selection

To build a classical learning system, a set of training samples with a set of features are required. Nowadays data sets that we process are becoming increasingly large, not only in terms of the number of patterns (instances), but also the dimension of features (attributes). Theoretically, more features will make sure more accurate learning systems. However, the real world shows us that it is not the general rule. On one side, many data sets may include some irrelevant or redundant features, the existence of which may degrade the efficiency of most learning algorithms. For example, Langley [28] and Rendell et al. [29] showed that the presence of irrelevant features reduced the predictive accuracy of the learning algorithms. Koller et al. [30] demonstrated that irrelevant or redundant features can have a bad affection on the distribution of relevant features [31]. Langley and Sage [32] showed that the NB learner can not perform optimally on the voting data because of the existence of redundant features. Fu and Wang [33] demonstrated that deleting those irrelevant features can not only improve the classification accuracy, but also reduce the structural complexity of the radial basis function (RBF) neural network and facilitate rule extraction. All these examples tell us that feature selection is necessary in data preprocessing systems to make sure an accurate learning model.

On the other side, more features usually imply that more training samples (instances) are required for building an accurate learning model (classifier) [34]. As a result, more learning time and computational cost will also be needed. In fact, a smaller set of features with finite samples can improve the accuracy of the model [34], [35]. This is especially true when all or most of the class information is involved in a small feature subset [34], i.e., only a minority of information being useful for users. For the example of population classification based on the HapMap genotype data [10], the genotype data has 210 samples and millions of single nucleotide

polymorphism (SNPs) (i.e., features). However, Park et al. [24] has showed that, among millions of SNPs, it is enough for less than one hundred SNPs to completely classify the three populations.

Therefore, in order to improve the efficiency of learning algorithms like reducing the learning time and computation burden, feature selection is a necessary job.

1.3 Objectives of Feature Selection

Feature selection helps us to acquire better understanding about the data by telling us which features are important and how they are related with each other. It alleviates the storage requirements, and helps to improve the performance of learning models in terms of dealing with the curse of dimensionality, speeding up learning process, improving generalization capability and model interpretability [36].

Feature selection reduces the dimensionality of data by selecting discriminative features from original huge numbers of features which include useful, noisy or redundant features. Therefore, feature selection can simplify experimental data, make data more comprehensible and improve data mining performance. When applied on some other areas, such as mathematics, statistics, computer science, engineering and bioinformatics, feature selection helps to improve the generalization ability of many machine learning methods, improve the learning performance of learning systems, build simpler and comprehensible models based on the data. For example, Zhou and Wang [37] proposed a modified t-test ranking method, and utilize it to select a small number of most informative SNPs from original 4 million SNPs for a good classification of three populations. Liu et al. [38] adopted six different feature selection methods on two sets of data (i.e., gene expression profiles and proteomic patterns from ovarian cancer patients). The simulation results

demonstrated that feature selection is a necessary step to improve the classification accuracy and reduce the cost of building classification models.

1.4 Major Contributions of the Thesis

In this dissertation, we firstly proposed a general hybrid approach to selecting class-dependent feature subsets for different classes. This feature selection can be describe in two steps. The first step is to separate the original multi-class problems into several two-class problems. The second step is the combination of filters and wrappers. “Filters” is to utilize a ranking measure to rank features for each class. “Wrappers” is to find a desirable feature subset for each class. Then we proposed a heuristic weight measure to realize the classification of class-dependent feature subsets. Thirdly, we propose a modified t-test ranking measure to rank features with vector values.

In conclusion, the major contributions of the thesis include the following:

- (i) A general hybrid approach to class-dependent feature selection is proposed.
- (ii) A class-dependent SVM classifier is constructed, in which a weight measure is initiated to alleviate the unfair comparison between different feature spaces aroused by class-dependent feature selection.
- (iii) A performance comparison is made on the classification accuracy between the strategy of “one-against-one” and “one-against-all” with class-dependent feature selection.
- (iv) A performance comparison is conducted on the classification accuracy between class-dependent feature selection and class-dependent feature extraction.

- (v) A modified t-test ranking measure is proposed and utilized to deal with SNP data.
- (vi) Both the classifications of 3 populations and 4 populations are realized by a small number of informative SNPs. The ranking performances of the modified t-test and F-statistic measure are compared.

1.5 Organization of the Thesis

This dissertation is organized as follows. In Chapter 2, we review the backgrounds of feature selection and classification, and also existing feature selection methods. In Chapter 3, we mainly describe the general hybrid approach to class-dependent feature selection proposed by us. In Chapter 4, we review the theory background of the support vector machine and proposed a heuristic weight measure to realize the classification of class-dependent feature subsets. Besides, a novel application of the proposed feature selection method on face recognition is demonstrated. In Chapter 5, we review existing research on SNP data and propose a modified t-test ranking measure to rank SNPs. Through the SVM, a few informative SNPs are selected from millions of SNPs to realize a good classification of 3 populations. In Chapter 6, we make a summary about the work we have done and discuss the future work.

Chapter 2

Literature Review

In this chapter, we firstly discuss feature selection about its various definitions in different fields, its characteristics and some related concepts like feature's relevance and feature's redundancy. Then we review feature selection techniques based on various categories, e.g., based on whether the learning algorithm is involved in the feature selection process or not (filters, wrappers and hybrids), or categories based on different evaluation measures and searching algorithms.

2.1 Feature Selection

Among numerous techniques to achieve data dimensionality reduction (DDR) [11], [12], feature selection is a very common one. From different angles, feature selection can have different definitions. For example, Narendra et al. [41], Kira et al. [42], John et al. [43], and Koller et al. [30] gave their definitions from the point of view of machine learning. Dash and Liu [13] summarized four existing definitions, i.e., *classical*, *idealized*, *improving prediction accuracy* and *approximating original class distribution* and gave a comprehensive definition on feature selection. They defined feature selection as the process of selecting a feature subset which has the minimal size, while not leading to any significant reduction of the classification accuracy [13]. In the mean time, the selected feature subset keeps nearly the same

CHAPTER 2. LITERATURE REVIEW

class distribution as the original class distribution over all features [13]. Bins [34] and Yu [44] provided definitions on feature selection from the point of view of mathematics. As Bins described [34], assume that a set of samples with dimension d are denoted as S^d ($S^d \subset \mathbb{R}^d$) and each sample X^d has d features. If we select k features ($k < d$) from the original d features, there will be $\binom{d}{k}$ possible combinations, all of which form the feature set F^k ($F^k \subset \mathbb{R}^k$) and one of which is denoted as f^k . Let t represent the target function, t' represents the approximation function of t and $E_f(t)$ represents the expected value of the function t for features in the feature set f . The goal of feature selection is to select k features (i.e., f_0^k) which makes the expected value of the difference between the target and prediction function among all possible combinations of size k minimal, i.e., $E_{f_0^k}(|t(X^d) - t'(X^k)|) = \min\{E_{f^k}(|t(X^d) - t'(X^k)|)\}, \forall f^k \subset F^k$. Yu [44] defined feature selection as the process of maximizing a selection criterion J on the selected feature space.

Besides being called as feature selection, feature selection is also known as attribute selection, feature reduction, or variable subset selection. In machine learning domain, the term “attribute selection” is mostly used. In statistics, feature selection or subset selection is often used alternatively. However, in bioinformatics domain, feature selection has various names due to applications on various biological data. For example, when detecting powerful genes from biological data like DNA microarray data [45], [46], the process of selecting genes is called as “discriminative gene selection” [46]. When selecting most informative single nucleotide polymorphisms (SNPs) from the HapMap genotype data [10], the term “tag SNP selection” [47], [48] is used instead. No matter in which domains, the essence and objective of feature selection are the same, i.e., selecting relevant or informative features to make the subsequent research [49] more efficient.

Both feature selection and feature extraction can be used to reduce data dimensionality, they are rather different. Feature extraction [50], [49] is defined as

CHAPTER 2. LITERATURE REVIEW

transforming the original features into a set of features in a new feature space. The transformation can be linear or non-linear and the new feature space is expected to have lower dimension than the original feature space. The most typical example of feature extraction is principal component analysis (PCA) [51], [50], [52], [53], [49]. PCA firstly calculates a covariance matrix from the input data. Then PCA calculates the eigenvalues and eigenvectors of the covariance matrix and sorts all eigenvalues in a descending order. Subsequently, PCA takes a predefined number of components (eigenvectors) to form a transformation matrix. Although feature extraction (e.g., PCA) is an efficient data dimensionality reduction technique, the newly generated features have different meanings from the original features. Therefore, it is hard for us to interpret the meaning of newly formed features. Also because of this fact, feature extraction is less adopted than feature selection.

Feature selection does not transform the original features. It is the process of eliminating irrelevant or redundant features and leaving the best subset of features which retains sufficient information so as to discriminate well among classes. For feature selection, at first we need to know what is the meaning of relevancy and redundancy. Then we need to determine features' relevancy and redundancy. At last, we need to remove those irrelevant or redundant features. In the following two subsections, we will discuss features' relevancy and redundancy.

2.1.1 Feature Relevance

In order to understand the meaning of "relevance" in the context, we have to deal with the problem of "relevant to what" [54]. On this point, Blum and Langley provided several different definitions in [54], such as "relevant to target", "relevant to sample" and "relevant to distribution". Among the three definitions, "relevant to target" is the simplest form of relevance [54], which determines a feature's

CHAPTER 2. LITERATURE REVIEW

relevance based on the target concept. If the target concept is classification, we determine whether a feature is relevant or not by investigating whether changing the feature will affect the classification of some existed samples in the instance space. A “relevant” feature will change the classification of those samples. Raman et al. [31] also listed three kinds of feature relevance from empirical models. For example, “relevant to prediction accuracy” refers to finding features which are relevant to optimizing the prediction accuracy of the learning algorithms [31]. In this case, the notion of “relevant to prediction accuracy” is the same as the notion of “relevant to the target”. Since the target concept in this dissertation is classification, we will select relevant features based on the concept of “relevant to target” or “relevant to prediction accuracy”. Sometimes, the degree of relevance is also mentioned. For example, John et al. [43] and Liu [50] described relevance in terms of two degrees for better understanding, i.e., strong and weak. By definition, whether features are strongly relevant or weakly relevant, they have contributions to the prediction accuracy. In most real applications, we use ranking measures [36] to evaluate features’ relevance. The higher the ranking values of features, the more relevant features. Those top ranked features should be strongly relevant and will be selected most frequently. On the contrary, irrelevant features should be removed since they have no contributions to the prediction accuracy.

2.1.2 Feature Redundancy

To do efficient feature selection, it is not enough for us to select only relevant features, since the selected feature subset may still include redundant features [55], [36]. Therefore, another concept related to features, i.e., redundancy [36], is also worthy of consideration. Redundancy measures the similarity or dependency [56], [40] among features. Highly correlated features have high redundancy and

CHAPTER 2. LITERATURE REVIEW

the inclusion or deletion of those features will not affect the prediction accuracy. Some evaluation measures like correlation coefficient (ρ) [57], [56], least square regression error (e) [57], [56], maximal information compression index [56], mutual information [58], [40] and so on, have been used to detect the redundancy of features. By those measures, highly correlated features should be grouped together and some of them should be removed to reduce the size of the feature subset. Many existing feature selection algorithms did not measure the redundancy of features, such as RELIEF, RELIEFF [39] and Focus [59]. However, there are some authors like Mitra et al. [56], Kwak et al. [58] and Peng et al. [40] who proposed to remove redundant features in their feature selection algorithms.

2.1.3 Optimality of Feature Subsets

For any one given task, if want to find the optimal or approximately optimal set of features, an exhaustive search for all possible features subsets is necessary. Whereas, considering computation time, this seems very impractical, especially when the number of features is very large. These double requirements have made problems related with feature selection NP-hard [60], [61]. Therefore, we tend to find a desirable feature subset only that there is a balance-off between the quality of the subset and the computation time.

2.2 Feature Selection Methods

As a conventional research topic, the history of feature selection or subset selection [50] can be dated back to as early as the 70's [62], [63]. While for the categorization of feature selection, as yet there is not a unit standard, although many attempts have been made. For example, Siedlecki et al. [64] classified feature selection methods into past, present and future based on the development

CHAPTER 2. LITERATURE REVIEW

of the branch and bound (BB) method [65] and its variants. Dash and Liu [13] provided another kind of categorization scheme in terms of two essential steps of feature selection: generation procedure and evaluation function. As to generation procedure [13], feature selection includes three branches, i.e., complete, heuristic and random search strategies. As to evaluation function [13], there are 4 types of commonly used functions: distance, correlation, consistency and classification error rate. Based on the taxonomy of Dash and Liu [13], Li [66] provided a more detailed categorization scheme by adding one branch of individual feature ranking into generation procedure and grouping the four types of evaluation functions into filter and wrapper models. Based on different classification techniques adopted in feature selection, Jain and Zongker [67] classified feature selection algorithms into the one based on artificial neural networks and the one based on statistical pattern recognition (SPR) which can be further classified into some sub-categories [44]. Based on whether the class labels of samples are given or not, there are two broad categories: supervised and unsupervised feature selection [56], [68]. Since different feature selection methods have different time complexity, there is also a categorization based on the time complexity. For example, heuristic search methods like sequential forward (or backward) search (SFS) [69] and beam search [70] belong to the quadratic complexity ($O(D^2)$) [13]. Complete search methods like the sequential floating search [71], [56] belong to the time complexity of $O(2^D)$. Some other local categorizations, such as probabilistic search algorithms (e.g., random-hill climbing [72] and Las Vegas Filter (LVF) [73]) vs. non-probabilistic search algorithms, are made from only search strategies adopted. Forward selection vs. backward elimination is simply based on the starting point of the search process [44].

In fact, feature selection is commonly categorized [74], [50] into filter approaches [75], [42] and wrapper approaches [69], [43]. This category is based on whether the

feature selection process includes any learning algorithm or not. Filter approaches select a feature subset based on the characteristics of data itself. On the contrary, wrapper approaches have learning algorithms as the “black box” and each candidate feature subset will be evaluated by the “black box” so as to find the one with the best classification accuracy. In this chapter, we will discuss feature selection methods according to this common categorization together with two special groups: *class-dependent* and *class-independent* feature selection. The two special groups are based on whether the feature subset selected is dependent on the class label or not (see Table 2.1). The group of class-dependent feature selection is also known as class-specific feature selection [76], [77], [78].

2.2.1 Filter Approaches

Filter approaches (Figure 2.1) are the earliest kind of feature selection methods [44]. They are also seen as preprocessing before any one learning method (classifier) is involved. Some filter approaches like the RELIEF algorithm [42] and its extended version RELIEFF [39] first evaluate each feature’s relevance in terms of the intrinsic properties of the data and then select features according to a predefined threshold (see the fifth row of Table 2.1). The RELIEF [42] and RELIEFF [39] algorithms assign a weight to each feature and then update the weight according to distance characteristics of training instances. These weights imply features’ relevance. All features are ranked according to their weights. Features with weights above a predefined threshold are selected, whereas, features with weights below the threshold are deleted. In fact, this weight measure is an individual feature ranking strategy. Instead of using the ranking strategy, some filter approaches attempt to find a proper feature subset through various search strategies (see the 7th-9th rows of Table 2.1), e.g., branch and bound [41], [65], [99], best-first search [84], Focus

CHAPTER 2. LITERATURE REVIEW

Table 2.1: Categorization of feature selection methods.

Dependent on learning algorithms or not			
Feature selection methods	Based on measures		Examples
Filter Approaches	Ranking Measures	Probabilistic measures	Bhattacharyya's and Kullback-Liebler's distance measures [79] [34], Matusita's, Joshi's and Kolmogorov's probabilistic dependency measures [79] [34]
		Entropy measure	Information theoretic ranking criterion [80] [81], Peng et al.'s mRMR [40], Chow et al.'s [82]
		Interclass distance measure	RELIEF [42], RELIEFF [39], CSM [33]
		Consistency measure	Focus [83] and Lashkia et al.'s [55]
	Search Strategies	complete search	B&B [41], BFS [84] and Focus [83]
		heuristic search	DTF [85], cross-entropy filter [30]
		random search	LVF [86]
Wrapper Approaches		heuristic search	SFS [69], SBS [69], BSE-SLASH [87], BDS [79]
		random search	GA [88], SA [89]
		complete search	Beam search [79]
Hybrid Approaches			Hybrid [90], wrapper-filter [91]
Dependent on class information or not			
class-dependent	[92], [76], [77], [78], [93], [94], [95], [96], [97]		
class-independent	[42], [83], [39], [33], [98], etc.

Figure 2.1: Filter approaches to feature selection

[83], the decision tree filter [85] and cross-entropy filter [30]. These representative filter schemes will be introduced in the following.

2.2.1.1 Filter Approaches Based on Ranking Measures

Various kinds of evaluation functions are used as ranking measures in filter approaches to measure the relevance of a feature or set of features (Table 2.1)). Ranking measures or evaluation functions are based on the intrinsic properties of the data. It is implicitly assumed that the relevance of features is positively correlated with the accuracy it produces, i.e., the more relevant the features, the better the accuracy obtained. Although Kohavi et al. [74] have pointed out that the assumption is not always true, in most cases it is an enough approximation. However, it also means that there is not any one single evaluation function which can always work best for all applications. Doak [79] grouped the filter evaluation functions into four categories: probabilistic distance measures, probabilistic dependence measures, interclass distance measures and entropy measures [34]. Dash

CHAPTER 2. LITERATURE REVIEW

and Liu [13] also categorized filter evaluation functions into four types: distance measures, information measures (e.g., entropy measures in [79] [34], information-theoretic ranking [100]), dependency measures and consistency measures, in which distance measures can be seen as the combination of interclass distance measures and probabilistic distance measures in [79]. In the following, we will discuss those filter evaluation functions based on the two existing categorizations [79] [13].

- (i) Probabilistic Measures. Probabilistic measures can be divided into probabilistic distance measures and probabilistic dependency measures [79] [34]. Probabilistic distance measures are functions of two elements, i.e., the conditional probability of the feature vector F (i.e., $P(F|c)$) and the probability of class c (i.e., $P(c)$). Probabilistic distance measures calculate the probabilistic distance of a single feature or a feature subset between different classes [79]. For example, Bhattacharyya [79] [34] evaluated the relevance of a feature or feature subset F by the first function in Table 2.2, where c_1 and c_2 denote two classes. Kullback-Liebler [79] [34] utilized the second function in Table 2.2 to measure the relevance of the feature subset F . Kolmogorov [79] utilized the third function in Table 2.2 to measure the probabilistic distance. From the three different functions, we can see that J will be zero if the probabilities of the feature or feature subset F under two different classes are equal. In this case, F has no use for distinguishing class c_1 from c_2 . When the class conditional probability has great variations between different classes, J has a large value ($J \leq 1$) and the feature or feature subset will be kept. In a word, the bigger the value of J , the stronger the feature subset's distinguishing power. Probabilistic dependency measures are also called correlation measures or association measures [34]. They are functions of conditional probability of

CHAPTER 2. LITERATURE REVIEW

Table 2.2: Probabilistic measures.

Measure examples		Measure functions
Distance Measure	Bhattacharyya's	$J(F) = -\log \int \sqrt{P(F c_1)P(F c_2)}dF$
	Kullback-Liebler's	$J(F) = \int [P(F c_1) - P(F c_2)] \log \frac{P(F c_1)}{P(F c_2)} dF$
	Kolmogorov's	$J(F) = \int P(F c_1) - P(F c_2) dF$
Dependency Measure	Matusita's	$J(F) = \sum_{i=1}^C [P(c_i) \sqrt{\int (\sqrt{P(F c_i)} - \sqrt{P(F)}) dF}]$
	Joshi's	$J(F) = \sum_{i=1}^C [P(c_i) \int [P(F c_i) - P(F)] \log \frac{P(F c_i)}{P(F)} dF]$
	Kolmogorov's	$J(F) = \frac{1}{2} \sum_{i=1}^C [P(c_i) \int P(F c_i) - P(F) dF]$
	Patrick-Fisher's	$J(F) = \sum_{i=1}^C P(c_i) \int \sqrt{P(F c_i) - P(F)} dF$

feature vectors $P(F|c)$ and the probability of the feature vector $P(F)$, which makes dependency measures different from distance measures. Examples of probabilistic dependency measures [34] (see Table 2.2) include Matusita's, Joshi's, Kolmogorov's and Patrick-Fisher's. For all those expressions in Table 2.2, c_i denotes the i -th class and C is the number of classes. $P(F|c_i)$ denotes the probability of the feature subset given the class c_i . From these functions we can see that if the feature subset is independent on classes, the feature subset has no classification power. If the feature subset has great dependency on classes, the value of J will be great. Therefore, we can rank feature subsets based on the rule that the more different the two probabilities, the more dependent the relationship between feature subsets and classes, and thereby the stronger the distinguishing power of the feature subset [34].

- (ii) Entropy Measures. Entropy measures are also known as information measures [13]. Different from previous probabilistic measures which use the a priori probability, entropy measures adopt the a posteriori probability $P(c_i|F)$

CHAPTER 2. LITERATURE REVIEW

in its evaluation function. One classical example of entropy measures is Shannon's $J(F) = -\sum_{i=1}^C P(c_i|F) \log P(c_i|F)$, which is based on the gain of the feature subset F for class c_i . The greater the entropy measure, the more dependent the class and the feature vector. Information theoretic ranking criterion (ITRC) (see Table 2.1) is another example of entropy measures used in feature selection by [80], [81]. The mutual information between each discrete feature and the target is described as follows:

$$J_i = \sum_{x_i} \sum_{c_i} p(X = x_i, c = c_i) \log \frac{p(X = x_i, c = c_i)}{p(X = x_i)p(c = c_i)} \quad (2.1)$$

Here, J_i measures the statistical dependency between the feature x_i and the target c_i and thereby is used to record the relevancy of the feature x_i . $p(X = x_i)$ denotes the probability distribution of the feature x_i and $p(c = c_i)$ denotes the class prior probability. $p(X = x_i, c = c_i)$ is the joint probability distribution. The calculation will be harder if the number of classes and variables are greatly increased, especially if input features or target are continuous [36]. Based on the theory of mutual information, Peng et al. [40] proposed the minimal-redundancy-maximal-relevancy (mRMR) method (see Table 2.1) to evaluate each feature about their relevance and also redundancy. More details about the mRMR method will be given in the next chapter.

- (iii) Interclass Distance Measures. Interclass distance measures are intended to calculate average pairwise distance between training examples of different classes for different feature subsets. The common examples of interclass distance measures include the Euclidean distance and Mahalanobis distance. The Euclidean distance of two samples (X_i and X_j) is described as $J(X_i; X_j; F) = \sqrt{\sum_{X_{im}, X_{jn} \in F} (X_{im} - X_{jn})^2}$. The Mahalanobis distance is

CHAPTER 2. LITERATURE REVIEW

described as $J(F) = (m_1 - m_2)^T \Sigma^{-1} (m_1 - m_2)$. Here m_1 and m_2 refer to mean values of training samples for two classes, respectively, over the feature subset F . Σ^{-1} denotes the inverse of Σ , i.e., the covariance matrix of two classes over features in F . The evaluation rule is that the bigger the distance, the better distinguishing power the feature subset has. The well-known filter feature evaluation measures, i.e., RELIEF and its variants RELIEFF (see Table 2.1), are based on the Euclidean distance of training samples to measure the relevancy of features. Class separation measures (CSM) [33] are based on the interclass distance and between-class distance to evaluate each single feature or feature subset.

- (iv) Consistency Measures. Consistency measures are mentioned in [13] [55] [101], which are intended to find a feature subset with the minimal size and attributes in the feature subset are consistent with the training set. Focus [83] is a representative filter approach based on the consistency measure. Lashkia et al. [55] proved the consistency of attributes directly reflects the relevance of attributes and thereby utilized the notion of consistency to select relevant attributes.

2.2.1.2 Filter Approaches Based on Various Search Strategies

Instead of using evaluation functions to rank a single feature or feature subset, filter approaches also select features by diverse search strategies, which can be grouped into three types: complete, heuristic and random search [13], [66] (see examples in Table 2.1). For the feature subset with D features, the original way is to search all 2^D feature subsets and determine the best one based on some evaluation criteria. This search is known as the exhaustive search, the search complexity of which becomes very intractable with the increase of D . Therefore, other complete, heuristic and random searches are put forward to reduce the search complexity.

CHAPTER 2. LITERATURE REVIEW

- Complete Search Methods

Exhaustive searches belong to complete searches, while a complete search is not always exhaustive [102], [13]. Dash and Liu [13] summarized several typical complete search examples (see Table 2.1), e.g., the branch and bound (B&B) algorithm [41], the best first search (BFS) [84] and Focus [83], which are different in the adoption of evaluation functions. The B&B is a very old method which utilizes distance measures like the Mahalanobis distance as the evaluation function [41]. In the search space with the order 2^D , fewer feature subsets are evaluated by setting a bound on the value calculated from the evaluation function. The BFS [84] also adopts the distance measure as the evaluation function. BFS utilizes nodes to represent feature subsets and each node's state in the search space consists of a string with 'D' bits for 'D' features, where '1' indicates the feature is present and '0' indicates the feature is absent. All states are connected by compound operators like addition or deletion of a single feature. The node with the best score will be always selected. The complexity of the search space is $O(2^D)$, and therefore the node search will be stopped if the score after 'k' attempts degrades, where 'k' is predefined by users. Focus [83] is another representative work of complete search but with a different evaluation function (i.e., consistency measure). It is originally designed for the boolean domain [83], in which Focus algorithm attempts to find a feature subset with minimum size and perfect discrimination power [54]. This method starts the search from each single feature, then goes on to pairs of features, triples, and so forth. Each subset generated thus will be checked by the consistency measure. Here, the subset is said to be consistent if no two instances with this subset have different class labels. The search will not stop until it finds a minimal and sufficient feature subset which is consistent [31].

CHAPTER 2. LITERATURE REVIEW

For above complete search methods, different heuristic functions are adopted to reduce the computation complexity while not at the expense of losing the optimal subset. Nevertheless, in case of data with high dimension, a complete search is not the ideal choice.

- Heuristic Search Methods

The representative work of filter approaches in terms of heuristic search includes decision-tree filter [85] (DTF), cross-entropy filter [30] and so on. The decision-tree filter starts from an empty set and features are evaluated by information gain of the training set [85]. Suppose a two class problem has the training set S and feature x , the information gain [103] of the feature x relative to S is defined as follows:

$$Gain(S, x) = Entropy(S) - \sum_{v \in values(x)} \frac{|S_v|}{|S|} Entropy(S_v) \quad (2.2)$$

where ‘ $values(x)$ ’ indicates the set of all possible values of x and S_v is the subset of S among which x has value v . $Entropy(S)$ is the entropy of S [103]. The second item in Eq. (2.2) denotes the sum of weighted entropies of each subset S_v . $|S_v|/|S|$ is the weight denoting the proportion of the number of samples for x with value v over the total number of samples. Given the number of classes C and the proportion of each class p_i , we have the following equation:

$$Entropy(S) = \sum_{i=1}^C -p_i \cdot \log_2 p_i \quad (2.3)$$

According to the information gain measure, the larger the value of $Gain(S, x)$ is, the better class discrimination the attribute x provides. The decision tree filter selects those features that appear on the pruned decision trees [103], [104], [13]. Cardie [85] adopted decision tree features with a k-nearest

CHAPTER 2. LITERATURE REVIEW

neighbor to solve a natural language processing task. The results show the decision tree feature subset generated can help the k-nearest neighbor to improve its prediction accuracy [31], [85].

Unlike the decision-tree filter, the cross-entropy filter [30] starts with all features. The cross entropy is to find a feature subset F' which makes $P_r(c|F' = f_{F'})$ close to $P_r(c|F = f)$. Here c is one of classes and F is the original feature set. $P_r(c|F = f)$ is the probability that assigns an unlabeled instance to a particular label in $\mathcal{C}$ (class set) with feature values f assigned to F . $f_{F'}$ refers to assigning f to the feature subset F' . If we use μ to denote the distribution of the original feature subset and δ to denote the approximated distribution of the selected feature subset, the cross entropy is defined as: $D(\mu, \delta) = \sum_{x \in \mu} \mu(x) \log \frac{\mu(x)}{\delta(x)}$, which measures the extent of “error” when using an approximated distribution to substitute the original distribution. The optimal feature subset is the one (F') obtained by minimizing $\sum_f P_r(f) D(P_r(c|F = f), P_r(c|F' = f_{F'}))$.

- Random Search Methods

Randomized algorithms have the advantage of protecting the search from converging to local minimum caused by the sequential searches [31]. Las Vegas filter (LVF) [86] is one example of randomized algorithms. LVF utilizes a Las Vegas algorithm [86] to randomly search the feature space. The search is guided by a probabilistic choice generated by Las Vegas algorithm and the selection of features is based on a consistency measure. LVF can always find a good subset as soon as quickly, whereas it will not be more efficient than those heuristic search algorithms to find the optimal feature subset [13].

Figure 2.2: Wrapper approaches to feature selection

2.2.2 Wrapper Approaches

Intuitively, wrapper approaches (Fig. 2.2) are to “wrap” feature selection around a learning algorithm. Wrapper approaches utilize some heuristic search techniques (see Table 2.1), such as sequential forward and backward search [32], [49], hill climbing [87], best-first search (Kohavi et al. [105] as cited in [54]) and beam search (Aha et al. [70] as cited in [54]) to search for possible feature subsets, and use a learning algorithm (classifier) to evaluate those feature subsets and determine the optimal one in terms of the classification accuracy. For example, Langley and Sage [32] adopted the search strategy of backward elimination with the help of the simple nearest neighbor classifier to find the proper feature subset. The algorithm begins with all features, each time removing one feature from the whole set and evaluating the performance of the current feature subset by the nearest neighbor classifier. The feature whose removal leads to the best learning performance is finally gotten rid of. This process is repeated until no improvement on accuracy can

CHAPTER 2. LITERATURE REVIEW

be obtained. Caurana and Freitag's wrapper approach [87] uses hill climbing as the search strategy and the decision tree as the learning algorithm. Compared to filter approaches, wrapper approaches are computationally more expensive due to the introduction of a learning machine (classifier). But for the same reason, wrapper approaches usually lead to better classification accuracies than filter approaches. In order to strike a balance between the computation speed and the classification accuracy for wrapper approaches, many researchers have tried to reduce the evaluation time of learning algorithms. For example, Caurana and Freitag [87] adopted the method of caching decision trees. Moore and Lee [106] tried to speed up the evaluation process by reducing the number of training instances. In real applications, if the number of features or instances is not very large, the usual choice is a wrapper approach so as to achieve better classification accuracy.

2.2.2.1 Wrapper Approaches Based on Various Search Strategies

According to [31], wrapper approaches are realized by various search strategies. Same as the category of filter approaches based on various search strategies, wrapper approaches can be grouped into heuristic, complete and random search strategies. The most widely used two heuristic search methods are sequential forward selection (SFS) and sequential backward selection (SBS), which were introduced by Devijver and Kitler [69]. These two sequential selection methods add or remove one feature at a time. Therefore, wrappers based on sequential search methods are efficient in computation, whereas they easily get trapped in local minima. To tackle this problem, random search methods, such as the genetic algorithm (GA) [88] and simulated annealing (SA) [89], are put forward. As to examples of complete search, we will only discuss beam search [79], [13] here.

CHAPTER 2. LITERATURE REVIEW

- (i) **Heuristic Search.** Sequential forward selection (SFS) starts with an empty set and adds one feature at a time until all features are added. The feature will be selected if adding it to the previous subset most increases the performance of the learner (learning algorithm). Given a feature vector with dimension D , this process will repeat $D - 1$ times. In the whole process, there will be $D(D + 1)/2$ feature subsets formed and trained. Sequential backward selection (SBS) is also called sequential backward elimination. Different from SFS, SBS starts with the full set of features and removes one feature at a time. The feature whose removal each time most improves the target learner's performance will be eliminated. The process will be repeated $D - 1$ times and there are $D(D + 1)/2$ feature subsets trained. Since these two sequential search methods never turn back to previous decisions about including or eliminating variables for making new decisions, Guyon also called them nested subset methods that perform a greedy search [36]. SBS is said to be more expensive than SFS if stopped earlier [36]. However, for a given data set without prior information on feature correlation, it is not clear whether one selection method will definitely outperform the other one [31]. Besides, there are some other heuristic search methods for wrappers, which are variants of SFS or SBS, or both [13]. For example, backward stepwise elimination-SLASH (BSE-SLASH) [87] is changed from SBS and thereby has the same starting point as SBS. Bi-directional search [79] (BDS) allows simultaneous search from both ends, i.e., forward from an empty set and backward from the full set, and stops with success when two directional searches meet. In Table 2.3, we compare the typical heuristic searches with exhaustive searches in terms of computational complexity. Wrappers based on SBS and SFS both need to evaluate $D(D + 1)/2$ feature subsets and the computation complexity is $\log(D(D + 1)/2)$, i.e., close to $\log D$, whereas wrappers based on the

CHAPTER 2. LITERATURE REVIEW

Table 2.3: Comparison of search methods.

Search methods	Starting point	Number of feature subsets	Computation complexity
Exhaustive search	any	2^D	$O(2^D)$
Wrappers' SFS	empty set	$D(D + 1)/2$	$O(D^2)$
Wrappers' SBS	full set	$D(D + 1)/2$	$O(D^2)$

exhaustive search have computation complexity D and are computationally intractable for a large number of variables.

- (ii) Complete Search. For complete search in wrappers, the beam search [79], [13] is one of the examples. The beam search works like best first search but stores only the best “n” nodes at each search step, where “n” is a fixed number called the “beam width”. Each node in the beam search denotes a unique path from the starting state to the current node’s state. In the search processes, the node with the higher cost will be replaced by a new node with the lower cost. The objective node would be the complete path with the least cost. If not setting bounds in space, the beam search becomes exhaustive. If the bound becomes one, it becomes forward selection search. If the bound becomes infinite, it becomes best first search. Beam search is proved to be extremely appropriate for data sets with a small number of samples and large number of features [70].
- (iii) Random Search. Genetic algorithm (GA) [88] firstly defines two elements: the genetic representation of solutions (individuals) and fitness function to evaluate solutions. The standard representation of solutions is an array of bits where ‘1’ denotes a feature was selected for the newer population and

CHAPTER 2. LITERATURE REVIEW

a ‘0’ otherwise [79]. The fitness function is problem dependent. GA begins with a random initial population and improves the population by evolution operators like mutation, crossover, inversion and selection. Each newly generated population will be evaluated by the fitness function and used for next iteration of the algorithm. The evolution will stop when the population with a satisfactory fitness level is found or the number of iterations has been in excess of a predefined threshold.

All these wrapper approaches are computationally expensive but provide greater increases to prediction accuracy than filter schemes. This is mainly due to the fact that they include the bias of the target learning algorithm. Due to this reason, both Doak [79] and John [43] argued in support of using a wrapper method rather than a filter method for improved performance. In addition, in order to take advantage of the two feature selection approaches, some authors proposed the hybrid approaches to feature selection [90] [91] by combining filters and wrappers together. This hybrid approach can be seen as one more branch of the common category (Table. 2.1).

2.2.3 Feature Selection Based on Class Information

Rather than the above categorization between filters and wrappers, which is based on whether or not the feature selection process is dependent on the learning algorithm, feature selection methods may also be grouped according to the class information (see Table. 2.1), i.e., class-independent feature selection [42], [33], [98] and class-dependent (or class-specific) feature selection [92], [76], [77], [78], [93], [94], [95], [96]. This category is based on whether or not the selected feature subset depends on classes. Class-independent feature selection chooses a feature subset without regard to the class in a given classification problem. Most of the feature

selection methods, filters and wrappers mentioned in the previous paragraph [42], [83], [39], [33], [98] belong to the category of class-independent feature selection. They have not considered the possibility that different groups of features may have different power in distinguishing different classes [93], [94], [95]. Class-dependent feature selection chooses feature subsets with regard to the class information, i.e., different feature subsets for different classes. Class-dependent feature selection [92], [76], [77], [78], [93], [94], [95], [96] can theoretically improve upon or at least match the performance of class-independent feature selection, because the latter can be considered as a special case of the former. In order to take advantage of the possibility that different features have different classification power and demonstrate it in practice, we propose to use the concept of the class-dependent feature selection [97].

2.3 Summary

We emphasize that both wrapper and filter approaches take the assignment of feature selection, but have their own characteristics. By comparison, the advantages and disadvantages of both approaches are summarized: filter approaches are computationally less expensive than wrapper approaches, but they may return larger feature subsets and ignore the effects of the selected feature subset on the performance of the learning algorithm. As to wrapper approaches, because of taking into account the biases of the learning algorithm, they provide better prediction accuracy on the learning algorithm than filter schemes. But wrapper approaches are computationally more expensive. They obviously increase the overall running time of the black-box induction algorithm by a multiplicative factor of $O(D^2)$ in the worst case.

CHAPTER 2. LITERATURE REVIEW

When deciding between filter approaches and wrapper approaches, we have to be aware of their advantages and disadvantages so as to be able to choose the right approach from the angle of ultimate goals and requirement, concerning the task to be solved. Another noticeable observation from these works is that there is no algorithm that performs optimally on all domains, as shown by variability in experimental results [50]. This is understandable as feature selection is a highly domain specific task.

Chapter 3

Feature Selection: A General Hybrid Approach to Class-dependent Feature Selection

Since the data sets that we process are becoming increasingly larger in the number of patterns and features or attributes, data preprocessing has already been a very necessary step for us to reduce the computation complexity, to save computational cost and also to improve the efficiency of many learning algorithms used on the data. Data preprocessing techniques can be classified into two categories: sample selection [54] to reduce the number of patterns and data dimensionality reduction [50], [36], [49] to reduce the dimensionality of the features (e.g., feature selection and feature extraction). In the majority of cases, the work of sample selection is not necessary because the number of samples that we process is not large itself. However, the work of feature selection or feature extraction is a necessary challenge due to the prevalence of high dimensional data with some irrelevant or redundant features.

In the previous chapter, we reviewed two widely used feature selection methods, i.e., filters and wrappers, and also the combination of them, i.e., hybrids. In this chapter, we introduce our proposed general hybrid approach to class-dependent

CHAPTER 3. FEATURE SELECTION: A GENERAL HYBRID APPROACH TO CLASS-DEPENDENT FEATURE SELECTION

feature selection. Here, the term “general” indicates the proposed method is a framework which can accommodate various feature selection techniques, such as various filters based on different ranking measures, various wrapper approaches based on different classifiers and different search measures, or combination of filters and wrappers. Whatever feature selection techniques are adopted in the framework, the emphasis of our method will be “class-dependent”. Let us consider the following scenario.

Suppose we have data on patient information, such as age, body weight, height, blood pressure, cholesterol, race, etc, and we need to use these data for diagnosing of three types of diseases, say A, B and E. Suppose there are underlying scientific reasons (not yet known to the world) that (1) a patient has disease A if and only if the body weight of the patient is above a certain limit; (2) a patient has disease B if and only if the blood pressure of the patient is below a certain limit; (3) a patient has disease E if and only if the cholesterol is within a certain range; and (4) all the other patient attributes are irrelevant to this diagnostic task. For this 4-class classification problem, with the 4 classes being disease A, disease B, disease E, and healthy (for the sake of simplicity, we assume that it is impossible to have more than one of these diseases simultaneously), one may train either one four-class classifier, or four 2-class (binary) classifiers, each of which discriminating one class from the other three classes. Suppose an effective feature selection technique is used. The single four-class classifier approach should yield three input features, i.e., body weight, blood pressure and cholesterol, whereas the approach with four 2-class classifiers should yield body weight for the disease-A classifier, blood pressure for the disease-B classifier, cholesterol for the disease-E classifier, and body weight, blood pressure and cholesterol for the healthy classifier. All the other input attributes are noise and should have been removed by the feature

CHAPTER 3. FEATURE SELECTION: A GENERAL HYBRID APPROACH TO CLASS-DEPENDENT FEATURE SELECTION

selection algorithm. We say that the four-class classifier has *class-independent* input features, whereas the approach with four 2-class classifiers has *class-dependent* input features. In general, we have the following reasons to expect the approach with four 2-class classifiers (using *class-dependent* feature subsets) to out-perform the single four-class classifier (using *class-independent* feature subsets): If an input data represents a patient with disease A, the blood pressure and cholesterol part of the data will act as noise to the multi-class classifier, whereas if an input data represents a patient with disease B, the body weight and cholesterol part of the data will act as noise to the multi-class classifier. In contrast, there will be no noise for the approach with four 2-class classifiers (we will later demonstrate this line of arguments with simulations). To deal with this scenario, class-dependent feature selection is an appropriate way.

In the following, we will describe the framework of our method and some possible techniques used in the framework for comparison.

3.1 Hybrid Approaches between Filters and Wrappers

Since one of the objectives of our feature selection is to improve the classification accuracy, wrapper methods are our choice. However, pure wrapper approaches are generally computationally expensive. Therefore, some hybrid approaches have been proposed instead [90] [91], which take advantages of filters' fast speed and wrappers' good classification performance. For example, Legrand [90] proposed a hybrid feature selection method in two stages. In the first filter stage, Legrand [90] proposed to use preferences aggregation to determine an ordered list of features subsets. Then in the second wrapper stage, Legrand [90] used a learning algorithm to evaluate all feature subsets. Zhu et al. also [91] proposed a filter-wrapper

CHAPTER 3. FEATURE SELECTION: A GENERAL HYBRID APPROACH TO CLASS-DEPENDENT
FEATURE SELECTION

approach to select features. For the filter part, Zhu et al. utilized a filter ranking method in the traditional genetic algorithm, which can accelerate the search process in identifying important feature subsets [91]. The filter ranking method is based on an univariate feature ranking information to realize the including or removing of features, and thereby refine the population of GA solutions. The genetic algorithm is used in the wrapper part. The result [91] showed that the proposed filter-wrapper approach to feature selection outperformed pure wrappers like GA in the classification accuracy with a small number of features. In addition, the computational cost of the filter-wrapper method is largely lower than the computational cost of pure wrappers due to the utilization of pre-ordering [90]. Therefore, in the following part, we will discuss possible filters and wrappers, respectively, for the hybrid method.

3.1.1 Filters for the Hybrid Method

Filters based on ranking measures are used in hybrid methods. Ranking measures like probabilistic distance measures, class separation measures [94], [33], interclass distance measures or information theoretic measures [79] can be used to measure the relevance or redundancy of features [65]. Some of those measures capture the relationship between the features and the target concept, i.e., the relevance, while other measures capture the relationship among features themselves, i.e., the redundancy. We will adopt the following three commonly used filter ranking measures, i.e., the RELIEF weight measure [107], the class separability measure [33], [65] and the mRMR measure [40], as the filter in our experiment for comparison.

- (i) *RELIEF Weight Measure*: RELIEF proposed by Kira and Rendell [107] is a filter approach to feature selection, and has been shown to be very efficient in evaluating features' importance. The evaluation measure on features is based

CHAPTER 3. FEATURE SELECTION: A GENERAL HYBRID APPROACH TO CLASS-DEPENDENT FEATURE SELECTION

on how well those features distinguish among instances that are close to each other. Here two notations are used to describe the two nearest neighbors of the given instance: *nearest hit* (nearest neighbor of the given instance from the same class) and *nearest miss* (nearest neighbor of the given instance from different classes). The original *RELIEF* algorithm is depicted as follows:

- Introduce a weight vector and initialize it to zero: $(w_1, \dots, w_i, \dots, w_D) = 0$. D is the number of features.
- Randomly select an instance $\mathbf{X}$ from training instances S and find its *nearest hit* $\mathbf{X}_h$ and *nearest miss* $\mathbf{X}_m$.
- Calculate and update the weight w_i of the i -th feature x_i : $w_i = w_i + (x_i - x_{mi})^2 - (x_i - x_{hi})^2$, $i = 1, 2, \dots, D$. x_{mi} is the i -th element of $\mathbf{X}_m$ and x_{hi} is the i -th element of $\mathbf{X}_h$. Each feature's weight is updated by the differences (Euclidean distance) between the samples and the misses (miss differences) and between the samples and the hits (hit differences). The weight is increased if the miss differences are greater than the hit differences, which means that the feature is relevant.
- Repeat the 2nd and 3rd steps over all training instances.

In our approach to class-dependent feature selection, multi-class problems will firstly be converted into multiple two-class problems. Therefore for each two-class problem, the original *RELIEF* algorithm [107] can be used to rank features. However, for multi-class problem, the extended version *RELIEFF* [39] will be used. That is, for the class-independent feature selection method, the *RELIEFF* will be used to rank all features' importance. For multi-class problems, each sample will have one hit and more than one miss. The differences between the sample and those misses are weight-averaged [34]. The

CHAPTER 3. FEATURE SELECTION: A GENERAL HYBRID APPROACH TO CLASS-DEPENDENT FEATURE SELECTION

weight adopted in our experiment is the a priori probability of the miss class [34].

After obtaining the weight of each feature, we rank the importance of the features according to the rationale: the larger the weights, the more important the features.

- (ii) *Class Separability Measure*: The class separability measure (CSM) [33] evaluates how well two classes are separated by a feature vector. The greater the distance between different classes, the easier the classification task. Therefore, the feature subset that can maximize the distances between different classes may also maximize the classification accuracy and is therefore considered to be more important. The j -th sample is represented as $\{\mathbf{X}_j, t_j\}$, where $\mathbf{X}_j = \{x_{j1}, x_{j2}, \dots, x_{jD}\}$ is the input data, and $t_j \in \{1, 2, \dots, C\}$ is the class label of $\mathbf{X}_j$. Class separability consists of two elements, i.e., the distance between patterns within each class S_w [33], [65]

$$S_w = \sum_{c=1}^C P_c \sum_{j=1}^{n_c} \left[(\mathbf{X}_{cj} - \bar{\mathbf{m}}_c) (\mathbf{X}_{cj} - \bar{\mathbf{m}}_c)^T \right]^{1/2} \quad (3.1)$$

and the distance between patterns among different classes S_b [33], [65]

$$S_b = \sum_{c=1}^C P_c \left[(\bar{\mathbf{m}}_c - \bar{\mathbf{m}}) (\bar{\mathbf{m}}_c - \bar{\mathbf{m}})^T \right]^{1/2} \quad (3.2)$$

Here P_c is the probability of the c -th class, n_c is the number of samples in the c -th class, $\mathbf{X}_{cj}$ is the j -th sample in the c -th class, $\bar{\mathbf{m}}_c$ is the mean vector of the c -th class, and $\bar{\mathbf{m}}$ is the mean vector of all samples in the data set.

The ratio S_w/S_b can be used to measure the separability of the classes [33], [65]: the smaller the value, i.e., the smaller the distances within each class and

CHAPTER 3. FEATURE SELECTION: A GENERAL HYBRID APPROACH TO CLASS-DEPENDENT FEATURE SELECTION

the greater the distance among different classes, the better the separability. The importance of a feature may be evaluated by the ratio S'_w/S'_b , i.e., *firstly removing the feature from the data set and then calculating the ratio S_w/S_b* . The greater S'_w/S'_b is, the more important the removed feature (attribute) is. For example, if removing feature 1 from the data set leads to greater S'_w/S'_b , compared with removing feature 2, we may consider feature 1 is more important compared to feature 2 for classifying the data set, and vice versa. Hence we may evaluate the importance level of the features according to the ratio S_w/S_b [33] with a feature removed each time in turn.

- (iii) *Minimal-Redundancy-Maximal-Relevancy Measure*: The minimal-redundancy-maximal-relevancy (mRMR) feature selection method was proposed by Peng et al. [40]. It is based on the mutual information (MI) theory [108] to select features in terms of their relevancy and redundancy. This measure is equivalent to the criterion of the maximal dependency, i.e., maximal relevancy between features and target classes and minimal redundancy between features [40]. Different from two previous feature ranking measures (RELIEFF and CSM), the mRMR measure evaluates features on both relevancy and redundancy.

Let $f^D = \{x_i | i = 1, 2, \dots, D\}$ denote a feature set. According to the definition of MI in [109], the MI value of two features x_i and x_j is denoted as $I(x_i; x_j)$ ($i, j = 1, 2, \dots, D$), which describes the correlation of the two features. The greater the MI value of two features, the more correlated the two features, and the higher redundancy between the two features. In the same manner, the MI value $I(x_i; c)$ (c is one of C classes) is used to denote the relevancy of the feature x_i . Peng et al. [40] used the incremental search

CHAPTER 3. FEATURE SELECTION: A GENERAL HYBRID APPROACH TO CLASS-DEPENDENT FEATURE SELECTION

strategy to sequentially select features with the maximal relevancy and minimal redundancy (see Eq.(3.3)). The sequence in which features are included is equivalent to the ranking order of features. Suppose f^{m-1} denotes the feature subset consisting of $m - 1$ top ranked features, the m -th ranked feature x_m should be selected according to the following expression:

$$\max_{x_m \in (f^D - f^{m-1})} [I(x_m; c) - 1/m \sum_{x_j \in f^{m-1}} I(x_m; x_j)] \quad (3.3)$$

Here x_m is selected from the complementary set of f^{m-1} , i.e., $f^D - f^{m-1}$.

Although RELIEF/RELIEFF can not detect redundancy in features, RELIEF/RELIEFF is often adopted due to its efficient computation [34], [110]. The CSM is used to detect features' relevancy in classification problems by many authors, e.g., Fu et al. [33], even as a linear ranking measure. Compared with these two feature importance ranking measures, the mRMR measure may be more effective since it is a non-linear method and evaluates features in terms of both relevancy and redundancy.

3.1.2 A Feature Extraction Method–Kernel Fisher Discriminant Analysis

In the proposed framework, we only utilized feature selection methods. In fact, feature extraction methods also can be applied. To demonstrate feature selection methods is more fit for the proposed framework, we will adopt one feature extraction method to compare with feature selection methods. Among those common feature extraction methods, e.g., principle component analysis (PCA) [111], [96], fisher discriminant analysis (FDA) and its variant kernel FDA [112], [113], [114], we choose kernel FDA (KFDA) in our experiment, since KFDA is a class-specific transformation and is hence expected to be superior in classification problems.

CHAPTER 3. FEATURE SELECTION: A GENERAL HYBRID APPROACH TO CLASS-DEPENDENT FEATURE SELECTION

The KFDA was proposed by Mika et al. [112]. It first maps the original data (e.g., X) into a new feature space (e.g., F), and then conducts the FDA algorithm in the feature space to extract nonlinear discriminative features from the original data. According to [112], the KFDA algorithm is described as maximizing the following objective:

$$J(w) = \frac{w^T M_B^\phi w}{w^T M_W^\phi w} \quad (3.4)$$

Here ϕ is the transformation function which realizes the kernel trick, i.e., $\phi(x) \rightarrow F$. F is the transformed feature space and w is one of all possible solutions, a subset of F . M_B^ϕ is the between-class scatter matrix in the feature space,

$$M_B^\phi = \sum_{c=1}^C P_c \left[(\bar{m}_c^\phi - \bar{m}^\phi)^T (\bar{m}_c^\phi - \bar{m}^\phi) \right] \quad (3.5)$$

and M_W^ϕ is the within-class scatter matrix in the feature space.

$$M_W^\phi = \sum_{c=1}^C P_c \sum_{j=1}^{n_c} \left[(\mathbf{X}_{cj}^\phi - \bar{m}_c^\phi)^T (\mathbf{X}_{cj}^\phi - \bar{m}_c^\phi) \right] \quad (3.6)$$

In this case, $X_{cj}^\phi = \phi(X_{cj})$, $\bar{m}_c^\phi = 1/n_c \sum_{j=1}^{n_c} \phi(X_{cj})$ and $\bar{m}^\phi = 1/N \sum_{j=1}^N \phi(X_j)$. X_j and X_{cj} are the same as previous descriptions in the CSM measure. N is the total number of samples. Common kernel functions include Gaussian RBF, polynomials and so on. In this thesis, we adopt the Gaussian RBF kernel, which is denoted as $k(x, y) = \exp(-\frac{\|x-y\|^2}{\delta})$ and δ is a positive constant. In our experiment, we have $\delta=2^{-7}$.

3.1.3 One-against-one Strategy for Class-dependent Feature Selection

In our experiment, the “one-against-all” strategy [115] is used in previous class-dependent feature selection and feature extraction methods. The “one-against-one” strategy [116], [117], [115] can also be used to realize the class-dependent

CHAPTER 3. FEATURE SELECTION: A GENERAL HYBRID APPROACH TO CLASS-DEPENDENT
FEATURE SELECTION

feature selection. The basic theory of “one-against-one” is to construct all possible two-class classifiers, i.e., $C(C - 1)/2$ classifiers for C classes. Each classifier is trained on samples only from two classes. When testing, we count the vote of each class and adopt the Max Wins algorithm [116] to determine the final output. In our experiment, we combine the “one-against-one” strategy with the mRMR ranking measure to select class-dependent features.

3.1.4 Wrappers for the Hybrid Method

Many induction algorithms can be used as wrappers [105] to select feature subsets for the hybrid method. For example, the radial basis function (RBF) neural networks [118] [33] can be chosen since they have been extensively applied in fields like function approximation, prediction or forecasting, regression, signal processing and control theory. Other learning algorithms like decision trees (DTs) [104] [119] [120] [121], NB [51] [122] [105] [123], and fuzzy neural networks (FNNs) [25] can also be chosen.

Since the goal of wrappers to feature selection is to obtain a maximal classification accuracy on an unseen data set, and the common metric to guide the feature subset selection is the accuracy each feature subset leads to. Classifiers with strong classification capability is necessary. SVM is a very powerful pattern recognition technique [124] [125] [126] [127] [11] and has successful application fields [45] [46] [25] [97] with the advantage of fast speed and strong classification capabilities. In the NIPS 2003 feature selection challenge [128], most winners chose the SVM as their classifiers because of its good classification performance. In addition, through rough comparison, we also found out that SVM always produced better results among most of the data sets than other classifiers like RBF. Hence, we will choose the SVM in our dissertation.

3.2 The Proposed General Hybrid Approach to Class-dependent Feature Selection

Our general hybrid approach to selection of class-dependent features adopts the “one-against-all” strategy and consists of the following two steps (Fig. 3.1). In the first step, we convert a C -class classification problem to C 2-class classification problems, i.e., problem 1, problem 2, $\dots$, and problem C . The goal of problem c is to correctly separate the original class c from the other patterns, where $c = 1, 2, \dots, C$. For problem c , we divide all training patterns into two classes: class A_1 is the original class c , and class A_2 consists of all the other training patterns.

The second step is to search for a desirable feature subset for each binary classification problem. Firstly, an attribute importance ranking measure is used to rank features for each class. As previously mentioned, the three ranking measures, i.e., RELIEF [107], CSM [33], [65], and the mRMR measure [40], are used in our work for comparison. We regard the attribute importance ranking thus obtained *class-dependent attribute importance ranking*.

After obtaining the attribute importance ranking of each class, we then choose a desirable feature subset for each class through a classifier, e.g., the support vector machine (SVM), for a given feature subset search algorithm. For each class, we use forward search (or bottom-up search) to form different attribute subsets [33]. That is, we start with the most important feature as the first feature subset, and then each time add one attribute into the previous subset in the order of importance ranking to form a new feature subset, until a stopping condition is satisfied, e.g., the validation accuracy starts to decrease or all the attributes in this class have been added. For high-dimensional data sets, it will be computationally expensive to incrementally form all feature subsets and evaluate them through a classifier. Therefore, we will first select certain number of features, e.g., the top 70 from 649

CHAPTER 3. FEATURE SELECTION: A GENERAL HYBRID APPROACH TO CLASS-DEPENDENT
FEATURE SELECTION

Figure 3.1: Schematic diagram for our general hybrid approach to class-dependent feature selection.

CHAPTER 3. FEATURE SELECTION: A GENERAL HYBRID APPROACH TO CLASS-DEPENDENT
FEATURE SELECTION

ranked features for the HDR-multifeature (or HDR) data set from the LIBSVM data [129], and then incrementally form different feature subsets. Inputting each feature subset into the classifier, we can obtain different classification accuracy for different feature subsets. Then we choose the attribute combination with the highest classification accuracy or lowest error rate as the desirable feature subset for a given feature subset search algorithm.

We use a feature mask to represent a feature subset. For example, we use a “0” or “1” in a feature mask to indicate the absence or presence of a particular feature, respectively. For example, if originally there are 5 features, denoted as $\{x_1, x_2, x_3, x_4, x_5\}$, and the desirable feature subset turns out to be $\{x_1, x_2, x_3\}$ with the fourth and fifth features deleted, the feature mask will be $(1, 1, 1, 0, 0)$.

3.3 Applications

3.3.1 Data Sets

To demonstrate the class-dependent situation like the patient scenario, we generate two artificial data sets. Data set A has 750 samples with 5 classes and 10 input features $\{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}\}$. Feature x_i ($i=1,2,\dots,5$) is useful in discriminating only class i ($i=1,2,\dots, 5$) from all other classes. The feature x_6 is relevant to all the classes, whereas x_7, x_8, x_9 and x_{10} are irrelevant to all the 5 classes. For example, x_1 is randomly generated with the distribution of mean value 1 and variance 0.001 for the class 1, and for other classes distributed with random values within the range of $[0.01,30]$ with a great variance. x_2 is randomly generated with the distribution of mean value 3 and variance 0.001 for the class 2, and for all the other classes except the class 2 distributed with random values within the range of $[0.01, 30]$. Similarly, x_i with $i=3, 4, 5$ is randomly generated. x_6 is randomly generated with 5 different mean values for 5 classes and each mean

CHAPTER 3. FEATURE SELECTION: A GENERAL HYBRID APPROACH TO CLASS-DEPENDENT FEATURE SELECTION

is distributed with a variance of 0.0001. x_i with $i=7, 8, 9, 10$ is randomly generated within $[0.01, 30]$ with a great variance. Data set B, has 1500 samples with 10 classes and 20 features. Feature x_i ($i=1, 2, \dots, 10$) is useful in discriminating only class i ($i=1, 2, \dots, 10$) with all other classes except the class i . The feature 11 is relevant to all the classes, whereas all the left features from 12 to 20 are irrelevant to all the 10 classes. All features are randomly generated in the same manner as the artificial data A.

We apply our class-dependent feature selection method with the CSM and RELIEFF ranking measures, respectively, both using the “one-against-all” strategy on the two artificial data sets and several low-dimensional data sets from the UCI Machine Learning Repository [8], i.e., the Breast cancer, Ionosphere, Thyroid, Iris, Ecoli, Waveform, Segment and Vehicle data sets. For the first 6 UCI machine learning repository data sets and the Waveform data set with 21 features, we apply class-dependent feature selection with the CSM and RELIEFF ranking measures. For the two artificial data sets, the Waveform data set with 40 features, Segment and Vehicle data sets, we apply class-dependent feature selection methods with the CSM and mRMR ranking measures, respectively, class-dependent feature extraction methods, all using the “one-against-all” strategy, and the class-dependent feature selection with the mRMR ranking measure using the “one-against-one” strategy on three high-dimensional data sets, i.e., the HDR data set from the UCI Machine Learning Repository [8], the DNA and the Scene data from the LIBSVM data [129]. The data sets and their characteristics are summarized in Table 3.1. Since the Waveform data set has two groups of data, we use Waveform40 to denote the group with 40 features, and Waveform21 to denote the group with 21 features.

CHAPTER 3. FEATURE SELECTION: A GENERAL HYBRID APPROACH TO CLASS-DEPENDENT
FEATURE SELECTION

Table 3.1: Basic information of the data sets used in this thesis.

Data set	Number of samples	Total number of classes	Total number of attributes
Artificial data A	750	5	10
Artificial data B	1500	10	20
Breast cancer	699	2	9
Ionosphere	351	2	34
Thyroid	215	3	5
Iris	150	3	4
Ecoli	336	8	7
Wine	178	3	13
Waveform21	1000	3	21
Waveform40	1000	3	40
Segment	2310	7	19
Vehicle	846	4	18
HDR	2000	10	649
DNA	3186	3	180
Scene	2407	6	294

3.3.2 Results and Discussions

For the two artificial data sets, we use the class-independent and class-dependent methods on them for comparison, both together with the class separability ranking measure and the “one-against-all” measure. For the classification of the two data sets, we train and test using the 10-fold cross validation. That is, we separate the original data into ten subsets of equal size, each time with nine subsets for training and the left one for testing. This process repeats 10 times. Table 3.2 and Table 3.4 provide the times that each feature is selected in 10 simulations (folds). For example, the first row of data in Table 3.2 shows that in the 10 simulations, the class-independent feature selection method selected the 1st feature 10 times, the 2nd feature 7 times, the 3rd feature 10 times, the 4th feature 7 times, The last 3 features (underlined) with zeroes were almost never selected. Similarly, the first row of data in Table 3.4 shows that in the 10 simulations, the class-independent

CHAPTER 3. FEATURE SELECTION: A GENERAL HYBRID APPROACH TO CLASS-DEPENDENT
FEATURE SELECTION

feature selection method selected the 1st feature 1 times, the 2nd feature 8 times, the 3rd feature 4 times, the 4th feature 1 times, ..., the 11th feature 10 times and the last 9 features (underlined) were never selected. Therefore, if the value of times is zero, it means the feature is completely eliminated in the 10 simulations; If the value of times is 10, it means the feature is always selected or very relevant in 10 simulations. For both feature selection methods, we can see that, for artificial data A (Table 3.2), features 7~10 are always eliminated in 10 simulations because they are irrelevant (noise), and features 12~20 are always eliminated in 10 simulations because they are noise for data B (Table 3.4). These results prove that the two feature selection methods are very effective in deleting noises (irrelevant features). On the other hand, there are some differences between the results obtained from these two methods. For the *class-independent* feature selection method, features 1, 3, 5 and 6 are selected in 10 simulations, features 2 and 4 are selected in 7 simulations for the artificial data A. While for the *class-dependent* feature selection method, each class has its own specific feature subset. For example (see Table 3.2), classes 1, 2 and 5 have stable feature subsets {1, 6}, {2, 6} and {5, 6}, respectively, in 10 simulations. The feature subsets for classes 3 and 4 have variations in 10 simulations. The class-independent and class-dependent feature selection methods nearly produce the same difference on artificial data B as on artificial data A. When comparing the average number of features selected by the class-dependent feature selection method with that by the class-independent feature selection method, the *class-dependent* feature selection method selects much fewer features than those selected by the *class-independent* feature selection method. The results are listed in Table 3.3 for the artificial data A and Table 3.5 for the artificial B. For example, in Table 3.3, the first row of data shows that the class-independent feature selection method selected 6 features in the 1st simulation, 4 features in the 2nd simulation,

CHAPTER 3. FEATURE SELECTION: A GENERAL HYBRID APPROACH TO CLASS-DEPENDENT
FEATURE SELECTION

Table 3.2: The total number of times that each of the ten features is selected in 10 simulations for artificial data set A, using the CSM ranking and one-against-all measures.

Feature selection approach	Classes	The total times that each feature is selected (relevant) in 10 simulations
Class-independent	all classes	10 7 10 7 10 10 <u>1 0 0 0</u>
Class-dependent	class 1	10 0 0 0 0 10 <u>0 0 0 0</u>
	class 2	0 10 0 0 0 10 <u>0 0 0 0</u>
	class 3	1 1 10 2 0 2 <u>0 0 0 0</u>
	class 4	0 0 0 10 1 1 <u>0 0 0 0</u>
	class 5	0 0 0 0 10 10 <u>0 0 0 0</u>

Table 3.3: The total number of features selected in each of the 10 simulations, the average number and its standard deviation (std), using class separability measure (CSM) to rank features for the artificial data A.

Feature selection approach	CSM ranking method		
	Classes	Total number of features selected in each of the 10 simulations	Average $\pm$ std
Class-independent	all classes	6 4 4 4 6 6 6 6 6 7	5.5 $\pm$ 1.0
Class-dependent	class 1	2 2 2 2 2 2 2 2 2 2	2 $\pm$ 0
	class 2	2 2 2 2 2 2 2 2 2 2	2 $\pm$ 0
	class 3	1 1 1 1 1 1 1 4 4 1	1.6 $\pm$ 1.3
	class 4	1 1 3 1 1 1 1 1 1 1	1.2 $\pm$ 0.6
	class 5	2 2 2 2 2 2 2 2 2 2	2 $\pm$ 0
	average	1.6 1.6 2 1.6 1.6 1.6 1.6 2.2 2.2 1.6	1.8 $\pm$ 0.6

4 features in the 3rd simulation, ..., with an average of 5.5 features selected in the 10 simulations. The artificial data A has on average 5.5 features selected by the class-independent feature selection method, while for the class-dependent feature selection method, it has on average 1.8 features selected (see the last row of the Table 3.3). Table 3.5 follows the same rule as Table 3.3. For example, artificial data B has on average 1.6 features selected by the class-dependent feature selection method and 5.4 features selected by the class-independent method. After considering the standard deviation, the conclusion is still the same.

For the first 7 benchmark data sets from the UCI machine learning database,

CHAPTER 3. FEATURE SELECTION: A GENERAL HYBRID APPROACH TO CLASS-DEPENDENT FEATURE SELECTION

Table 3.4: Same as Table 3.2, for the artificial data set B, which has 10 classes and 20 features.

Feature selection approach	Classes	The total times each feature is selected (relevant) in 10 simulations
Class-independent	all classes	1 8 4 1 4 7 4 5 6 4 10 0 0 0 0 0 0 0 0 0
Class-dependent	class 1	10 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0
	class 2	0 10 0 0 0 0 0 0 0 0 0 10 0 0 0 0 0 0 0 0
	class 3	0 0 10 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
	class 4	0 0 0 10 0 0 0 0 0 0 0 10 0 0 0 0 0 0 0 0
	class 5	0 0 0 0 10 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0
	class 6	0 0 0 0 0 10 0 0 0 0 0 1 0 0 0 0 0 0 0 0
	class 7	0 0 0 0 0 0 10 0 0 0 0 10 0 0 0 0 0 0 0 0
	class 8	0 0 0 0 0 0 0 10 0 0 9 0 0 0 0 0 0 0 0 0
	class 9	0 0 0 0 0 0 0 0 10 0 9 0 0 0 0 0 0 0 0 0
	class 10	0 0 0 0 0 0 0 0 0 10 10 0 0 0 0 0 0 0 0

Table 3.5: Same as Table 3.3, for the artificial data B.

Feature selection approach	CSM Ranking method		
	Classes	Total number of features selected in each of the 10 simulations	Average $\pm$ std
Class-independent	all classes	7 8 7 9 3 6 4 4 2 4	5.4 $\pm$ 2.2
Class-dependent	class 1	1 1 2 1 2 2 1 1 1 1	1.3 $\pm$ 0.5
	class 2	2 2 2 2 2 2 2 2 2 2	2 $\pm$ 0
	class 3	2 1 1 1 1 1 1 1 1 1	1.1 $\pm$ 0.3
	class 4	2 2 2 2 2 2 2 2 2 2	2 $\pm$ 0
	class 5	1 1 1 1 1 1 3 1 1 1	1.2 $\pm$ 0.6
	class 6	1 1 1 1 1 1 1 1 1 1	1 $\pm$ 0
	class 7	2 2 2 2 2 2 2 2 2 2	2 $\pm$ 0
	class 8	2 2 2 2 2 2 2 2 2 1	1.9 $\pm$ 0.3
	class 9	1 2 2 2 2 2 2 2 2 2	1.9 $\pm$ 0.3
	class 10	2 2 2 2 2 2 2 2 2 2	2 $\pm$ 0
	average	1.6 1.6 1.7 1.6 1.7 1.7 1.8 1.6 1.6 1.5	1.6 $\pm$ 0.3

CHAPTER 3. FEATURE SELECTION: A GENERAL HYBRID APPROACH TO CLASS-DEPENDENT FEATURE SELECTION

we also train and test them using the 10-fold cross validation. From Tables 3.6-3.12, we list the total number of features selected in each of the 10 simulations, the average number of selected features and its standard deviation for the two feature selection methods. What's more, we calculate the overall average number of features selected and the standard deviation for the class-dependent method, which are listed in the last row of each table for rough comparison. According to the results listed, we can see that some data sets have few or even no irrelevant or redundant features, i.e. the Breast Cancer (Table 3.6), Thyroid (Table 3.8), Iris (Table 3.9), and Ecoli data sets (Table 3.10). The data set Waveform21 (Table 3.12) has many irrelevant or redundant features deleted by our method. Since the Ionosphere and Breast cancer data sets have only 2 classes, the results by the class-dependent feature selection method are the same as the results by the class-independent feature selection method (see Table 3.6-3.7). While for other multi-class data, the results for the two feature selection approaches are rather different. For most of the data sets, our class-dependent feature selection obtains the feature subset with a smaller size in comparison with the class-independent feature selection. The Wine data set has on average 11 features selected by the class-independent method, while 8, 8, and 6 features are selected for 3 different classes for the class-dependent method. Some data sets have stable feature subsets for certain classes, that is, the same feature subsets are obtained using the two ranking measures in all the 10 simulations. For instance, the Iris data set has 1 feature (with zero standard deviation) selected for class 1 in each of 10 simulations for both RELIEF and CSM ranking measures (Table 3.9). The same thing happens for class 8 of the Ecoli data set (Table 3.10).

For the other three benchmark data sets, i.e., the Waveform40, Segment and Vehicle data set, from the UCI machine learning database, we respectively utilize the

CHAPTER 3. FEATURE SELECTION: A GENERAL HYBRID APPROACH TO CLASS-DEPENDENT
FEATURE SELECTION

Table 3.6: Same as Table 3.3, with RELIEF and CSM to rank features, for the Breast cancer data set with 9 feautres.

Feature selec- tion approach	RELIEF		CSM	
	Total number of fea- tures selected in each of the 10 simulations	Average± std	Total number of fea- tures selected in each of the 10 simulations	Average± std
Class- independent	6 6 5 6 6 9 9 5 8 8	6.8±1.5	9 9 9 9 9 8 8 8 9 9	8.7±0.5
Class- dependent	6 6 5 6 6 9 9 5 8 8	6.8±1.5	9 9 9 9 9 8 8 8 9 9	8.7±0.5

Table 3.7: Same as Table 3.3, with RELIEF and CSM to rank features, for the Ionosphere data set with 34 features.

Feature selec- tion approach	RELIEF		CSM	
	Total number of fea- tures selected in each of the 10 simulations	Average± std	Total number of fea- tures selected in each of the 10 simulations	Average± std
Class- independent	19 18 16 32 28 20 26 14 29 27	22.9±6.2	22 22 22 20 21 20 22 20 20 25	21.4±1.6
Class- dependent	19 18 16 32 28 20 26 14 29 27	22.9±6.2	22 22 22 20 21 20 22 20 20 25	21.4±1.6

Table 3.8: Same as Table 3.3, with RELIEF and CSM to rank features, for the Thyroid data set with 5 features.

Feature selec- tion approach	Classes	RELIEF		CSM	
		Total number of fea- tures selected in each of the 10 simulations	Average ± std	Total number of fea- tures selected in each of the 10 simulations	Average ± std
Class- independent	all classes	4 0 4 4 4 4 5 4 4 5	4.3±0.5	4 3 4 4 4 3 5 4 4 5	4±0.7
Class- dependent	class 1	4 4 4 3 4 4 4 5 4 4	4.0±0.5	5 5 5 5 5 5 5 5 5 5	5
	class 2	4 4 3 4 4 2 5 3 4 4	3.7±0.8	4 3 4 4 3 2 5 3 3 3	3.4±0.8
	class 3	3 3 4 3 3 5 5 3 3 4	3.6±0.8	4 4 5 4 4 4 5 4 4 4	4.2±0.4
	average	3.7 3.7 3.7 3.3 3.7 3.7 4.7 3.7 3.7 4	3.8±0.4	4.3 4 4.7 4.3 4 3.7 5 4 4 4	4.2±0.4

CHAPTER 3. FEATURE SELECTION: A GENERAL HYBRID APPROACH TO CLASS-DEPENDENT
FEATURE SELECTION

Table 3.9: Same as Table 3.3, with RELIEF and CSM to rank features, for the Iris data set with 4 features.

Feature selection approach	Classes	RELIEF		CSM	
		Total number of features selected in each of the 10 simulations	Average $\pm$ std	Total number of features selected in each of the 10 simulations	Average $\pm$ std
Class-independent	all classes	2 4 1 1 4 4 4 3 3 2	2.8 $\pm$ 1.2	4 3 4 4 4 4 4 4 4 4	3.9 $\pm$ 0.3
Class-dependent	class 1	1 1 1 1 1 1 1 1 1 1	1.0 $\pm$ 0	1 1 1 1 1 1 1 1 1 1	1 $\pm$ 0
	class 2	3 2 4 4 4 3 3 4 3 3	3.3 $\pm$ 0.7	4 4 4 4 3 4 3 4 4 4	3.8 $\pm$ 0.4
	class 3	1 1 1 2 3 1 2 4 3 3	2.1 $\pm$ 1.0	1 2 1 3 3 1 2 4 3 3	2.3 $\pm$ 1.1
	average	1.7 1.3 2 2.3 2.7 1.7 2 3 2.3 2.3	2.1 $\pm$ 0.5	2 2.3 2 2.7 2.3 2 2 3 2.7 2.7	2.4 $\pm$ 0.4

Table 3.10: Same as Table 3.3, with RELIEF and CSM to rank features, for the Ecoli data set with 7 features.

Feature selection approach	Classes	RELIEF		CSM	
		Total number of features selected in each of the 10 simulations	Average $\pm$ std	Total number of features selected in each of the 10 simulations	Average $\pm$ std
Class-independent	all classes	7 7 7 7 6 7 7 6 7 7	6.8 $\pm$ 0.4	7 7 7 7 7 6 7 7 7 7	6.9 $\pm$ 0.3
Class-dependent	class 1	5 5 5 5 4 5 5 5 4 5	4.8 $\pm$ 0.4	7 7 7 7 6 6 7 7 6 7	6.7 $\pm$ 0.5
	class 2	4 2 2 4 2 4 6 5 2 3	3.4 $\pm$ 1.4	7 7 7 7 7 7 7 7 7 7	7
	class 3	7 6 6 6 7 5 7 7 6 6	6.3 $\pm$ 0.7	7 7 6 6 6 6 7 6 6 6	6.3 $\pm$ 0.5
	class 4	6 4 5 3 2 5 5 5 5 6	4.6 $\pm$ 1.3	5 3 6 3 5 3 3 2 3 3	3.6 $\pm$ 1.3
	class 5	4 3 4 5 6 3 7 3 3 6	4.4 $\pm$ 1.5	5 6 4 7 7 4 6 5 5 4	5.3 $\pm$ 1.2
	class 6	7 7 7 3 5 5 2 6 7 7	5.6 $\pm$ 1.8	3 3 3 3 3 5 3 3 3 3	3.2 $\pm$ 0.6
	class 7	1 1 1 1 1 2 1 1 1 1	1.1 $\pm$ 0.3	1 1 1 1 1 3 1 1 1 1	1.2 $\pm$ 0.6
	class 8	1 1 1 1 1 1 1 1 1 1	1.0 $\pm$ 0.0	1 1 1 1 1 1 1 1 1 1	1.0 $\pm$ 0
	average	4.4 3.6 3.9 3.5 3.5 3.7 4.2 4.1 3.6 4.4	3.9 $\pm$ 1.1	4.5 4.4 4.4 4.4 4.5 4.4 4.4 4.4 4	4.3 $\pm$ 0.7

CHAPTER 3. FEATURE SELECTION: A GENERAL HYBRID APPROACH TO CLASS-DEPENDENT
FEATURE SELECTION

Table 3.11: Same as Table 3.3, with RELIEF and CSM to rank features, for the Wine data set with 13 features.

Feature selection approach	Classes	RELIEF		CSM	
		Total number of features selected in each of the 10 simulations	Average $\pm$ std	Total number of features selected in each of the 10 simulations	Average $\pm$ std
Class-independent	all classes	10 4 6 4 6 5 11 9 6 9	7.0 $\pm$ 2.5	11 13 11 12 13 11 12 9 11 12	11.5 $\pm$ 1.2
Class-dependent	class 1	11 9 9 8 8 10 11 6 10 11	9.3 $\pm$ 1.6	7 7 7 7 9 7 11 8 11 11	8.5 $\pm$ 1.8
	class 2	11 12 12 10 6 10 11 11 12 4	9.9 $\pm$ 2.7	7 6 6 6 13 7 8 13 6 6	7.8 $\pm$ 2.8
	class 3	5 7 6 6 5 4 7 7 5 10	6.2 $\pm$ 1.7	4 5 3 8 4 7 8 4 7 7	5.7 $\pm$ 1.9
	average	9 9.3 9 8 6.3 8 9.7 8 9 8.3	8.5 $\pm$ 2	6 6 5 7 8.7 7 9 8.3 8 8	7.3 $\pm$ 2

Table 3.12: Same as Table 3.3, with RELIEF and CSM to rank features, for the Waveform21 data set with 21 features.

Feature selection approach	Classes	RELIEF		CSM	
		Total number of features selected in each of the 10 simulations	Average $\pm$ std	Total number of features selected in each of the 10 simulations	Average $\pm$ std
Class-independent	all classes	17 19 19 20 17 19 19 17 17 19	18.3 $\pm$ 1.2	19 19 19 19 16 18 19 19 16 17	18.1 $\pm$ 1.3
Class-dependent	class 1	18 19 17 15 19 18 20 18 19 19	18.2 $\pm$ 1.4	18 15 15 15 15 15 19 17 13 15	15.7 $\pm$ 1.8
	class 2	14 18 15 19 14 12 18 18 15 15	15.8 $\pm$ 2.3	15 14 15 16 16 15 17 15 17 16	15.6 $\pm$ 1.0
	class 3	14 18 16 16 17 17 15 16 16 13	15.8 $\pm$ 1.5	17 14 17 17 17 15 16 14 17 15	15.9 $\pm$ 1.3
	average	15.3 18.3 16 16.7 16.7 15.7 17.7 17.3 16.7 15.7	16.6 $\pm$ 1.8	16.7 14.3 15.7 16 16 15 17.3 15.3 15.7 15.3	15.7 $\pm$ 1.4

CHAPTER 3. FEATURE SELECTION: A GENERAL HYBRID APPROACH TO CLASS-DEPENDENT FEATURE SELECTION

RELIEF and CSM to rank features for the class-independent and class-dependent methods using the “one-against-all” strategy. From the results in Table 3.13, we can see that the Vehicle data set has few irrelevant or redundant features, whereas the Waveform40 and Segment data sets have many irrelevant or redundant features deleted by the class-independent and class-dependent methods. More importantly, our class-dependent feature selection obtains a smaller feature subset in comparison with the class-independent feature selection, especially for the CSM. For example, the Waveform40 data set has on average 24 features selected for the class-independent method with the CSM, while for the class-dependent method, the average numbers of features selected for the 3 classes are 19. The Segment data set has on average 15 features selected by the class-independent method with the RELIEF measure, 19 features selected by the CSM, while on average 10 features selected by the class-dependent method with the RELIEFF measure and 11 features by the CSM (see the last row of Table 3.13). Also rather different numbers of features are selected for the 7 classes by the class-dependent method.

For the three high-dimensional data sets in Table 3.1, i.e., HDR, Scene and DNA, we use a different training and testing method, i.e., randomly separating the original data into the training and testing parts. For example, for the HDR data set, we randomly select 1000 samples for training and 1000 for testing. For the DNA data set, we randomly select 2000 samples for training and 1186 for testing. For the Scene data set, we randomly select 1211 for training and 1196 for testing. The whole training and testing processes are repeated 10 times.

For those three high-dimensional data sets, we utilize the mRMR and CSM ranking measures, respectively, to rank features using the “one-against-all” strategy. The relevant results are presented in Table 3.14. For the HDR data set, the class-independent feature selection method with the mRMR measure selects

CHAPTER 3. FEATURE SELECTION: A GENERAL HYBRID APPROACH TO CLASS-DEPENDENT
FEATURE SELECTION

Table 3.13: The average number of features ($\pm$ std) selected in 10 simulations , by class-independent and class-dependent methods with “one-against-all” strategy, for the Waveform40, Segment and Vehicle data sets.

Feature se- lection approach	Classes	CSM ranking			RELIEFF ranking		
		Waveform 40	Segment	Vehicle	Waveform 40	Segment	Vehicle
Class- independent	all classes	23.8 $\pm$ 3	18.5 $\pm$ 0.8	17.1 $\pm$ 1	23.8 $\pm$ 7.4	15.3 $\pm$ 1.1	17.6 $\pm$ 0.7
Class- dependent	class 1	19.3 $\pm$ 3	8.6 $\pm$ 0.2	17.5 $\pm$ 0.8	23.9 $\pm$ 4.8	8.8 $\pm$ 0.4	16.2 $\pm$ 1.9
	class 2	16.7 $\pm$ 1	2.6 $\pm$ 0.8	17.3 $\pm$ 0.8	22.9 $\pm$ 4.8	2.5 $\pm$ 0.9	16.0 $\pm$ 2.1
	class 3	20 $\pm$ 4	15.8 $\pm$ 0.6	16.8 $\pm$ 1	23.1 $\pm$ 5.9	14.3 $\pm$ 2.3	17.1 $\pm$ 1
	class 4		8.1 $\pm$ 0.3	18 $\pm$ 0		11.3 $\pm$ 3.2	17.2 $\pm$ 1
	class 5		17.1 $\pm$ 0.3			12.3 $\pm$ 1.6	
	class 6		13.2 $\pm$ 3.7			10.7 $\pm$ 1.3	
	class 7		3.4 $\pm$ 1.3			6.9 $\pm$ 3.8	
	average	18.7 $\pm$ 2.9	11.2 $\pm$ 2.3	17.4 $\pm$ 0.8	23.3 $\pm$ 5.2	10 $\pm$ 2.5	16.6 $\pm$ 1.6

on average 48 features from 649 features, and the class-dependent method with the mRMR measure selects only 21 features averaged from all class-dependent features. In comparison, the CSM ranking measure has on average 65 features selected by the class-independent and 33 features selected by the class-dependent method. For the Scene and DNA data sets, the class-dependent feature selection with both ranking measures, i.e., the mRMR and CSM, selects fewer features than those by the class-independent feature selection method. When comparing the two ranking measures, we can see that the mRMR ranking measure shows advantages over the CSM in terms of the average number of features selected. In addition, we utilized the feature extraction method KFDA on these three data sets for both class-dependent and class-independent situations. Since original features are transformed after feature extraction, it is not meaningful to compare the number of features selected by feature selection and feature extraction methods. While, the comparison of classification accuracies is given in the next chapter. Also

CHAPTER 3. FEATURE SELECTION: A GENERAL HYBRID APPROACH TO CLASS-DEPENDENT FEATURE SELECTION

Table 3.14: Same as Table 3.13, with the CSM and mRMR ranking measures, for the HDR, DNA and Scene data sets. (FS denotes feature selection)

FS method	Classes	CSM ranking			mRMR ranking		
		HDR	Scene	DNA	HDR	Scene	DNA
Class-in depen- dent	all classes	64.6±5.4	94.4±9.8	46.8±9.9	47.8±4.8	84.5±16.3	40.5±8.9
Class- dependent	class 1	15.5±9.6	38.2±5.1	19±11.5	10.8±3.1	24.6±11.2	18±9.9
	class 2	38.7±7.3	40.4±1.1	37.8±7.5	26.7±13.4	27±7.1	32.3±4.7
	class 3	42.8±8.5	43±3.1	35.4±11.9	10.4±0.5	25.2±10.4	25.5±7.6
	class 4	36.3±11	38±5.1		25.4±11.3	26.6±9.9	
	class 5	36.9±12.6	31.8±9.4		10±2.4	15.6±5.3	
	class 6	35±10.6	44.4±4.9		28±14.3	30.1±7.3	
	class 7	29.1±12.9			19±7.8		
	class 8	26.1±13.1			27.2±10.9		
	class 9	40.5±3.7			15.4±6.4		
	class10	26.7±7.4			13.2±9.9		
	average	32.76±3.1	41.93±2.3	33.4±6.2	20.9±3.9	35.8±6.4	28.3±4.9

the comparison between the "one-against-one" measure and the "one-against-all" measure is focused on the classification accuracy in the next chapter. Also there are some other benchmark data sets like Golub’s microarray data which has even higher dimension than the HDR, DNA and Scene data sets, we did not adopt the Golub’s microarray data in our thesis. This is because that our proposed approach costs more time in computation than conventional feature selection approaches, it says nothing of processing data with huge dimensions like Golub’s microarray data. More importantly, the experiment results from those UCI bench mark data sets, HDR and DNA data sets had already shown the advantage of our approach.

3.4 Summary

In this Chapter, we proposed the framework of a hybrid approach to class-dependent feature selection, where three kinds of filter approaches to feature selection, i.e.,

CHAPTER 3. FEATURE SELECTION: A GENERAL HYBRID APPROACH TO CLASS-DEPENDENT FEATURE SELECTION

CSM, RELIEFF and mRMR, are applied on several benchmark data sets for a comparison. From the ranking results, the SVM is selected as the wrapper to determine the feature subset for the class-dependent and class-independent feature selection. Three filter ranking measures' performance on those data are tested in the next Chapter, which shows that the mRMR ranking measure has strong capabilities in ranking relevant and non-redundant features. In addition, a feature extraction method and "one-against-one" measure are also adopted to compare with the proposed feature selection method (i.e., with the "one-against-all" measure). It shows that the proposed feature selection method with the "one-against-all" measure has better results than the feature extraction method and also the method with "one-against-one" measure.

Feature selection is challenging because of the trade-off between the time complexity and optimality. The feature selection method proposed in our approach includes two major elements, i.e., class-dependency and wrappers, both of which need more time to seek desirable feature subsets, compared with class-independency and filters. We have shown that wrappers usually ensure higher classification accuracy than filters and class-dependent methods can have more potential to reduce the dimensionality of features than class-independent methods [78]. In order to find the approximately optimal feature subset for each class, various feature ranking measures, wrappers together with various search strategies, e.g., exhaustive search, forward search and backward search, can be adopted in the framework of our approach. In this thesis, the RELIEFF, CSM and mRMR ranking measures are adopted and partially compared. The more powerful the ranking measure, the better the selected feature subset and the better the classification performance. For search methods, although the exhaustive search can help to find an optimal feature subset, it also involves much more time than other search methods, especially for high-dimensional data sets. In this thesis, we use the sequential forward

CHAPTER 3. FEATURE SELECTION: A GENERAL HYBRID APPROACH TO CLASS-DEPENDENT FEATURE SELECTION

search to seek for feature subsets after ranking features. At same time, the SVM is chosen as the wrapper due to its good characteristics and originally being a binary classifier. There are other classifiers, e.g., the multi-layer perceptron (MLP) and radial basis function (RBF), which can also be used in our framework.

Since our feature selection method requires determination of feature subsets of every class, it is likely to be more computationally expensive compared to other class-independent feature selection methods. However, the extra computational cost may be worthwhile in certain applications where improvements of accuracy or reduction of data dimensionality are very important and meaningful.

Chapter 4

Classification of Class-dependent Feature Subsets and One of Its Applications

The support vector machine (SVM), as a learning algorithm, has been successfully used in the field of the computational biology [17], such as microarray gene expression analysis [130], [26], cancer classification [25], DNA splice site detection [131] and protein secondary structure prediction [132]. It proves to have significantly better performance than or at least match that of traditional machine learning approaches, including neural networks [132]. In this Chapter, we review the background of the SVM and state an application of the SVM on the classification of biomedical data sets with class-dependent feature subsets.

4.1 The Support Vector Machines

As a classifier, the basic idea of the SVM [133] refers to a space transformation with a non-linear mapping, i.e., transforming the linearly inseparable data in the original feature space into a new linearly separable space. The dimensionality of the new feature space is much higher than the original one so that the optimal separating hyperplane, i.e., the best decision function, can be found in the new feature space

CHAPTER 4. CLASSIFICATION OF CLASS-DEPENDENT FEATURE SUBSETS AND ONE OF ITS APPLICATIONS

[134]. To make this more concrete, let us begin with a linearly separable case with two classes. Assume that the whole training data set is $S = \{(\vec{x}_1, y_1), \dots, (\vec{x}_n, y_n)\}$, in which $y_i \in \{-1, +1\}$ ($i = 1, 2, \dots, n$) is the class label of the input sample $\vec{x}_i$. The SVM tries to find an optimal hyperplane (Eq. (4.1))

$$\vec{w}^T \cdot \vec{x}_i + b = 0 \quad (4.1)$$

to maximize the margin of the separation between the two classes of data. In Eq. (4.1), $\vec{w}$ is a weight vector connecting the input to the hyperplane and b is called a bias. The optimal hyperplane is obtained by solving the following constrained optimization problem:

$$\text{minimizing :} \quad \frac{1}{2} \|\vec{w}^T\|^2 \quad (4.2)$$

$$\text{subject to :} \quad y_i(\vec{w}^T \cdot \vec{x}_i + b) \geq 1 \quad (i = 1, 2, \dots, n) \quad (4.3)$$

If the data is linearly inseparable, the constrained optimization problem will be different. For the linearly inseparable case, a set of nonnegative slack variables $\xi_1, \xi_2, \dots, \xi_n$ are introduced to penalize the training error and a regularization parameter ζ is introduced to control the trade-off between the training error and the margin. The corresponding constrained optimization problem is changed into:

$$\text{minimizing :} \quad \frac{1}{2} \|\vec{w}^T\|^2 + \zeta \sum_{i=1}^n \xi_i \quad (4.4)$$

$$\text{subject to :} \quad y_i(\vec{w}^T \cdot \vec{x}_i + b) \geq 1 - \xi_i \quad (\xi_i \geq 0, i = 1, 2, \dots, n) \quad (4.5)$$

When using the Lagrangian function to solve the constrained minimization problem, the problem becomes:

$$\text{maximizing :} \quad \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \vec{x}_i^T \vec{x}_j \quad (4.6)$$

CHAPTER 4. CLASSIFICATION OF CLASS-DEPENDENT FEATURE SUBSETS AND ONE OF ITS APPLICATIONS

$$\text{subject to:} \quad \alpha_i y_i = 0 \quad (0 \leq \alpha_i \leq \zeta) \quad (4.7)$$

α_i and α_j are the lagrangian multipliers within the range of $[0, \zeta]$ (ζ is the regularization parameter). For this linearly inseparable problem, a space transformation is needed. Suppose that ϕ is a vector of transformation functions, which can realize the transformation from a lower dimensional input space into a higher dimensional space. The transformation of the input vector $\vec{x}_i$ (with the dimension D) from the input space to the transformed space is defined as:

$$\phi(\vec{x}) = [\phi_1(\vec{x}), \phi_2(\vec{x}), \dots, \phi_D(\vec{x})]^T \quad (4.8)$$

The decision function in the transformed space

$$\sum_{j=1}^D w_j \phi_j(\vec{x}) + b = 0 \quad (4.9)$$

can be further simplified as:

$$\sum_{j=0}^D w_j \phi_j(\vec{x}) = 0 \Rightarrow \vec{w}^T \cdot \phi(\vec{x}) = 0 \quad (4.10)$$

where $w_0 = b$ is a given bias and $\phi_0(\vec{x}) = 1$. $\vec{w}^T$ is the transposed matrix of $\vec{w}$. w_j ($j=1, 2, \dots, D$) is a given weight connecting the feature space to the output space. The connecting weight can be obtained through a quadratic programming (QP) solver ([135], [127]) (Note that $\vec{w} = (w_1, w_2, \dots, w_D)$ does not include the bias):

$$\vec{w} = \sum_{i=1}^n \alpha_i y_i \phi(\vec{x}_i) \quad (4.11)$$

Applying Eq. (4.11) to Eq. (4.10), we obtain

$$\sum_{i=1}^n \alpha_i y_i \phi^T(\vec{x}_i) \cdot \phi(\vec{x}) = 0 \quad (4.12)$$

CHAPTER 4. CLASSIFICATION OF CLASS-DEPENDENT FEATURE SUBSETS AND ONE OF ITS APPLICATIONS

Instead of calculating the transformation function ϕ , an inner product kernel function is introduced to realize the mapping:

$$K(\vec{x}, \vec{x}_i) = K(\vec{x}_i, \vec{x}) = \phi(\vec{x}_i) \cdot \phi(\vec{x}) \quad (4.13)$$

The optimal hyperplane is found by solving

$$\sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j K(\vec{x}_i, \vec{x}_j) \quad (4.14)$$

$$\text{subject to:} \quad \alpha_i y_i = 0 \quad (0 \leq \alpha_i \leq \zeta) \quad (4.15)$$

The inner product kernel function represents the transformation function ϕ and makes the inseparable data separable in new feature space. In summary, the introduction of kernel function can help avoid the curse of dimensionality during optimization in high-dimensional spaces. Hence, choosing an appropriate kernel function is very important to the SVM's performance. There are three popular types of kernels [134], [136] used in most SVM packages, i.e., the polynomial learning machine, the radial-basis function (RBF) network and the two-layer perceptron. Because the RBF kernel has many advantages compared with other kernels, e.g., with less parameters, we will choose the RBF kernel in our application.

4.2 The Classifier with Class-dependent Feature Subsets

The SVM was originally designed for binary classification [125], although most classification problems are multi-class problems. In order to solve those multi-class problems, some methods have been proposed to effectively extend the binary SVM to multi-class classification. The earliest used method is “one-against-all” [137], which is to construct C SVMs for a C -class problem. The i -th SVM is

CHAPTER 4. CLASSIFICATION OF CLASS-DEPENDENT FEATURE SUBSETS AND ONE OF ITS APPLICATIONS

trained with all the examples in the i -th class with positive labels, and all the other examples with negative labels. Another method is to construct $C(C - 1)/2$ classifiers and each classifier is trained on data only from two classes, which is called the “one-against-one” method [116] [138] [139]. Directed Acyclic Graph SVM (DAGSVM) [117] is also a method based on solving binary classifications, which has the same training phase as “one-against-one” method and uses a rooted binary directed acyclic graph in the testing phase. Also several “all-together” methods [140],[141],[142] are proposed, which are to consider all classes at once. With regard to those methods, Hsu and Lin [115] made a comparison and pointed out that “all-together” methods have advantages on support vectors, whereas it is computationally more expensive. As to the “one-against-one” strategy and DAGSVM methods, they are suitable for practical use. In addition, Hsu and Lin [115] also concluded that the “one-against-all” [137] is a good method whose performance is comparable to “one-against-one” and DAGSVM [117]. Here two approaches can be considered. One is to recode the SVM classifier with “all-together” method for our purpose, which may be time-consuming and inconvenient. The other one is to adopt the LIBSVM [129] in our experiment, which is an efficient software for SVM classification. We can make some modifications to form a *class-dependent* SVM classifier for our data. In our experiment, we will choose the second approach and realize the classifier using the “one-against-all” idea, which is also consistent with our feature selection process. In Fig. 4.1, we present the classifier’s architecture by using LIBSVM to train and test the patterns with *class-dependent* feature subsets. We describe the classification in the following two processes, given N training data $(X_1, y_1), \dots, (X_j, y_j), \dots, (X_N, y_N)$, where $X_j \in R^D (j = 1, \dots, N)$ and $y_j \in \{1, 2, \dots, c, \dots, C\}$ is the class of X_j .

CHAPTER 4. CLASSIFICATION OF CLASS-DEPENDENT FEATURE SUBSETS AND ONE OF ITS APPLICATIONS

Figure 4.1: Architecture of the general classifier with class-dependent feature subsets (after training and during testing).

CHAPTER 4. CLASSIFICATION OF CLASS-DEPENDENT FEATURE SUBSETS AND ONE OF ITS APPLICATIONS

(i) The training process:

In this process, we use training patterns to construct C SVM models, where C is the number of classes. The c -th SVM model is trained using all of the training examples in class c with positive labels and all the other training examples (not in class c) with negative labels. What's more, all training samples are filtered by the feature mask of class c . Then all training samples has M^c features left for training the c -th model. For instance, given a training sample X and its class label y , we use X^c to denote the training sample to form the c -th model and M^c to denote the feature number of X^c . Here, M^c is the number of '1' in the feature mask of class c . $y = +1$ if X^c belongs to class c and $y = -1$ otherwise. The classifier in Fig. 4.1 includes several models. Each model will solve one problem. For example, the c -th SVM model solves the following problem (see Eq. (4.16)) [115]:

$$\begin{aligned}
 \min_{\vec{w}^c, b^c, \xi^c} \quad & \frac{1}{2}(\vec{w}^c)^T \cdot \vec{w}^c + C^c \sum \xi^c \\
 & (\vec{w}^c)^T \cdot \phi(X^c) + b^c \geq 1 - \xi^c, \text{ if } y = +1, \\
 & (\vec{w}^c)^T \cdot \phi(X^c) + b^c \leq -1 + \xi^c, \text{ if } y = -1, \\
 & \xi^c \geq 0
 \end{aligned} \tag{4.16}$$

Here all indices c denote the c -th SVM model, $\vec{w}^c$ denotes a linear weight vector connecting the feature space to the output space, ϕ denotes a transformation function which is to map input X^c to a high dimensional space, C^c is a tradeoff parameter between the error and margin, and ξ^c are "slack variables" in optimization theory. Minimizing $\frac{1}{2}(\vec{w}^c)^T \cdot \vec{w}^c$ is equivalent to maximizing the margin between two classes of data. When data sets are not

CHAPTER 4. CLASSIFICATION OF CLASS-DEPENDENT FEATURE SUBSETS AND ONE OF ITS APPLICATIONS

linearly separable, $C^c \sum \xi^c$ is used as the penalty term to reduce training errors.

(ii) The testing process:

In the testing stage after training, we classify the input test pattern according to the classifier (Fig. 4.1). The testing sample will be input into all models. Each model has an output, which indicates the probability distinguishing the current class from all other classes in terms of the current class' feature subset. A common way to generate an overall output from individual outputs of all models is to choose the maximum individual as the overall output of the class-dependent classifier, e.g., as Oh et al. [94] and Liu et al. [96] mentioned. However, Baggenstoss pointed out that it is not fair to directly compare outputs produced from different feature spaces [76] [77]. Baggenstoss [77] proposed the PDF projection theorem and utilized it to project those class-specific features' probability density functions (PDFs) back into the original space where a fair comparison is possible. In our thesis, we propose a heuristic method, i.e., a weight measure, to deal with the problem of unfair comparisons on outputs from different feature spaces.

Suppose the output of the c -th model is denoted as O^c ($c \in \{1, 2, \dots, C\}$), which is the probability that one sample belongs to the c -th class when using the c -th feature subset obtained by the c -th model to classify the sample between two classes, i.e., class A (the c -th class) and class B (all other classes except the c -th class). For C models, we have C outputs, i.e., $O^1, O^2, \dots, O^c, \dots, O^C$, each representing the probability that the sample belongs to the current class, respectively. Instead of comparing the C outputs obtained from different feature spaces, we assign a weight to each output

CHAPTER 4. CLASSIFICATION OF CLASS-DEPENDENT FEATURE SUBSETS AND ONE OF ITS APPLICATIONS

and compare the weighted outputs, e.g., $p^c \cdot O^c$ in Fig. 4.1. These weights, i.e., $p^1, \dots, p^c, \dots, p^C$ in Fig. 4.1, are obtained on the basis of a common feature space. That is, we first combine all class-dependent feature subsets together to form one union set, i.e., $\cup_{c=1}^C X^c$. Then we classify the sample using the union set as the feature subset with C classes as target classes, together with a classifier, such as an SVM. The classifier produces outputs $p^1, \dots, p^c, \dots, p^C$, each representing the probability that the sample belongs to each of C classes, respectively. These probabilities are calculated according to the method of [143]. We assign these probabilities obtained from the common feature space to the corresponding $O^1, \dots, O^c, \dots, O^C$ for a fair comparison. In this case, one testing sample X is classified according to the following equation:

$$\text{class of } X \equiv \operatorname{argmax}_{c \in \{1, \dots, C\}} (p^c O^c) \quad (4.17)$$

The class of X is determined by the maximum of C decision functions. In addition, we should notice these weights should be obtained in the training process. Since class-dependent feature subsets vary with different training samples in the training process, those weights also vary with different learning processes.

In Chapter 3, we obtain only the feature selection results from various kind of feature selection methods. Based on those results, we still can not say the class-dependent feature selection method outperforms the class-independent feature selection method. The key point, we mostly care about, is whether the classification accuracy will be increased more by the class-dependent feature selection method than by the class-independent method. In Table 4.1, we present classification accuracies of 6 data sets. We compare the classification accuracies of several

CHAPTER 4. CLASSIFICATION OF CLASS-DEPENDENT FEATURE SUBSETS AND ONE OF ITS APPLICATIONS

Table 4.1: Comparisons of mean classification accuracy with different measures for different methods (10-fold cross validation). Original refers to the classification accuracy without feature selection.

Data set	Original (%)	RELIEF (%)		CSM (%)	
		Class-independent	Class-dependent	Class-independent	Class-dependent
Breast cancer	96.49	96.34	96.34	96.49	96.49
Ionosphere	94.02	94.59	94.59	95.44	95.44
Thyroid	96.28	95.81	95.81	96.28	95.81
Iris	97.33	95.33	96.67	97.33	96.67
Ecoli	86.61	85.71	86.31	86.61	86.91
Wine	95.51	95.51	96.63	95.51	98.32
Waveform21	85.80	85.80	86.70	85.70	86.10

methods (see Table 4.1): the classification accuracies on the original data without feature selection, those with *class-independent* feature selection method and with *class-dependent* feature selection method, respectively, using the RELIEF weight measure and CSM ranking measures. From the results in Table 4.1, we can see that our class-dependent feature selection method has better classification accuracies compared to the class-independent feature selection method for some data sets, e.g., the Wine and Waveform21 data sets. While for other data sets, e.g., the Breast cancer data set and the Thyroid data set, our class-dependent feature selection maintained nearly the same accuracy compared to class-independent feature selection. For Wine and Waveform21 data sets, the improvement of the classification accuracies can be attributed to the class-dependent feature selection method which selects different feature subsets for different classes (see Table 3.11 and 3.12). This result demonstrates the effectiveness of our method and also justify our assumption in Chapter 3.

CHAPTER 4. CLASSIFICATION OF CLASS-DEPENDENT FEATURE SUBSETS AND ONE OF ITS APPLICATIONS

Table 4.2: Comparisons of classification accuracies among two feature selection methods with output probability not weighted and weighted, with CSM measure and “one-against-all”, for the artificial data A and B.

Feature selection approach (Wrappers)		Mean accuracy $\pm$ std (min/max) (%)	
		Artificial data A	Artificial data B
Without feature selection		93.87 $\pm$ 1.69 (90.67/96.00)	84.53 $\pm$ 2.10 (82.00/87.33)
CSM Without weighted	Class-independent	96.53 $\pm$ 2.28 (93.33/100)	91.53 $\pm$ 1.60 (89.33/94.00)
	Class-dependent	99.87 $\pm$ 0.42 (98.67/100)	99.87 $\pm$ 0.42 (98.67/100)
CSM weighted	Class-dependent	99.73 $\pm$ 0.56 (98.67/100)	99.60 $\pm$ 0.72 (98.00/100)

In Tables 4.2–4.5, the proposed general classifier is used for the class-dependent feature selection method and the normal SVM is used for the class-independent method. In Table 4.2, we can see that for the two artificial data sets, both the *class-independent* and *class-dependent* feature selection methods have greatly increased the classification accuracy, and the *class-dependent* feature selection method has increased more than the *class-independent* feature selection method. The same thing happens for the Waveform40 data set in Table 4.3. For the Segment and Vehicle data sets in Table 4.3, although the increase on the accuracy is not so obvious as that for the two artificial data sets, the *class-dependent* feature selection method produces better results than the *class-independent* feature selection method. Besides, in Table 4.2 and Table 4.3, we list the comparison between the results obtained from the proposed general classifier in Fig. 4.1 and those obtained from the classifier which has not introduced weights to each class-dependent model’s output. The proposed general classifier in Fig. 4.1 consistently produces better results than the classifier without weights on each model’s output.

CHAPTER 4. CLASSIFICATION OF CLASS-DEPENDENT FEATURE SUBSETS AND ONE OF ITS APPLICATIONS

Table 4.3: Same as Table 4.2, for Waveform40, Segment, and Vehicle data sets. The Table also presents the results by the RELIEFF ranking measure without using the weight approach for those data sets.

Feature selection approach (Wrappers)		Mean accuracy $\pm$ std (min/max)(%)		
		Waveform40	Segment	Vehicle
Without feature selection		84.50 $\pm$ 2.22 (80.00/86.00)	96.28 $\pm$ 1.16 (95.24/98.70)	79.43 $\pm$ 3.28 (75.29/85.88)
CSM + Without weighted	Class-independent	85.00 $\pm$ 2.75 (80.00/89.00)	96.32 $\pm$ 1.26 (95.24/99.13)	79.45 $\pm$ 2.55 (76.47/83.53)
	Class-dependent	86.60 $\pm$ 2.72 (83.00/90.00)	96.41 $\pm$ 0.98 (94.94/97.97)	80.16 $\pm$ 3.53 (75.29/87.06)
CSM + weighted	class-dependent	87.10 $\pm$ 2.08 (85.00/90.00)	96.80 $\pm$ 0.89 (95.67/98.70)	81.93 $\pm$ 2.43 (78.82/85.88)
RELIEFF + without weighted	class-independent	83.90 $\pm$ 3.51 (78.00/90.00)	96.62 $\pm$ 1.10 (95.24/98.70)	79.67 $\pm$ 3.23 (76.47/85.19)
	class-dependent	85.60 $\pm$ 2.88 (80.00/92.00)	96.49 $\pm$ 0.81 (95.67/98.27)	80.50 $\pm$ 3.75 (75.29/85.19)

We also list some published results on numbers of selected features with *class-independent* feature selection methods SOAP [144] and RELIEFF with threshold 0.05 and classification accuracies with the 1 Nearest-Neighbor (1NN) classifier (Table 4.4). Ruiz et al. [144] adopted the filter feature selection methods to select features, it had good performance for some data sets. For example, the Ionosphere data set originally had 34 features with an accuracy 86.78%. With the SOAP method, on average 31.55 features were selected and the accuracy was 87.07%. With the RELIEFF method, on average 30.88 features were selected with an accuracy 87.49%. But the performance of those filter feature selection methods was not stable. For example, in the Vehicle data set, the SOAP method had 6 features selected from original 18 features with an accuracy 46.5%, and the RELIEFF method had on average 5.81 features selected with an accuracy 61.28%. Although the number of features was greatly reduced, but the classification accuracies were far less

CHAPTER 4. CLASSIFICATION OF CLASS-DEPENDENT FEATURE SUBSETS AND ONE OF ITS APPLICATIONS

Table 4.4: Total numbers of selected features and the classification accuracies for SOAP and RELIEFF methods [144].

Data set	Feature numbers for SOAP	Accuracy for SOAP with 1NN	Feature numbers for RELIEFF	Accuracy for RELIEFF with 1NN
Waveform40	12.99	79.33%	5.77	73.09%
Segment	7	91.29%	15.04	97.19%
vehicle	1.09	46.50%	5.81	61.28%

than the original accuracy 69.48% without feature selection [144]. In these cases, we believe our wrapper approach with class-dependent feature selection method is better compared to these filter methods, in terms of stable performance of maintaining or improving the classification accuracy.

In Table 4.5, we make the following comparisons about the classification results: between class-independent and class-dependent feature selection using the CSM and mRMR measures, respectively, by the “one-against-all” strategy, between the “one-against-all” strategy and the “one-against-one” strategy for class-dependent feature selection using the mRMR measure, between class-independent and class-dependent feature extraction using the “one-against-all” strategy and between feature selection and feature extraction methods (i.e., KFDA). The mRMR ranking measure tends to have better performance than the CSM ranking measure. The results obtained (see Table 4.5) by the class-dependent feature selection using the “one-against-all” strategy (see grayed rows) are a little better than those obtained by the “one-against-one” strategy. Besides, the “one-against-one” strategy attempts to be computationally more expensive than the “one-against-all” strategy, especially when the number of classes is large. For feature extraction methods, we can see that the class-dependent feature extraction method produces better accuracies than the class-independent feature extraction method for all data sets

CHAPTER 4. CLASSIFICATION OF CLASS-DEPENDENT FEATURE SUBSETS AND ONE OF ITS APPLICATIONS

Table 4.5: Classification comparisons among different feature selection methods for the HDR, DNA and Scene data sets.

Feature selection approach		Mean accuracy $\pm$ std (min/max)(%)		
		HDR	Scene	DNA
Without feature selection		98.13 $\pm$ 0.45 (97.30/98.70)	78.67 $\pm$ 1.14 (76.09/79.35)	95.89 $\pm$ 0.48 (95.03/96.54)
Feature selection (mRMR + weighted)	class-independent	96.93 $\pm$ 0.42 (96.60/97.80)	77.30 $\pm$ 1.32 (74.99/78.92)	95.71 $\pm$ 0.52 (95.03/96.37)
	class-dependent + "one-against-all"	98.30 $\pm$ 0.60 (97.40/98.90)	78.93 $\pm$ 0.83 (77.56/79.43)	96.60 $\pm$ 0.33 (96.04/96.88)
	class-dependent + "one-against-one"	97.85 $\pm$ 1.70 (93.50/99.00)	73.90 $\pm$ 1.34 (72.16/77.09)	96.12 $\pm$ 0.71 (95.19/97.55)
Feature Selection (CSM + weighted)	class-independent	93.96 $\pm$ 2.20 (91.90/96.70)	73.66 $\pm$ 1.69 (70.57/75.5)	95.64 $\pm$ 0.83 (93.76/96.63)
	class-dependent + "one-against-all"	97.90 $\pm$ 0.39 (97.20/98.30)	78.27 $\pm$ 1.69 (73.66/79.35)	96.19 $\pm$ 0.30 (95.95/96.71)
Feature extrac- tion (KFDA) (Filters)	class-independent	97.23 $\pm$ 0.42 (96.80/97.70)	68.84 $\pm$ 1.31 (65.97/70.32)	95.81 $\pm$ 0.46 (95.03/96.54)
	class-dependent + "one-against-all"	97.52 $\pm$ 0.27 (97.10/98.90)	70.14 $\pm$ 2.28 (68.23/76.22)	95.74 $\pm$ 0.39 (95.19/96.21)

CHAPTER 4. CLASSIFICATION OF CLASS-DEPENDENT FEATURE SUBSETS AND ONE OF ITS APPLICATIONS

except the DNA data. Also our proposed class-dependent feature selection method produces better accuracy than the class-dependent feature extraction method, especially for the Scene data with 8.79% increase in accuracy. This can be explained by the fact that filters are used in the feature extraction methods and wrappers are used for feature selection methods. Beside this, there is another important reason why we do not choose feature extraction methods in our general approach. That is, different from feature selection methods, feature extraction methods make it hard for us to tell the meaning of those features extracted from the original features.

4.3 A novel application of our method on face recognition

Face recognition has always been an active research topic in pattern recognition, since face recognition has a wide range of applications in our daily life. Researchers in different fields, such as engineers, neuroscientists, or psychophysicists, developed different techniques from different points of view. Therefore, novel feature selection approaches or novel applications are always in demand due to different requirements.

In this section, we will demonstrate a novel application of our feature selection method on face recognition. This novel application is not to show whether or not our method outperforms all other existing methods on classification performance, but intends to show that class-dependent feature selection is better than class-independent feature selection. In this case, it is possible for us or other researchers to combine our proposed feature selection method with existing feature extraction methods so that better performance is obtained. Since face recognition is another different topic, we will first give a brief review about the history of face

CHAPTER 4. CLASSIFICATION OF CLASS-DEPENDENT FEATURE SUBSETS AND ONE OF ITS APPLICATIONS

recognition and then put forward to apply our method on this problem. The relevant experimental process, results and discussions are presented in the following subsections.

4.3.1 Background and Method

Automatic face recognition has experienced a relatively long process from the 1960s until now. In the 1960s, the first semi-automated face recognition system was developed [145]. The recognition process of this system included locating features of eyebrows, eyes, noses and so on, calculating distances and ratios to a common reference point, and template matching. Later in the 1970s, some subjective features like hair color and lip thickness were developed by Goldstein et al. [146] to automate the recognition system. However, these two early solutions have the drawback of manual computation. In order to deal with this problem, in the 1980s, Kirby and Sirovich [147] applied PCA to extract important information by singling out important face features. This application brought out the well-know eigenface [52] and was thought of the first successful example of automatic face recognition systems. After that, another new algorithm, namely fisherface [52], was developed by using PCA to reduce dimensions for LDA. More techniques like the variants of PCA [148], [149], fisher linear discriminant analysis (FLDA) [150], [149], [151], general tensor discriminant analysis (GTDA) [152], and independent component analysis (ICA) [153], have been used to extract face features [151]. Among those methods, PCA [148], [53],[154], [149], [155] has the most application in face recognition.

Besides feature extraction, feature selection is another way of obtaining compact face information. Feature selection employs geometrical facial features and geometric relationships between them, such as the distance between eyes, the width

CHAPTER 4. CLASSIFICATION OF CLASS-DEPENDENT FEATURE SUBSETS AND ONE OF ITS APPLICATIONS

of nose, and the length of jaw line [156], [157]. By feature selection, a feature vector is obtained to represent one face image. Although features obtained by this method can be easier to interpret than those by feature extraction, it is not cost-effective work to determine which features are desirable for face images and then calculate those features. In this case, feature extraction is more reliable than feature selection to obtain fully representative information of face images. Therefore, we will first use PCA to preprocess face images.

For the sake of completeness, we will briefly describe the rationale of PCA here. PCA [52], [53], [49] calculates principal components of images, thereby transforming training images (denoted as matrix M with N samples in rows and D features in columns) into a new space of the principal components. The basic steps are: (1) calculating the covariance matrix of data matrix M ; (2) determining eigenvalues and eigenvectors of this covariance matrix; (3) selecting k ($k < d$) significant eigenvectors to form transformation matrix M_{tr} with the first row corresponding to the most important eigenvectors; and (4) obtaining the projected images by calculating $H^T = M_{tr}M^T$, here H is a matrix with N rows and k columns. H^T and M^T denote the transposition matrix of H and M . Through PCA, the dimension of original images is reduced to k . The dimension of kept components decides the amount of information lost. Therefore, in our experiment, we may decide a large dimension to keep enough information for further feature selection. For selecting class-dependent features from high dimensional data, it is impractical for us to sequentially add each feature and evaluate all feature subsets. For the example of the ORL face database [9], we will use it as our experimental data. Since it has a high dimension, each time we can add 3 features into the previous feature subset. Each feature subset is evaluated by the SVM and the feature subset with the highest classification accuracy is chosen as the superior one of the current class. In

CHAPTER 4. CLASSIFICATION OF CLASS-DEPENDENT FEATURE SUBSETS AND ONE OF ITS APPLICATIONS

order to demonstrate the advantage of our method over class-independent feature selection, we also conduct the experiment on using class-independent feature selection so as to provide a comparison with class-dependent feature selection method. In the end, we provide discussions about the experimental results.

4.3.2 Experiments and Results

We select the Cambridge ORL face database [9] in our experiment. It includes 40 subjects (faces or classes), each of which has 10 slightly different face images. Therefore, the total number of face images is 400. Each face image is in gray scale and has 112 by 92 pixels. For convenient processing, we reshape each original image matrix into a column vector, which has the dimension 10304. All image vectors constitute a new image matrix with 400 samples in rows and 10304 features in columns. Considering the effect of illumination on different face images, we adopted the normalization as [151] described, i.e., subtracting the mean of each image and being divided by the variance. After that, all images were equally distributed in terms of energy [151].

Since normalized face images have a high dimension, PCA is adopted to extract principal components for each face image. The dimension of the original face images is reduced to 399 by PCA. The class-dependent feature selection method is used to further reduce features for each class. Before feature selection, we separate all 400 face images into 200 training images and 200 testing images. For the training set, we calculate features' importance ranking for each class. Although principal components (i.e., features obtained from PCA) are sorted according to another importance measure, i.e., the eigenvalue, we believe that features' rankings for different classes are different. In Figure 4.2, we plot the number of features selected for each class. We can see that class-dependent feature selection method

CHAPTER 4. CLASSIFICATION OF CLASS-DEPENDENT FEATURE SUBSETS AND ONE OF ITS APPLICATIONS

selected rather different features for different classes, i.e., as many as 8 features for several classes and as few as 1 features for other classes (see Figure 4.2). When combining features subsets of all classes in Figure 4.2, we include 26 different features in the union set. Through the SVM, we obtained different feature masks for different classes. For example, in Table 4.6, we provide feature masks of the first 5 classes. For classification, the 200 testing samples were processed in the same way as the training set and tested on a class-dependent SVM classifier to produce a 98% classification accuracy, which is better than the classification result by class-independent feature selection methods (see Table 4.7 and 4.8). Table 4.7 provides classification results of different number of features (principal components) by using normal class-independent feature selection method. The class-independent feature selection method selected 30 features to produce classification accuracy 97.5%. Although the increased accuracy 0.5% is minor for 400 face images, it is very significant for some real applications like identity confirmation or searching for potential criminals and terrorists in attendance at the event from the huge database. Finally we compared our result with that of some existing methods in Table 4.8. Kim et al. tried different number of eigenvalues together with a polynomial parameter of the linear SVM. The best accuracy 97.5% was obtained when the number of eigenvalues is set as 120 and the polynomial degree parameter is set as 6. All those experiments were done on the ORL database but with different recognition methods.

As to the training time of the classifiers, we roughly compare the proposed method with other methods and found the training time of the proposed method is mostly related to the number of classes and the number of selected features. The more the number of the classes and the more the number of selected features, the more the training time of the proposed method. The training time on the

CHAPTER 4. CLASSIFICATION OF CLASS-DEPENDENT FEATURE SUBSETS AND ONE OF ITS APPLICATIONS

Figure 4.2: Number of features selected for each classes using the class-dependent feature selection method.

original data is related to the dimension of the data. The higher the dimension of the data, the more the training time. For those low-dimensional data sets with small classes, such as the Iris data, both the training time on the original data and using the proposed method is less. For the artificial data B with 10 classes and 20 features, since only 1 feature is selected for each class, the training time is also nearly the same as the training time on the original data. For DAN data set, since the number of the classes is small and the number of selected features is also small compared to the original dimension, the training time of the proposed method is less than the training time on the original data. Therefore, the proposed feature selection method can sometimes improve the classification performance in terms of both the training time and the classification accuracy.

CHAPTER 4. CLASSIFICATION OF CLASS-DEPENDENT FEATURE SUBSETS AND ONE OF ITS APPLICATIONS

Table 4.6: Feature masks of the first 5 classes

Class 1	1 1 0 0 0 0 0 0
Class 2	1 1 1 0 0 0 0 0
Class 3	1 1 1 1 0 0 0 0
Class 4	1 1 1 1 1 1 1 0
Class 5	1 1 1 0 0 0 0 0

Table 4.7: Classification accuracy for variant numbers of features using class-independent feature selection method (10 fold cross validation)

Feature number	5	10	15	20	25	30
Accuracy	80.50%	87.50%	92.00%	94.75%	95.25%	97.50%

Table 4.8: Comparisons of classification accuracies with existing methods

Method	Proposed	Class-independent after PCA	Kim et al.'s [148]	Lu et al's [150]
Classifier	SVM with RBF kernel	SVM with RBF kernel	Linear SVM	Nearest Neighbor
Number of selected features	26	30	120	22
Accuracy	98.0%	97.5%	97.5%	96.0%

CHAPTER 4. CLASSIFICATION OF CLASS-DEPENDENT FEATURE SUBSETS AND ONE OF ITS APPLICATIONS

4.4 Summary

In this Chapter, we utilized the support vector machines to realize the classification of class-dependent feature subsets. By comparing the strategies of “one-against-all” and “one-against-one”, we choose the “one-against-all” to built one model for each class. Since each model produced one output for any one unlabeled sample, we proposed a heuristic weight measure to realize the comparison in different feature spaces. In order to prove our proposed general classifier, we conducted classification experiments on two artificial data sets and some publicly available data. For artificial data sets, our class-dependent feature selection method selected a feature subset for each class. All class-dependent feature subsets included only one feature, therefore the number of irrelevant features or redundant features was greatly reduced and the classification accuracy was substantially increased. For other six real data sets, the experiment results further proved that each feature may indeed have different classification capability in discriminating different classes, and our proposed method improved or at least maintained the classification accuracy, while reducing the number of features. For face recognition, although many fully-fledged feature extraction techniques like PCA, LDA and elastic bunch graph matching [158] exist and have good successful experiences, we still proposed this novel application on face recognition to detect if our proposed class-dependent method has advantages in classification performance over the class-independent feature selection method. In the process of experiments, PCA was used as a preprocessing step to extract face features. Then both class-dependent and class-independent feature selection were used to select features for recognition. The classification results showed that the introduction of our proposed method selected features specific for each class (face). Therefore, the feature dimension of each class is less than that produced by the class-independent feature selection. Also a better

CHAPTER 4. CLASSIFICATION OF CLASS-DEPENDENT FEATURE SUBSETS AND ONE OF ITS APPLICATIONS

classification performance is obtained compared to the normal class-independent feature selection methods. Besides, the corresponding face recognition system after class-dependent feature selection has one more advantage over the normal face recognition system: when adding one more class (face) into the existing system, the class-dependent recognition system does not need to re-train the whole system [92]. However, the recognition system based on class-independent feature selection needs to re-train the whole system if one more class is added.

In conclusion, the classification results showed that our general wrapper approach to *class-dependent* feature selection described in Chapter 3 was very effective in deleting irrelevant or redundant features for different classes.

Looking back on existing class-dependent feature selection methods [94], [77], [76], [78], it is necessary for us to state the differences between our method and those methods here. Oh et al. [94] did class-dependent feature selection, whereas feature subsets obtained in [94] were based on a predefined dimension for all classes, unlike our feature subsets which were determined through a classifier. In addition, since the CENPARMI numerical database used in [94] is not publicly available, we did not make a direct comparison between our method and Oh's method [94]. When comparing our method with Baggenstoss' class-specific method [77], [76], [78], we found the following differences. Firstly, in selecting the class-dependent feature subsets, we utilized the strategy of "one-against-all" to construct binary classification problems and select class-dependent features. Baggenstoss proposed a "one-against-reference class" strategy, i.e., introducing a reference class [77], [76], [78] against each class. Because Baggenstoss thought that the strategy of "one-against-all" for class-dependent feature selection lacks theoretical basis for dimensionality reduction [77]. Baggenstoss demonstrated an example in the area of signal processing about selecting the correlated Gaussian noise as the reference

CHAPTER 4. CLASSIFICATION OF CLASS-DEPENDENT FEATURE SUBSETS AND ONE OF ITS APPLICATIONS

class [76] and presented mathematical descriptions on how to select the reference class [77], [78]. Baggenstoss' "one-against-reference class" strategy provides theoretical basis for dimensionality reduction, however, some potential assumptions and constraints [78] are required to determine a reference class. By comparison, the "one-against-all" strategy is easy to be executed in practical applications. For example, both Oh et al. [94] and Liu et al. [96] adopted the "one-against-all" strategy in their class-specific methods. Secondly, in the process of designing our general class-dependent classifier, we propose to assign a weight on each model's output for a fair comparison among those class-dependent feature spaces. Here, each weight is the likelihood a sample belongs to each class when classifying the sample with a union set of various class-dependent feature subsets. Although this weight approach is a heuristic method, it produces good results in our experiments. As to Baggenstoss' class-specific classifier [77], Baggenstoss proposed the PDF projection theorem, i.e., utilizing a J-function to project those class-specific features' PDFs back into the original data space to obtain a fair comparison. In this way, he also settled the curse of dimensionality for PDF estimation in the low-dimensional class-specific feature space. How to effectively adopt Baggenstoss' method in our problems will be a subject to future studies.

Chapter 5

Effective Feature Selection and Classification on the HapMap Genotype Data

In the previous two Chapters, we described our general hybrid approach to class-dependent feature selection and applied it to deal with small and medium size data. In this Chapter, we will describe a biological data with huge dimension and propose a ranking measure.

The structure of this Chapter is arranged as follows: in Section 5.1, the motivation of our experiment is explained. In Section 5.2, the background of the biological data is presented. In Section 5.3, the proposed ranking measure is described in detail. Subsequently, the experimental results and discussions are demonstrated. At last, we make a summary of our work in this Chapter.

5.1 Motivation

Bioinformatics is a new emerged technology which combines statistics, molecular biology and computational technologies to analyze and process biological information, such as gene, DNA, RNA and proteins. With the availability of large volume of biomedical or biological data, the major problem for dealing with this kind of

CHAPTER 5. EFFECTIVE FEATURE SELECTION AND CLASSIFICATION ON THE HAPMAP GENOTYPE DATA

data is the high dimensionality of the input space compared to the relatively small number of available samples. For example, the HapMap genotype data [24] has several million features but with only two hundred samples. The problem of learning from a few data samples in a high-dimensional feature space is often known as the “curse of dimensionality” [159]. Classification algorithms dealing with this kind of problem may suffer from the problem of “overfitting”. In addition, the large number of features would increase the computational cost significantly. Another challenge arises from irrelevant features. Among huge numbers of features in the HapMap data, informative features only occupy a small portion while most of the features are irrelevant for genotyping or classification. Not only does the presence of these irrelevant genes interfere with the discrimination power of the relevant ones, but also incur the difficulty in gaining insights about genes’ biological relevance. For example, a critical concern for biomedical researchers on cancer diagnosis is to identify the marker genes that can discriminate tumors. Therefore, selecting informative genes/features is a very important job in dealing with problems like cancer diagnosis.

5.2 Background

A single nucleotide polymorphism (SNP) is described as a single base variation. For example, when any one single nucleotide of A, T, C and G in the genome sequence is replaced by one of any other 3 nucleotides, e.g., from $AAAT\underline{C}CGG$ to $AAAT\underline{T}CGG$, the single base variation ($C \Rightarrow T$) is called an SNP. According to [160], SNPs have the following three characteristics: 1) very common in the human genome (an SNP occurs every 100 to 300 bases along the 3-billion-base human genome); 2) among all the SNPs, two of every three SNPs are the variations from cytosine (C) to thymine (T); 3) very stable from generation to generation. Being a

CHAPTER 5. EFFECTIVE FEATURE SELECTION AND CLASSIFICATION ON THE HAPMAP GENOTYPE DATA

kind of genetic variations, the SNPs determine the difference between any two unrelated individuals, for example, different physical traits, different predispositions to diseases, and different responses to medicine. Because of these characteristics, various kinds of research on SNPs have been developed. For example, association studies on SNPs are conducted in [21], [161], [162], [48]. The source population classification of an individual is based on SNPs and explain the differences of populations. For efficient population identification on the HapMap genotype data [10], as few informative SNPs as possible are required from the original 4 million SNPs.

Existing research on SNPs can be categorized into two fields: association studies and population studies. In association studies [21], [161], [162], [48], informative SNPs were usually selected based on certain correlation measure and therefore could represent other SNPs in the close proximity. For example, Bafna et al. [21] and Halldrsson et al. [161] proposed to select a subset of tag SNPs with the minimum size and highest informativeness value calculated from a self-defined informativeness measure, which evaluates how well a single SNP or a set of SNPs predict another single SNP or another set of SNPs within the neighborhoods. Eran et al. [162] proposed to select the informative SNPs with the maximum prediction accuracy, which is obtained from a prediction accuracy measure evaluating how well the value of an SNP is predicted by the values of only two closest tag SNPs. Phuong et al. [48] proposed the method of selecting informative SNPs by removing redundant features. Redundancy is measured by the feature similarity between two features, i.e., the linkage disequilibrium (LD) measure [48] γ^2 .

In population studies, SNPs are selected based on their abilities to classify populations. Related research, such as selecting genetic markers with the highest informativeness for inference of individual ancestry [22], selecting informative marker panels for population assignment [23] and detecting ethnically variant SNPs

CHAPTER 5. EFFECTIVE FEATURE SELECTION AND CLASSIFICATION ON THE HAPMAP
GENOTYPE DATA

[24], has already been explored. Rosenberg et al. (2003) proposed to use the informativeness for assignment (I_n) to measure the ability of each genetic loci or marker (feature) to infer individuals' ancestry. In 2005, Rosenberg et al. proposed the univariate, greedy, and maximum algorithms to select marker panels. The three algorithms were realized through a given performance function, e.g., the optimal rate of correct assignment (ORCA) [22], to measure the probability of correctly assigning an individual to the population from which the genotype of the individual has originated with the greatest possibility. Very recently, Park et al. [24] developed a systematic approach based on nearest shrunken centroid method (NSCM) [163] to identify ethnically variant SNPs. According to [163], Park et al. calculated a shrunken value for each SNP of each class, and compared each SNP's different shrunken values for different classes to determine the SNP's classification capability. The less the difference among the SNP's shrunken values for different classes, the less important the SNP for classifying three different ethnic groups (classes) [10]. 100,736 SNPs were obtained and the top 82 SNPs [24] could completely classify the three populations.

5.3 Methods

As we know, for the experimental data with very high dimension, feature ranking [36], [127] is very necessary, since only a very small part of features may be important for the classification. Instead of using the feature ranking measures in Chapter 3, we will present another two feature importance ranking measures which often appear in biomedical or bioinformatics literatures [164], [165], [166] [163], [167], [25]. We modified one of the two ranking measures, i.e., t-test [168], [163], [167], [25], and compare its ranking ability with that of the other ranking measure, i.e., F-statistics [164], [165], [166].

5.3.1 Modified T-test

The original t-test, i.e., the student t-test [168], can be used to evaluate whether the means of two classes are statistically different from each other by calculating a ratio between the difference of means and variability of the two classes. It has been adopted by [167], [169] to rank features (genes) for microarray data and for mass spectrometry data [170], [171]. We notice that the original t-test is only applied on 2-class problems. In the following multi-class problems, Tibshirani et al. [163] developed the nearest shrunken centroid method, i.e., calculating a t-statistic value (Eq. (5.1)) for each gene of each class. This t-statistic value measures the difference of each feature's mean value between one class and all the classes, and the difference is standardized by the within-class standard deviation (see Eq. (5.1)).

$$t_{ic} = \frac{\bar{x}_{ic} - \bar{x}_i}{M_c \cdot (S_i + S_0)} \quad (5.1)$$

$$S_i^2 = \frac{1}{N - C} \sum_{c=1}^C \sum_{j=1}^{n_c} (x_{ij} - \bar{x}_{ic})^2 \quad (5.2)$$

$$M_c = \sqrt{1/n_c + 1/N} \quad (5.3)$$

Here t_{ic} indicates the t-statistics value for the i -th feature of the c -th class, $\bar{x}_{ic}$ indicates the i -th feature's mean value in the c -th class and $\bar{x}_i$ indicates the i -th feature's mean value for all classes, x_{ij} represents the i -th feature of the j -th sample in class c , N is the total number of all the samples for all the C classes and n_c is the number of samples for the c -th class, S_i is the within-class standard deviation and S_0 is set to be the median value of S_i for all the features. M_c is to make $M_c S_i$ be the estimated standard error. This t-statistic value of Tibshirani et al. [163] measured the deviation between each class and the mean of all classes and was

CHAPTER 5. EFFECTIVE FEATURE SELECTION AND CLASSIFICATION ON THE HAPMAP GENOTYPE DATA

used to constitute a classifier. The authors did not refer to using the t-statistic of each class to rank features for all the classes. In [25], Wang et al. extended the t-statistic algorithm to rank features for all the classes. That is, the t-score (t-statistic value) of the feature i is calculated as the largest t-score for all classes:

$$t_i = \max \left\{ \frac{|\bar{x}_{ic} - \bar{x}_i|}{M_c \cdot (S_i + S_0)}, c \in \{1, 2, \dots, C\} \right\} \quad (5.4)$$

Due to the characteristic of the SNP data [10], i.e., with nominal values for each feature (e.g., AA , AT and TT), Eq. (5.4) can not be used on SNP data with nominal values. We proposed a modified t-test ranking method, which can be used to deal with SNP data with vector features. That is, nominal values are converted into vector formats. In the following, we generalized the t-score of each feature in 3 steps:

- (i) Suppose the feature set is $F = \{f_1, \dots, f_i, \dots, f_g\}$, and the i -th feature f_i has m_i different nominal values, i.e., $f_i \in \{x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(m_i)}\}$.
- (ii) Transform each nominal feature value into a vector with the dimension m_i , e.g., $x_i^{(1)} \Rightarrow \vec{X}_i^{(1)} = (0, \dots, 0, 1)$, $x_i^{(2)} \Rightarrow \vec{X}_i^{(2)} = (0, \dots, 1, 0)$, $\dots$, $x_i^{(m_i)} \Rightarrow \vec{X}_i^{(m_i)} = (1, \dots, 0, 0)$. $\vec{X}_i$ is used to denote any one of $\vec{X}_i^{(j)}$ ($j=1, 2, \dots, m_i$).
- (iii) Replace all the numerical features in Eq. (5.4) and (5.2) with those vectors, we obtain Eq. (5.5) and (5.6).

$$t_i = \max \left\{ \frac{|\bar{\vec{X}}_{ic} - \bar{\vec{X}}_i|}{M_c \cdot (S_i + S_0)}, c \in \{1, 2, \dots, C\} \right\} \quad (5.5)$$

$$S_i^2 = \frac{1}{N - C} \sum_{c=1}^C \sum_{j=1}^{n_c} (\vec{X}_{ij} - \bar{\vec{X}}_{ic})(\vec{X}_{ij} - \bar{\vec{X}}_{ic})^T \quad (5.6)$$

Here $\bar{\vec{X}}_{ic}$ denotes the mean value of all vectors $\vec{X}_i^{(j)}$ ($j=1, 2, \dots, m_i$) in the c -th class. The ranking rule is: the larger the t-scores, the more relevant the features.

5.3.2 F-statistics

In our experiment, we will use another ranking measure, i.e., F-statistics, to make a comparison with the modified t-test. The version of F-statistics used in our experiment is based on the definition of [172], which was originally developed by [164] and used in population genetics to describe the level of heterozygosity in a population.

Given a SNP genotype data with C sub-populations and each feature expressed as bi-allelic SNPs (i.e., consisting of any two different nucleotides from the four nucleotides *ATCG*), the F-statistics (F_{st}) is calculated as:

$$F_{st} = Var(p) / (\bar{p} * \bar{q}) \quad (5.7)$$

where p and q are corresponding to the two alleles' frequencies, respectively, in one population. $\bar{p}$ and $\bar{q}$ refer to the two alleles' mean frequencies for all the population classes. Var_p represents the variance (See Eq. (5.8)) of one allele,

$$Var(p) = \sum_{c=1}^C (p_c - \bar{p})^2 / C \quad (5.8)$$

Here, p_c designates the frequency of one allele for the c -th population, and the mean frequency is easy to obtain from:

$$\bar{p} = \sum_{c=1}^C p_c \quad (5.9)$$

The ranking rule is the same as the modified t-test, i.e., the larger the F_{st} value, the more significant the SNP for population classification.

5.3.3 The Classifier

As to the classification assignment, we still choose the support vector machine (SVM) [142], [127] in our experiment because of its attractive features as we mentioned in Chapter 4. It will be used not only in the final classification, but also in

CHAPTER 5. EFFECTIVE FEATURE SELECTION AND CLASSIFICATION ON THE HAPMAP GENOTYPE DATA

the process of feature selection, in which we use it to test different feature subsets and determine the one with the highest classification accuracy. During the classification process, the kernel parameter γ and the penal parameter ν are determined through the double cross-validation method.

5.4 Experiment Preparation

5.4.1 Data Description

The international HapMap Project provides many kinds of data for researchers in the website (<http://www.hapmap.org>), where we downloaded the experimental data. The HapMap genotype data includes four populations: CEU, YRI, JPT and HCB, where CEU represents Utah residents with ancestry from northern and western Europe; YRI represents Yoruba individuals from Ibadan and Nigeria; JPT represents Japanese individuals from Tokyo, and HCB means Han Chinese individuals from Beijing. CEU and YRI each has 90 related samples, i.e., 30 father-mother-offspring trios. After removing the offsprings, 60 unrelated samples are obtained for CEU and also for YRI. For JPT and CHB populations, each of them has 45 unrelated samples. Therefore, we obtain 210 unrelated samples for the experiment. Since 4 populations are provided, respectively, by the HapMap Project, we will do the classification on the 4-population besides on the 3-population problem as [24] did.

After obtaining the genotype data, we notice that there are missing values for some individuals at some loci. For recovering missing values, some existing genotype phasing techniques, such as PHASE [173], minimum entropy population phasing [174], 2SNP [175] and genotype phasing based on minimum entropy (ENT) [176], are available. Park et al. [24] adopted a different method to recover those missing SNPs, i.e., imputing those missing values with the major allele of each

population class. We will firstly adopt the PHASE technique [173] to preprocess our data. After that, in later section, we will also adopt the preprocessing method of Park et al. [24] to make a comparison between our method with Park et al.'s.

5.4.2 Data Preprocessing 1

In the international HapMap Project [10], we notice that the phased haplotype data is provided for researchers besides the HapMap genotype data from the directory of (Index of /downloads/phasing/2006-07_phaseII/phased). The phased haplotype data is the one whose missing values have been recovered by the PHASE technique [173], [176]. The legend data and sample data in the same directory are also necessary since the legend data and sample data describe locus places (feature IDs), locus names (feature names), and sample names (individual IDs). The phased haplotype data is also called as the HapMap Phase II haplotype data. The haplotype data describes SNP alleles on a chromosome inherited from one of father and mother, whereas the genotype data describes SNP alleles on both chromosomes inherited from parents [176]. We give an example (see Fig. 5.1. (a)) to describe the relationship between the haplotype and genotype. For one individual, the combination of two alleles of haplotype 1 and haplotype 2 is one allele format of the genotype in the same locus.

For most features (loci), their SNP types (feature values) are expressed as bi-allelic SNPs, i.e., consisting of two single alleles from 4 nucleotides *ATCG*. For example, if one SNP consists of the two alleles *A* and *G* (see Fig. 5.1 (b)), all the possible feature values for this locus will be *AA*, *AG* and *GG*, in which *AA* and *GG* are called homozygous, and *AG* is called heterozygous. Since the phased haplotype data has two rows of haplotypes describing one individual, we transform the haplotype data into the genotype format (see Fig. 5.1 (a)) for computational

CHAPTER 5. EFFECTIVE FEATURE SELECTION AND CLASSIFICATION ON THE HAPMAP
GENOTYPE DATA

Figure 5.1: Example of haplotypes and genotypes.

CHAPTER 5. EFFECTIVE FEATURE SELECTION AND CLASSIFICATION ON THE HAPMAP
GENOTYPE DATA

convenience. When transforming data from the haplotype format into the genotype format, we adopt two kinds of transformations used for two algorithms (see Fig. 5.1 (c)). Also because the two ranking algorithms in our experiment can only process numerical data, we need to transform the original nominal values of the SNP genotype data into numerical values. For the t-test ranking measure, if simply transforming nominal values to normal numerical values according to Eq. (5.4) and doing the calculation, it will be possible to lose the meaning of different SNP types. Therefore, we proposed to use vectors to represent different SNP types. For example (see Fig. 5.1. (c)), according to the description of the modified t-test ranking measure, “11” (i.e., AA in Fig. 5.1. (c)) is represented by vector (0, 0, 1), “00” (GG) is represented by vector (0, 1, 0), and “10” (AG) is represented by vector (1, 0, 0). As to the F-statistics ranking measure, since it involves the calculation of two single alleles for each bi-allelic SNPs, we will use 1 and 0 to represent the two different alleles, respectively. For example (see Fig. 5.1 (c)), given the same SNP reference type A/G as the one in the modified t-test ranking measure, we use 1 to represent A and 0 to represent G. Then in each population, we can calculate each allele’s frequency and variation for each population, as well as those values for all the populations. Each SNP’s F-statistics value is calculated from Eq. (5.7). At the same time, we notice some special conditions. For example, if one locus with reference SNP type A/G only has the value AA for all the individuals, the frequency of the SNP allele A will be 100% and the frequency of allele G will be zero. Referring to Eq. (5.7), either $\bar{p}$ or $\bar{q}$ in the denominator will be equal to zero. In fact, this feature has no classification capability for any populations. Therefore, we set the F_{st} value of that feature as zero. In summary, the larger the numerator and the smaller the denominator in Eq. (5.7), the larger the value F_{st} and the more important the corresponding feature for classification.

5.4.3 Implementation 1

In the experimental process, we conduct 4 simulations, i.e., 4 different combinations of two rankings (the modified t-test and F-statistics) and two classifications (on 3 populations and 4 populations, respectively). From the 210 samples, we randomly choose 40 samples from YRI and CEU, respectively, and 30 samples from JPT and CHB, respectively, as the training set. The 70 samples left are used as the testing set. Each simulation is repeated for 30 times.

In this part, since the phased data of Chromosome Y is not available in the directory, 23 chromosomes' SNPs are combined together. We first rank the SNPs of 23 chromosomes, respectively. Then we choose each chromosome's top 100 SNPs, combine them together, and rank them again. In this way, features involved in the experiment are greatly reduced. This will not lead to the loss of important information. On the contrary, it will improve the efficiency of the experiment.

5.4.4 Results of Preprocessing 1 and Discussions

In each of 30 simulations, we select top 100 SNPs from the ranking list and form 100 different SNP subsets. The first subset consists of only the first ranked SNP. The second subset is formed by adding the second ranked SNP into the previous subset. Sequentially, all the other subsets are formed. This is the sequential forward selection (SFS) method mentioned in Chapter 2. Subsequently, we evaluate all subsets through the classifier SVM in terms of the classification accuracy. Due to the space limitation, we provide classification results of only 11 feature subsets, i.e., the subsets consisting of 1, 10, 20, 30, 40, 50, 60, 70, 80, 90 and 100 SNPs, respectively. Figure 5.2 shows the mean classification accuracies and standard deviations for the F-statistics ranking measure and the modified t-test ranking

CHAPTER 5. EFFECTIVE FEATURE SELECTION AND CLASSIFICATION ON THE HAPMAP GENOTYPE DATA

Figure 5.2: Classification accuracy results obtained by the F-statistics measure for different feature subsets with different numbers of top ranked features (SNPs) in 30 simulations, on 3 and 4 populations, respectively.

CHAPTER 5. EFFECTIVE FEATURE SELECTION AND CLASSIFICATION ON THE HAPMAP
GENOTYPE DATA

measure. Intuitively, the classification on 3 populations produces higher accuracies than on 4 populations, for both ranking measures. This can be interpreted by the fact that the JPT and CHB populations have very similar DNA sequences and it is hence hard to discriminate between these two populations. When comparing classification results by the F-Statistics ranking measure with those by the modified t-test ranking measure on 3 populations, we can see that mean accuracies produced by the modified t-test are higher than those by the F-statistics for most SNP subsets (see Figure 5.3). The superiority of the modified t-test over the F-statistics measure is more obvious when classifying 4 populations than classifying 3 populations. In addition, we compare the minimal and maximal accuracies for each of those 11 SNP subsets in the 30 simulations for the two ranking measures. It shows us that a complete classification of 3 populations is possible for the modified t-test ranking measure with appropriate SNP subsets. Whereas, there is always a little error (i.e., $1/70$) for the F-statistics measure with those 100 SNP subsets.

In the following, we find the subset which leads to the maximal classification accuracy from the 100 SNP subsets (see Table 5.1). We list the maximal classification accuracy in each of the 30 simulations, the number of SNPs that the relevant SNP subset includes, and the mean values ($\pm$ standard deviations) in the 30 simulations. Although the average number of SNPs that leads to the best classification is similar to each other for both ranking measures (see the 5th column of Table 5.1), the mean classification accuracies produced by two ranking measures are different (see the 2nd column of Table 5.1). The modified t-test ranking measure produces 97.09% mean accuracy, which is 1.04% higher than the accuracy produced by the F-statistics measure, i.e., 96.05%, for 3 populations. The mean accuracy produced by the modified t-test on 4 populations, i.e., 83.86%, is much higher than that produced by the F-statistics measure, i.e., 77.34%.

CHAPTER 5. EFFECTIVE FEATURE SELECTION AND CLASSIFICATION ON THE HAPMAP GENOTYPE DATA

Table 5.1: Classification results for 3 populations and 4 populations, respectively.

Feature Numbers	Maximum accuracy (%)	Mean accuracy±std (%)	Relevant feature numbers	Average number of features±std
f-statistics on 3 populations	94.29 98.57 95.71 97.14 94.29 94.29 95.71 95.71 95.71 97.14 97.14 95.71 97.14 95.71 98.57 92.86 97.14 97.14 97.14 92.86 97.14 95.71 95.71 95.71 97.14 95.71 98.57 95.71 97.14 92.86	96.05±1.58 (%)	12 85 15 42 29 86 37 56 78 90 71 46 79 79 49 8 75 74 74 100 83 50 81 67 38 82 81 93 84 53	63.6 ±25.8
f-statistics on 4 populations	78.57 78.57 85.71 70.00 88.57 78.57 80.00 82.86 70.00 80.00 65.71 78.57 82.86 68.57 81.43 80.00 70.00 78.57 81.43 74.29 72.86 84.26 74.29 84.29 88.57 68.57 75.71 64.29 75.71 70.00	77.34±6.57 (%)	53 98 82 96 91 100 93 81 99 90 59 99 88 55 73 81 56 99 100 100 74 74 98 94 88 72 81 99 98 90	85.2 ±15.1
Modified t-test on 3 populations	95.71 100.00 95.71 98.57 94.29 98.57 95.71 95.71 95.71 98.57 95.71 95.71 98.57 100.00 100.00 97.14 98.57 95.71 97.14 98.57 98.57 95.71 94.29 94.29 97.14 97.14 95.71 98.57 97.14 98.57	97.09±1.74 (%)	27 84 19 90 29 95 64 80 84 80 78 54 83 92 57 11 80 79 78 95 62 75 94 84 53 10 32 28 68 55	64.0 ±26.5
Modified t-test on 4 populations	82.86 77.14 87.14 82.86 82.86 84.29 88.57 80.00 81.43 82.86 82.86 84.29 87.14 84.29 85.71 81.43 82.86 84.29 84.29 75.71 81.43 84.29 81.43 84.29 90.00 84.29 85.71 85.71 85.71 90.00	83.86±3.16 (%)	31 84 92 94 86 100 84 99 83 87 96 61 83 93 99 95 80 81 88 99 99 75 95 82 95 51 83 73 99 55	84.1 ±16.3

CHAPTER 5. EFFECTIVE FEATURE SELECTION AND CLASSIFICATION ON THE HAPMAP GENOTYPE DATA

Figure 5.3: Classification accuracy results obtained by the modified t-test measure for different feature subsets with different numbers of top ranked features (SNPs) in 30 simulations, on 3 and 4 populations, respectively.

After determining the subset leading to the maximal classification accuracy in each of the 30 simulations, we need further determine what those SNPs are and which chromosomes those SNPs locate on. From the result in Table 5.1, we know there are on average 64 SNPs obtained for the desirable feature subset. Because of space limitation, we will not list all those SNPs. For example, in Table 5.2, we list 22 SNPs whose appearance frequencies are greater than 83.33% (i.e., appearing more than 25 times in the 30 simulations), mean ranking values and locations of these SNPs, using the F-statistics ranking measure on 3 populations. Similarly, we present results obtained by the modified t-test ranking measure in Table 5.3, in which 24 top ranked SNPs whose appearance frequencies are greater than 83.33% in the 30 simulations are presented. For both ranking methods, most of the SNPs come from chromosome 11 (chr11), except rs35397 from the chromosome 5 (chr5), rs2296224 from chromosome 1 (chr1) and rs199138 from chromosome 15 (chr15).

CHAPTER 5. EFFECTIVE FEATURE SELECTION AND CLASSIFICATION ON THE HAPMAP GENOTYPE DATA

Table 5.2: Top ranked features whose appearance frequencies are greater than 83.33% (25/30) in 30 simulations, and their mean ranking values by the F-statistics ranking measure for 3 populations.

Ranking No. on Mean Ranking Values	Name of SNPs	Chromosome	Mean ranking values in 30 simulations	Ranking No. on Ap- pearance Frequency
1	rs232045	chr11	0.9573	7
2	rs12786973	chr11	0.9547	6
3	rs7946015	chr11	0.9544	2
4	rs4756778	chr11	0.9524	3
5	rs7931276	chr11	0.9521	9
6	rs4823557	chr11	0.9518	5
7	rs10832001	chr11	0.9506	4
8	rs35397	chr5	0.9491	8
9	rs11604470	chr11	0.9480	12
10	rs10831841	chr11	0.9478	11
11	rs2296224	chr1	0.9456	10
12	rs12286898	chr11	0.9387	13
13	rs1869084	chr11	0.9341	20
14	rs4491181	chr11	0.9307	26
15	rs1604797	chr11	0.9258	1
16	rs7931276	chr11	0.9161	14
17	rs11826168	chr11	0.9103	19
18	rs477036	chr11	0.9072	16
19	rs7940199	chr11	0.9032	22
20	rs4429025	chr11	0.8711	25
21	rs6483747	chr11	0.8435	17
22	rs199138	chr15	0.8417	18

Among 22 SNPs in Table 5.2, rs1604797 and rs7946015 appear 30 times in the 30 simulations. Among 24 SNPs in Table 5.3, rs1604797, rs7946015 and rs10832001 appear 30 times in the 30 simulations. Interestingly, the mean ranking values of these SNPs with the highest appearance frequencies are not the highest.

All experiments are executed on MATLAB 7.1 on a personal computer with Windows XP operating system and Pentium 4 CPU (3.4GHZ) and 1GHZ RAM. We perform statistics about the training and testing time of the two ranking mea-

CHAPTER 5. EFFECTIVE FEATURE SELECTION AND CLASSIFICATION ON THE HAPMAP GENOTYPE DATA

Table 5.3: Top ranked features whose appearance frequencies are greater than 83.33% (25/30) in 30 simulations, and their mean ranking values by the modified t-test ranking measure for 3 populations.

Ranking No. on Mean Ranking Values	Name of SNPs	Chromosome	Mean ranking values in 30 simulations	Ranking No. on Ap- pearance Frequency
1	rs232045	chr11	8.0956	7
2	rs1869084	chr11	8.0886	9
3	rs4756778	chr11	8.0079	6
4	rs11218714	chr11	8.0047	11
5	rs10832001	chr11	7.9810	3
6	rs7946015	chr11	7.8988	2
7	rs11826168	chr11	7.8517	4
8	rs704737	chr11	7.7786	18
9	rs1083184	chr11	7.7778	24
10	rs16913196	chr11	7.7774	13
11	rs12786973	chr11	7.7499	5
12	rs12286898	chr11	7.7421	12
13	rs11604470	chr11	7.7401	16
14	rs35397	chr5	7.7257	8
15	rs7931276	chr11	7.7060	14
16	rs477036	chr11	7.6644	17
17	rs6483747	chr11	7.6625	19
18	rs7931276	chr11	7.5996	15
19	rs1604797	chr11	7.3847	1
20	rs10836565	chr11	7.3053	10
21	rs2296224	chr1	7.1358	21
22	rs4275650	chr11	7.0043	23
23	rs7924569	chr11	6.9431	20
24	rs2582905	chr11	6.9264	22

CHAPTER 5. EFFECTIVE FEATURE SELECTION AND CLASSIFICATION ON THE HAPMAP
GENOTYPE DATA

tures. It costs on average 5342.9 seconds for the F-statistics to rank all SNPs of 3 populations, while on average 5728.7 seconds for the modified t-test ranking measures. It may be because that calculating the median value S_0 makes the modified t-test ranking measure take more time than the F-statistics measure. Both algorithms cost more time on 4 populations than on 3 populations. The total training and testing time is 6915.1 seconds. In terms of classification accuracy, the modified t-test ranking measure is superior over the F-statistics measure. Besides, the modified t-test ranking measure is proposed to deal with vector features and provides a way for ranking nominal features.

5.4.5 Data Preprocessing 2

Different from the previous preprocessing (i.e., the PHASE technique), we recover those missing values by the method of Park et al [24]. The experimental data is the latest updated genotype data, which is available from the directory of (/Index of/genotypes/latest.ncbi_build36-/rs_strand/non-redundant). As to the transformation of nominal values to normal numeric values, it is the same as that of the previous part.

5.4.6 Implementation 2

Since chromosome Y is available for the genotype data, we will have data from 24 chromosomes. We first rank the SNPs of 24 chromosomes, respectively. Then we choose each chromosome's top 100 SNPs (except chromosome Y which only has 49 SNPs), combine the 2349 features together, and rank them again. We conduct the same 4 simulation assignments as those in the previous part.

5.4.7 Results of Preprocessing 2 and Discussions

In each of 30 simulations, we found that as few as 10 features can have very good classification. Therefore, we list the top 10 features obtained from F-statistics ranking measure in Table 5.4. The ranking list for 3 populations is the same as the one for 4 populations, whereas ranking values are different for the two conditions. Based on the ranking list in Table 5.4, if we use the SFS method to input different number of features into the SVM classifier, i.e., the first 1, 2, 3, 4, 5, ... and 10 features, respectively, we find out that the top five features can classify the 3 populations with the accuracy 100%, while they classify the 4 populations with the accuracy 57.14%. Similarly in Table 5.5, we list the top 11 features obtained from the modified t-test ranking measure. We can see that the ranking order is still the same for the 3-population and 4-population problems. Inputting different number of features into the classifier, e.g., 1, 2, 3, ..., and 11 features, respectively, we conclude that the top 11 features can completely classify the 3 populations, i.e., 100%. When using the same 11 features to classify the 4 populations, we obtain the accuracy 78.57% (55/70), in which 55 of 70 testing samples are correctly recognized. It shows that CEU and YRI are completely recognized, while both JPT and CHB are recognized as the third class. Besides, in the third column of Table 5.4- 5.5, we also provide the chromosomes on which the ranked SNPs locate.

Since some features have the same ranking values, we intend to further reduce the number of features from the candidate feature subsets obtained in Table 5.4- 5.5. We suppose that the features with the same ranking values have the same classification capability. Therefore, in the process of forming feature subsets, we avoid combining features with the same ranking values together. For example, we can form the feature subset $\{1, 2\}$, by combining feature 1 and feature 2. Likewise, we form the feature subset $\{1, 3\}$, $\{1, 4\}$, $\{1, 5\}$. As to the 4th feature, because it

CHAPTER 5. EFFECTIVE FEATURE SELECTION AND CLASSIFICATION ON THE HAPMAP
GENOTYPE DATA

Table 5.4: Top 10 features' ranking lists for the F-statistics ranking measure, for 3 populations and 4 populations.

Ranking No.	Name of SNPs	Chromosomes	Ranking values for 3 populations	Ranking values for 4 populations
1	rs11499	chr3	0.9669	0.9630
2	rs9820633	chr3	0.9053	0.9273
3	rs9876284	chr3	0.8710	0.8571
4	rs5825	chr4	0.8710	0.8571
5	rs2132498	chr4	0.8247	0.8070
6	rs4143483	chr4	0.8247	0.8070
7	rs16891982	chr5	0.8247	0.8070
8	rs7772008	chr6	0.8095	0.7907
9	rs1299386	chr7	0.8095	0.7907
10	rs1813166	chr7	0.8095	0.7907

Table 5.5: Same as Table 5.4, top 11 features' ranking list for the modified t-test ranking measure.

Ranking No.	Name of SNPs	Chromosomes	Ranking values for 3 populations	Ranking values for 4 populations
1	rs11499	chr3	9.6017	9.5666
2	rs5825	chr4	8.1264	8.1022
3	rs4143483	chr4	7.2546	7.2281
4	rs1299386	chr7	7.2546	7.2281
5	rs1813166	chr7	6.7457	6.7210
6	rs2040513	chr7	6.7457	6.7210
7	rs4131595	chr7	6.7457	6.7210
8	rs289632	chr8	6.5661	6.5421
9	rs1785847	chr18	6.5661	6.5421
10	rs2474273	chrX	6.5661	6.5421
11	rs4120141	chrX	6.4379	6.4144

CHAPTER 5. EFFECTIVE FEATURE SELECTION AND CLASSIFICATION ON THE HAPMAP GENOTYPE DATA

Table 5.6: Classification Accuracy on the 3 populations and 4 populations, using different feature subsets formed from the ranking list (F-statistics) in Table 5.4

Feature subsets	Accuracy for 3 pop- ulations	Accuracy for 4 pop- ulations
{1, 2}	71.43%	57.14%
{1, 3}	71.43%	57.14%
{1, 5}	100.0%	57.14%
{1, 4}	70.00%	55.71%
{1, 6}	71.43%	57.14%
{1, 7}	71.43%	57.14%
{1, 2, 5}	100.0%	57.14%
{1, 2, 3}	71.43%	57.14%
{1, 2, 3, 4}	70.00%	57.14%
{1, 2, 3, 4, 5}	100.0%	57.14%

has the same ranking value as the 3rd feature, we will not include the 3rd and 4th feature together. The result shows that the combination {1, 5} leads to a perfect classification accuracy (100%). We also list some other combinations' results for comparison (see Table 5.6). From the experimental result in Table 5.6 (using F-statistics measure), we can finally determine the best feature subset is {1, 5}, i.e, (*rs11499* and *rs2132498*). Likewise, we form different feature subsets from the result obtained in Table 5.5, test them for the 3-population problem. The best feature subset can be ($\{1, 8\} \Leftrightarrow \{rs11499, rs289634\}$) or ($\{1, 11\} \Leftrightarrow \{rs11499, rs4120141\}$), either of which leads to 100% classification accuracy (see Table 5.7). As to the classification on 4 populations, the accuracy always maintains at 57.14% or so for different feature subsets. Even though we use more features to classify the 4 populations, the accuracy has little change. This may due to the fact that CHB and JPT have very similar DNA sequences.

We conduct the experiment (i.e., preprocessing 2) with the intention to make

Table 5.7: Same as Table 5.6, features subsets are formed form the ranking list (modified t-test) in Table 5.5

Feature subsets	Accuracy for 3 pop- ulations	Accuracy for 4 pop- ulations
{1, 2}	70.00%	55.71%
{1, 3}	71.43%	55.71%
{1, 5}	71.43%	57.14%
{1, 7}	71.43%	57.14%
{1, 8}	100%	57.14%
{1, 11}	100%	57.14%
{1, 9}	71.43%	57.14%
{1, 10}	98.57%	55.71%
{1, 2, 8}	98.57%	57.14%
{1, 2, 11}	98.57%	57.14%
{1, 2,..., 11}	100%	78.57%

a comparison with the result of Park et al [24]. Through the experiment, we concluded that both of the two ranking measures can determine a feature subset which not only has very small size, but also produces a perfect classification. As to the difference between the two ranking measures, the modified t-test determined two desirable feature subsets, while F-statistics determined one feature subset. When comparing the results in Table 5.6 and in Table 5.7, we can see that the modified t-test ranking measure demonstrated a little superiority over F-statistics on classifying 4-populations.

5.5 Summary

In this Chapter, we demonstrated the problem of population classification. We proposed to use two feature importance ranking measures, i.e., the modified t-test and F-statistics, to rank large amount of SNPs, and then through a classifier to

CHAPTER 5. EFFECTIVE FEATURE SELECTION AND CLASSIFICATION ON THE HAPMAP GENOTYPE DATA

determine a desirable feature subset, which has a small size but leads to a very high classification accuracy. In the preprocessing phase, we tried two kinds of preprocessing methods to recover missing values. One technique is PHASE [173], [176] and the other one is what Park et al. mentioned in [24]. These two different preprocessing measures lead to rather different ranking results and classification results. Also different informative SNPs were found by two different preprocessing methods. Since the missing data recovered by the PHASE technique is adopted more often than the imputing measure of Park's [24], we will recommend to use this PHASE technique to recover missing data. However, we can not say the preprocessing method of Park et al. is wrong. The experiments just show that different preprocessing methods have very great effects on the final results.

Besides recovering missing values, we also transform SNP data from the nominal format into the numerical format. To use the modified t-test ranking measure to ranking SNPs, SNP data is transformed into vector format. To use F-statistics to ranking SNPs, the common transformation is used. After ranking SNPs, those informative SNPs with top ranking values are determined in a greedy manner. The modified t-test ranking measure is shown to be more efficient than the F-statistics measure, especially for the classification of 4 populations.

Chapter 6

Conclusions

Although there are many feature selection and classification methods, novel feature selection methods are still in demand due to different situations. For the example of Chapter 3, given the data on patient information, certain symptoms may indicate certain kind of specific illness. Therefore, it is possible to determine each kind of illness by one or some symptoms specific to that illness. This initiated our intention to propose a class-dependent feature selection framework. For the example of Chapter 5, given biological data with nominal features, by reviewing those ranking measures commonly used in biological fields, we propose a modified ranking measure so as to deal with data with nominal features.

In this Chapter, we will first make a summary about the methods proposed in this dissertation. Then we will discuss the advantages and disadvantages of these methods. At last, we will deploy some possible work in the future.

6.1 Summary of Work

In this dissertation, our research work is two-fold: feature selection and classification. In Chapter 1, we introduced some knowledge related to feature selection and classification, and concluded that the aim of feature selection in this dissertation is to simplify the expression of samples and hence improve the efficiency and

CHAPTER 6. CONCLUSIONS

performance of learning algorithms. Classification is the final objective and can be used to detect the effect of feature selection. In Chapter 2, we reviewed existing feature selection methods according to different categories. In Chapter 3, we proposed the general hybrid approach to selecting class-dependent feature subsets. This method provides a framework in which any one filter and wrapper can be combined to rank features and determine the desirable feature subset. For filters, we selected three kinds of feature ranking measures like RELIEF [107], CSM [33] and mRMR [40] in our experiment. For the selection of wrappers, we discussed only the SVM classifier because of many advantages of the SVM. Each of the three filters is combined with the SVM to do the feature subset selection. Besides, we want to emphasize that our method is class-dependent. Class-dependent feature selection is to consider the fact that different features have different classification powers for different classes, whereas class-independent (class-common) feature selection methods disregard this fact. In Chapter 4, we reviewed some background knowledge of the SVM and discussed the classification of class-dependent feature subsets in this dissertation. Since class-dependent feature subsets are selected, we utilize the “one-against-one” scheme to construct a classifier. More importantly, we proposed a heuristic weight measure to make the comparison among different feature subspaces possible. At the end of Chapter 4, a real application of our proposed method is presented. In Chapter 5, a modified t-test ranking method is proposed based on the characteristics of data given. The result shows the modified t-test method is superior over F-statistics and can deal with features in the format of vectors.

6.1.1 Feature Importance Ranking Measures

In Chapters 3, we studied three kinds of feature importance ranking schemes: RELIEF/RELIEFF [107], the CSM [33] and the mRMR [40]. RELIEF/RELIEFF

CHAPTER 6. CONCLUSIONS

is based on the information of instances to find relevant features. Although either of them can not determine the redundancy of features, RELIEF/RELIEFF is efficient in finding relevant features. Base on the rule to make the within-class distance smaller and the between-class distance large, the CSM is used to find relevant features. It also can not detect the redundancy of features. The mRMR ranking measure is recently proposed and has very good effect on ranking features. The mRMR utilizes the theory of mutual information to determine the relevancy and also redundancy of features. Due to the involvement of mutual information, the calculation of the mRMR ranking measure is more complex than RELIEFF and the CSM. However, the feature subset determined by the mRMR is theoretically better than RELIEF and the CSM for classification.

In Chapter 5, we studied two ranking measures used on biological data, i.e., t-test [163], F-statistics [172] [164], with the aim of applying one of them onto the SNP data. T-test is originally used for two-class problems. Wang et al. [46] modified it to rank features for multi-class problems. We modified the version of [46] and utilized it to deal with vector features transformed from nominal features. F-statistics is based on allele frequency to measure alleles' importance for different populations. Our study shows that both two methods can rank SNPs and locate important SNPs. The classification results show that the modified t-test method leads to better classification performance than the F-statistics method.

6.1.2 The Notion of “Hybrid”

If only filter approaches are utilized in the class-dependent feature selection, it is difficult to make sure the selected feature subset is what we want since filters decide feature subsets by some thresholds. If only wrapper approaches are adopted, they can find the desirable feature subset through various search methods. While

CHAPTER 6. CONCLUSIONS

compared to filters, wrappers need more computation time. The combination of filters and wrappers take advantages of pure filters and wrappers. It is based on a filter's ranking and a wrapper's evaluation on feature subsets to determine one feature subset. Compared to pure filter approaches, This hybrid method assures better classification accuracy. Compared to pure wrapper approaches, the hybrid method is computationally less expensive. As to the search for feature subsets, the optimal feature subset can be obtained only through exhaustive search. However, if considering the computation time, the exhaustive search is not allowed. In this dissertation, we tried to find a desirable feature subset through a greedy manner, which firstly includes the most relevant feature, then the second most relevant feature, sequentially all the other features. We make sure the desirable feature subset is as small as possible in size. At the same time, we improve and at least maintain the classification accuracy produced by all features. Experimental results show that the hybrid method can save time in computation and also greatly reduce the size of feature subsets, especially for data with much noise.

6.1.3 The Notion of “Class-dependent”

Compared to class-independent feature selection, class-dependent feature selection is computationally more expensive, since class-dependent feature selection selects a feature subset for each class. However, it demonstrates the fact that different features have different classification capabilities for different classes. It is true that if each class has its own completely different feature subset, class-dependent feature subsets tend to produce better classification accuracies than class-independent feature subsets. However, different feature subsets represent different feature spaces and outputs produced from different feature spaces can not be directly compared [76] [77]. A heuristic weight measure is proposed by us with the aim of seeking for

a fair comparison among different feature spaces. Although the proposed weight measure has a weak theoretical background, compared to the PDF projection theorem of [76] [77], it provides better classification results than the class-independent feature selection method, which achieves our aim in this dissertation.

6.1.4 Current Limitations and Future Works

Both the general hybrid approach to class-dependent feature selection and the modified t-test algorithm have their limitations. The computation time of the first algorithm has a positive linear relationship with the number of classes. More classes the problems include, more expensive the computation time is. This will limit the application of class-dependent feature selection approaches. We can improve this by seeking more efficient feature importance ranking measures, or better classification of class-dependent feature subsets. Besides these, two possible directions in the future research on classification can be: selecting relevant samples [31] [55] and investigating a taxonomy classification of different classes. For the taxonomy classification, several features may be able to completely classify some classes from other classes in the first level. Then in the second level, further classification assignments are developed. This process can be described as rough classification and fine classification.

The second algorithm (the modified t-test) is proposed to find some informative SNPs for highly efficient population classification. Although we determine some SNPs and obtain a good classification, we are not sure if all those SNPs are biologically important in real-world life. If not all SNPs, then which SNPs are important? For example, our algorithms rank SNPs and form SNP subsets using those SNPs with top ranking values. The desirable SNP subset provides a good classification of three populations in the simulation. However, it is very likely that only a few

CHAPTER 6. CONCLUSIONS

SNPs have important biological meanings. In addition, SNPs with low ranking values can not be definitely useless. They may be also important in some senses. It seems that our work lacks a link with the biological domains, which may lead to resources' waste and also hinder the development of bioinformatics' research work. Since our research on SNPs is only to find relevant SNPs, the correlation between SNPs should be studied so that the size of SNP subsets can be further reduced. We should notice that data preprocessing (missing data recovery) has very important effects on SNP selection and hence population classification. Therefore, classical genotype phasing techniques, such as PHASE [173], are preferred for recovering the missing information.

Publications

Journals

1. Lipo Wang, Nina Zhou and Feng Chu, "A general wrapper approach to selection of class-dependent features," *IEEE. Trans. on Neural Networks*, vol. 19, no. 7, pp. 1267-1278, July 2008.
2. Nina Zhou and Lipo Wang, "Efficient feature selection of informative SNPs and classification on the SNP genotype data," *BMC Bioinformatics* [online], 8(484), 2007. Available: <http://www.biomedcentral.com/1471-2105/8/484>
3. Nina Zhou and Lipo Wang, "A modified t-test feature selection method on the Hapmap genotype data," *Genomics, Proteomics & Bioinformatics (GPB)*, vol.5, no. 3-4, pp. 242-249, 2007. Available: www.sciencedirect.com/science/journal/16720229
4. Nina Zhou and Lipo Wang, "Processing bio-medical data with class-dependent feature selection," *International Journal of Information Technology and Intelligent Computing (IT&IC)*, no. 4, 2008.

Book chapter

1. Nina Zhou and Lipo Wang, "A novel support vector machine with class-dependent features for biomedical data," in *Pattern Recognition Technologies and Applications: Recent Advances*, Verma and Blumenstein, Eds., London, UK: Information Science Reference, 2008, pp. 284-298.

Conferences

1. Nina Zhou and Lipo Wang, "Face recognition with class-dependent features," *International Workshop on Biometrics*, 29 June, 2007, Hongkong.
2. Nina Zhou and Lipo Wang, "Perfect population classification on Hapmap data with a small number of SNPs," in the *14th International Conference on Neural Information Processing (ICONIP 2007)*, Lecture Notes in Computer Science, vol. 4985, pp. 789-797, 2007.
3. Nina Zhou and Lipo Wang, "A novel support vector machine with class-dependent features for biomedical data," *IEEE Conference on System, Man and Cybernetics*, vol. 2, pp. 1666-1670, 2006.
4. Lipo Wang, Hou Chai Quek, Keng Hoe Tee, Nina Zhou and Chunru Wan, "Optimal size of a feedforward neural network: how much does it matter?," *Proceedings of Joint International Conference on Autonomic and Autonomous Systems and International Conference on Networking and Services*, pp. 69-73, 2005.

Bibliography

- [1] P. Zhang, B. Verma, and K. Kumar, "Neural vs. statistical classifier in conjunction with genetic algorithm based feature selection," *Pattern Recognition Letters*, vol. 26, no. 7, pp. 909–919, 2005.
- [2] S. Mitra and T. Acharya, "Bioinformatics: an application," in *Data Mining: Multimedia, Soft Computing and Bioinformatics*. 2004, USA: Wiley and Interscience.
- [3] L. P. Wang, S. K. Halgamuge, and X. Yao, *Proceedings of the 1st International Conference on Fuzzy Systems and Knowledge Discovery: Computational Intelligence for the E-Age*, vol. 2, Orchid Country Club, Singapore, Available at: <http://dblp.uni-trier.de>, Nov. 2002.
- [4] H. H. Hsu, *Advanced Data Mining Technologies in Bioinformatics*, Hershey PA, USA: Information Science Reference, 2006.
- [5] X. Q. Zhu and L. Davidson, *Knowledge Discovery and Data Mining: Challenges and Realities*, Hershey PA, USA: Information Science Reference, 2007.
- [6] O. Sourina and N. Korolev, "Geometric querying of time-dependent data for data mining in Molecular Dynamics," *presented at the 2004 International Conference on Cyberworlds*, pp. 351–355, 2004.

BIBLIOGRAPHY

- [7] P. Zhang and L. H. Chen, "A novel feature extraction method and hybrid tree classification for handwritten numeral recognition," *Pattern Recognition Letters*, vol. 23, no. 1-3, pp. 45–56, 2002.
- [8] C. L. Blake and C. J. Merz, "UCI repository of machine learning databases," Technical report, Department of Information and Computer Science, University of California. Irvine, CA, 1998.
- [9] AT & T Laboratories Cambridge, *The ORL Databases of Faces*, Available at: http://www.cl.cam.ac.uk/Research/DTG/attarchive:pub/data/att_faces.zip, 1992-1994.
- [10] The International HapMap Consortium, "The international hapmap project," *Nature*, vol. 426, pp. 789–796, 2003.
- [11] S. Halgamuge and L. P. Wang, *Classification and Clustering for Knowledge Discovery*, Berlin: Springer, 2005.
- [12] S. Halgamuge and L. P. Wang, *Computational Intelligence for Modeling and Predictions*, Berlin: Springer, 2005.
- [13] M. Dash and H. Liu, "Feature selection for classification ," *Intelligent Data Analysis: An Intelligent Journal*, vol. 1, no. 3, pp. 131–156, 1997.
- [14] B. Verma and P. Zhang, "A novel neural-genetic algorithm to find the most significant combination of features in digital mammograms," *Appl. Soft Comput.*, vol. 7, no. 2, pp. 612–625, 2007.
- [15] S. Chao and L. H. Chen, "Feature dimension reduction for microarray data analysis using locally linear embedding," *presented at the 3rd Asia-Pacific Bioinformatics Conference*, vol. 1, pp. 211–217, 2005.

BIBLIOGRAPHY

- [16] L. A. Zadeh, "Fuzzy logic neural networks and soft computing," *Communications of the Association for Computing Machinery (ACM)*, vol. 37, pp. 77–84, 1994.
- [17] Z. R. Yang, "Biological applications of support vector machines," *Briefings in Bioinformatics*, vol. 5, no. 4, pp. 328–338, 2004.
- [18] Z. R. Yang and K.-C. Chou, "Mining biological data using self-organizing map," *Journal of Chemical Information and Computer Sciences*, vol. 43, no. 6, pp. 1748–1753, 2003.
- [19] M. J. Daly, J. D. Rioux, S. F. Schaffner, T. J. Hudson, and E. S. Lander, "High-resolution haplotype structure in the human genome," *Nature Genetics*, vol. 29, pp. 229–232, 2001.
- [20] H. Nguyen J M. Moore J. Roy B. Blumenstiel J. Higgins M. DeFelice A. Lochner M. Faggart S. N. Liu-Cordero C. Rotimi A. Adeyemo R. Cooper R. Ward E. S. Lander M. J. Daly D. Altshuler S. B. Gabriel, S. F. Schaffner, "The structure of haplotype blocks in the human genome," *Science*, vol. 296, pp. 2225–2229, 2002.
- [21] V. Bafna, B. Halldorsson, R. Schwartz, A. Clark, and S. Istrail, "Haplotypes and informative SNP selection: don't block out information," in *Proceedings of Research in Computational Molecular Biology (RECOMB)*, Berlin, Germany, 2003, pp. 19–27.
- [22] N. A. Rosenberg, L. M. Li, R. Ward, and J. K. Pritchard, "Informativeness of genetic markers for inference of ancestry," *American Journal of Human Genetics*, vol. 73, pp. 1402–1422, 2003.

BIBLIOGRAPHY

- [23] N. A. Rosenberg, "Algorithms for selecting informative marker panels for population assignment," *Journal of Computational Biology*, vol. 12, pp. 1183–1201, 2005.
- [24] J. S. Park, S. Y. Hwang, Y. S. Lee, and S.-C. Kim, "SNP@Ethnos: a database of ethnically variant single-nucleotide polymorphisms," *Nucleic Acids Research*, , no. 35, pp. D711–5, 2006.
- [25] L. P. Wang, F. Chu, and W. J. Xi, "Accurate cancer classification using expressions of very few genes," *IEEE Trans. on Computational Biology and Bioinformatics*, vol. 4, no. 1, pp. 40–53, 2007.
- [26] Z. R. Yang, "Mining gene expression data based on template theory," *Bioinformatics*, vol. 20, no. 16, pp. 2759–2766, 2004.
- [27] S. Schulze-Kremer, "Genetic algorithms for protein tertiary structure prediction," in *Proceedings of the European Conference on Machine Learning*. 1993, vol. 667, pp. 262–279, London, UK: Springer-Verlag.
- [28] P. Langley, *Elements of Machine Learning*, Berlin: Morgan Kaufmann, 1996.
- [29] A. L. Rendell and R. Sheshu, "Learning hard concepts through constructive induction: framework and rationale," *Computational Intelligence*, vol. 6, no. 4, pp. 247–270, 1990.
- [30] D. Koller and M. Sahami, "Toward optimal feature selection," in *Proceedings of the 13th International Conference on Machine Learning*, Morgan Kaufmann, 1996, pp. 284–292.
- [31] B. Raman and T. R. Ioerger, "Enhancing learning using feature and example selection," Technical report, Department of Computer Science, Texas A&M University, Tex. USA, 2003.

BIBLIOGRAPHY

- [32] P. Langley and S. Sage, "Induction of selective Bayesian classifiers," in *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence*, 1994, pp. 399–406.
- [33] X. J. Fu and L. P. Wang, "Data dimensionality reduction with application to simplifying RBF network structure and improving classification performance," *IEEE Trans. on System, Man, and Cyber.-Part B: Cyber.*, vol. 33, no. 3, pp. 399–400, 2003.
- [34] J. Bins, "Feature selection from huge sets in the context of computer vision," Ph.D thesis. Colorado State University, Colorado, 2000.
- [35] A. K. Jain and B. Chandrasekaran, "Dimensionality and sample size considerations in pattern recognition practice," in *Handbook of Statistics*, P. R. Krishnaiah and I. N. Kanal, Eds., North-Holland, 1982, vol. 2, pp. 835–855.
- [36] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, no. 7-8, pp. 1157–1182, 2003.
- [37] N. N. Zhou and L. P. Wang, "A modified t-test feature selection method and its application on the HapMap genotype," *Genomics, Proteomics and Bioinformatics*, vol. 5, pp. 242–249, 2007.
- [38] H. Q. Liu, J. Y. Li, and L. S. Wong, "A comparative study on feature selection and classification methods using gene expression profiles and proteomic patterns," *Genome Informatics*, vol. 13, no. 51, pp. 51–60, 2002.
- [39] I. Kononenko, "Estimating attributes: analysis and extensions of RELIEF," in *Proceeding of the European Conference on Machine Learning (ECML)*. 1994, pp. 171–182, Berlin, Heidelberg: Springer-Verlag.

BIBLIOGRAPHY

- [40] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.
- [41] P. M. Narendra and K. Fukunaga, "A branch and bound algorithm for feature selection," *IEEE Trans. on Computers*, vol. 26, no. 9, pp. 917–922, 1977.
- [42] K. Kira and L. A. Rendell, "The feature selection problem: traditional methods and a new algorithm," in *Proceedings of 10th National Conference on Artificial Intelligence (AAAI92)*. 1992, pp. 129–134, San Jose, CA: AAAI Press.
- [43] G. H. John, R. Kohavi, and K. Pfleger, "Irrelevant features and the subset selection problem," in *Proceedings of the 11th International Conference on Machine Learning*. 1994, pp. 121–129, Los Altos, CA: Morgan Kaufmann.
- [44] S. X. Yu, "Feature selection and classifier ensembles: a study of hyperspectral remote sensing data," Ph.D thesis, The University of Antwerp, Antwerp, 2003.
- [45] B. Liu, C. R. Wan, and L. P. Wang, "An efficient semi-supervised gene selection method via spectral biclustering," *IEEE Trans. on Nano-Bioscience*, vol. 5, no. 2, pp. 110–114, 2006.
- [46] L. P. Wang, F. Chu, and W. Xie, "Accurate cancer classification using expressions of very few genes," *IEEE Trans. on Bioinformatics and Computational Biology*, vol. 4, no. 1, pp. 40–53, 2007.

BIBLIOGRAPHY

- [47] L. Zhen and R. B. Altman, "Finding haplotype tagging SNPs by use of principle components analysis," *American Journal of Human Genetics*, vol. 75, no. 5, pp. 850–861, 2004.
- [48] T. M. Phuong, Z. Lin, and R. B. Altman, "Choosing SNPs using feature selection," in *Proceedings of IEEE Computational Systems Bioinformatics Conference (CSB'05)*, 2005, pp. 301–309.
- [49] L. P. Wang and X. J. Fu, *Data Mining with Computational Intelligence*, Berlin: Springer, 2005.
- [50] H. Liu and H. Motoda, *Feature Extraction, Construction, and Selection: A Data Mining Perspective*, Springer, 1 edition, 1998.
- [51] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*, NK: Wiley, 1973.
- [52] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," *IEEE Trans. on Pattern Analysis and machine intelligence*, vol. 19, no. 7, pp. 711–720, 1997.
- [53] S. W. Joo, "Face recognition using PCA and FDA with intensity normalization," Project Work. Available at: http://www.cs.umd.edu/swjoo/reports/739Q_report.pdf, 2003.
- [54] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artificial Intelligence*, vol. 97, no. 1-2, pp. 245–271, 1997.
- [55] G. V. Lashkia and L. Anthony, "Relevant, irredundant feature selection and noisy example elimination," *IEEE Trans. on Systems, Man, and Cybernetics-part B: Cybernetics*, vol. 34, no. 2, pp. 888–897, 2004.

BIBLIOGRAPHY

- [56] P. Mitra, C. A. Murthy, and S. K. Pal, "Unsupervised feature selection using feature similarity," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 304–312, 2002.
- [57] C. R. Rao, *Linear Statistical Inference and Its Applications*, NK: John Wiley, 1973.
- [58] N. Kwak and C. H. Choi, "Input feature selection by mutual information based on parzen window," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 12, pp. 1667–1671, 2002.
- [59] H. Almuallim and T. G. Dietterich, "Learning Boolean concepts in the presence of many irrelevant features," *Artificial Intelligence*, vol. 69, no. 1-2, pp. 279–305, 1994.
- [60] L. Hyafil and R. L. Rivest, "Constructing optimal binary decision trees is NP-complete," *Information Processing Letters*, vol. 5, no. 1, pp. 15–17, 1976.
- [61] A. L. Blum and R. L. Rivest, "Training a 3-node Neural Network is NP-complete," *Neural Networks*, vol. 5, no. 1, pp. 117–127, 1992.
- [62] M. Mocciardi and E. E. Gose, "A comparison of seven techniques for choosing subsets of pattern recognition properties," *IEEE Trans. on Computers*, vol. C-20, no. 9, pp. 1023–1031, 1971.
- [63] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Trans. on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 491–502, 2005.
- [64] W. Siedlecki and J. Sklansky, "On automatic feature selection," *International Journal on Pattern Recognition and Artificial Intelligence*, vol. 2, no. 2, pp. 197–220, 1988.

BIBLIOGRAPHY

- [65] P. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*, Englewood Cliffs, London: Prentice-Hall Int., 1982.
- [66] J. X. Li, "Feature construction, selection and consolidation for knowledge of discovery," Ph.D thesis, Graduate College, The University of Arizona, U.S, 2007.
- [67] A. Jain and D. Zongker, "Feature selection: evaluation, application, and small sample performance," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, pp. 153–158, 1997.
- [68] M. Dash and H. Liu, "Feature selection for clustering," in *Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*. 2000, pp. 110–121, Kyoto, Japan: Springer.
- [69] P. A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*, NK: Prentice Hall, 1982.
- [70] D. W. Aha and R. L. Bankert, "A comparative evaluation of sequential feature selection algorithms," in *Artificial intelligence and statistics*, V. D. Fisher and J. H. Lenz, Eds. 1996, NK: Springer-Verlag.
- [71] P. Pudil, J. Novovicova, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognition Letters*, vol. 15, no. 11, pp. 1119–1125, 1994.
- [72] D. Skalak, "Prototype and feature selection by sampling and random hill climbing algorithms," in *Proceedings of the 11th International Conference on Machine Learning (ICML)*, New Brunswick, NJ, USA, 1994, pp. 293–301.
- [73] H. Liu and R. Setiono, "Some issues on scalable feature selection," *Expert Systems with Applications*, vol. 15, no. 3, pp. 333–339, 1998.

BIBLIOGRAPHY

- [74] R. Kohavi and G. H. John, "Wrappers for Feature Subset Selection," *Artificial Intelligence*, vol. 97, no. 1-2, pp. 273–324, 1997.
- [75] H. Almuallim and T. G. Dietterich, "Learning with many irrelevant features," in *Proceedings of 9th National Conference on Artificial Intelligence*, 1991, vol. 2, pp. 547–552.
- [76] P. M. Baggenstoss, "Class-specific feature sets in classification," *IEEE Trans. on Signal Processing*, vol. 47, no. 12, pp. 3428–3432, 1999.
- [77] P. M. Baggenstoss, "The PDF projection theorem and the class-specific method," *IEEE Trans. on Signal Processing*, vol. 51, no. 3, pp. 672–685, 2003.
- [78] P. M. Baggenstoss, "Class-specific classifier: avoiding the curse of dimensionality," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 37–52, 2004.
- [79] J. Doak, "An evaluation of feature-selection methods and their application to computer security," Technical report. Department of computer science, University of California, Davis, 1992.
- [80] K. Torkkola, "Feature extraction by non-parametric mutual information maximization," *Journal of Machine Learning Research (JMLR)*, vol. 3, pp. 1415–1438, 2003.
- [81] G. Forman, "An extensive empirical study of feature selection metrics for text classification," *Journal of Machine Learning Research (JMLR)*, vol. 3, pp. 1289–1305, 2003.

BIBLIOGRAPHY

- [82] W. S. Chow and S.-Y. Cho, "Classification and feature selection," in *Neural Networks and Computing: Learning Algorithms and Applications*. 2007, pp. 275–278, Imperial College Press.
- [83] H. Almuallim and T. G. Dietterich, "Learning with many irrelevant features," in *Proceedings of the 9th National Conference on Artificial Intelligence*, Anaheim, California, 1991, pp. 547–552.
- [84] M. Ginsberg, *Essentials of Artificial Intelligence*, San Francisco, CA, USA: Morgan Kaufmann, 1994.
- [85] C. Cardie, "Using decision trees to improve case-based learning," in *Proceedings of the 10th International Conference on Machine Learning (ICML)*, University of Massachusetts, Amherst, MA, USA, 1993, pp. 25–32.
- [86] H. Liu and R. Setiono, "A probabilistic approach to feature selection – a filter solution," in *Proceedings of the 13th International Conference on Machine Learning*, Bari, Italy, 1996, pp. 319–327.
- [87] R. A. Caurana and D. Freitag, "Greedy attribute selection," in *Proceedings of the 11th International Conference on Machine Learning*. 1994, pp. 28–36, NEW Brunswick: Morgan Kaufmann.
- [88] D. E. Goldberg, "Genetic algorithms in search, optimization and machine learning," in *Data Mining: Multimedia, Soft Computing and Bioinformatics*. 1989, Massachusetts: Addison-Wesley.
- [89] S. Kirkpatrick, Jr. C. Gelatt, and M. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

BIBLIOGRAPHY

- [90] G. Legrand and N. Nicoloyannis, "Feature selection method using preferences aggregation metric framework," in *Machine Learning and Data Mining*, P. Perner and A. Imiya, Eds. 2005, pp. 203–217, Berlin, Heidelberg: Springer-Verlag.
- [91] Z. X. Zhu, Y. S. Ong, and M. Dash, "Wrapper-filter feature selection algorithm using a memetic framework," *IEEE Trans. on Sys., Man, and Cyber. – Part B: Cyber.*, vol. 37, no. 1, pp. 70–77, 2007.
- [92] P. Baggenstoss, "Class-specific feature sets in classification," *presented at the IEEE International Symposium on Intelligent Control (ISIC)*, pp. 413–416, 1998.
- [93] I. S. Oh, J. S. Lee, and C. Y. Suen, "Using class separation for feature analysis and combination of class-dependent features," *The 14th International Conference on Pattern Recognition*, vol. 1, pp. 453–455, 1998.
- [94] I. S. Oh, J. S. Lee, and C. Y. Suen, "Analysis of class separation and combination of class-dependent features for handwriting recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, pp. 1089–1094, 1999.
- [95] X. J. Fu and L. P. Wang, "A GA-based novel RBF classifier with class-dependent features," in *Proceedings of Congress on Evolutionary Computation*, 2002, vol. 2, pp. 1890–1894.
- [96] C. L. Liu and H. Sako, "Class-specific feature polynomial classifier for pattern classification and its application to handwritten numerical recognition," *Pattern Recognition*, vol. 39, no. 4, pp. 669–681, 2006.

BIBLIOGRAPHY

- [97] N. N. Zhou and L. P. Wang, "A novel support vector machine with class-dependent features for biomedical data," in *Pattern Recognition Technologies and Applications: Recent Advances*. 2008, Information Science Reference.
- [98] R. Gilad-Bachrach, A. Navot, and N. Tishby, "Margin based feature selection theory and algorithms," in *Proceedings of the 21th International Conference on Machine Learning*, Banff, Canada, 2004, vol. 21, pp. 337–344.
- [99] X. W. Chen, "An improved branch and bound algorithm for feature selection," *Pattern Recognition Letters*, vol. 24, no. 12, pp. 1925–1933, 2003.
- [100] A. Kandel M. Last and O. Maimon, "Information-theoretic algorithm for feature selection," *Pattern Recognition Letters*, vol. 22, no. 6-7, pp. 799–811, 2001.
- [101] M. Dash, H. Liu, and H. Motoda, "Consistency based feature selection," in *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*. 2000, pp. 98–109, Kyoto, Japan: Springer.
- [102] J. C. Schlimmer, "Efficiently inducing determinations: a complete and systematic search algorithm that uses optimal pruning," in *Proceedings of the 10th International Conference on Machine Learning*, 1993, pp. 284–290.
- [103] T. M. Mithchell, "Machine learning," in *Computer Science Series*. 1997, Singapore: MIT press and The McGraw-Hill Companies.
- [104] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.
- [105] R. Kohavi and G. H. John, "The Wrappers Approach," in *Feature extraction, construction and selection: a data mining perspective*, Huan Liu and Hiroshi Motoda, Eds. 1998, Boston: Kluwer Academic.

BIBLIOGRAPHY

- [106] A. W. Moore and M. S. Lee, "Efficient algorithms for minimizing cross validation error," in *Proceedings of the 11th International Conference on Machine Learning*. 1994, pp. 190–198, NEW Brunswick, NJ: Morgan Kaufmann.
- [107] K. Kira and L. A. Rendell, "The feature selection problem: traditional methods and a new algorithm," in *Proceedings of the 10th National Conference on Artificial Intelligence*, 1992, pp. 129–134.
- [108] Z. R. Yang and M. Zwolinski, "Mutual information theory for adaptive mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 4, pp. 396–403, 2001.
- [109] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, NK: John Wiley & Sons, 1991.
- [110] J. Bins and B. Draper, "Feature selection from huge feature sets," *presented at the IEEE International Conference on Computer Vision (ICCV), Vancouver, BC, Canada*, vol. 2, pp. 159–165, 2001.
- [111] L. H. Chen and S. Chang, "An adoptive learning algorithm for principle component analysis," *IEEE Trans. on Neural Networks*, vol. 6, no. 5, pp. 1255–1263, 1995.
- [112] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.R. Müller, "Fisher discriminant analysis with kernels," in *Proceedings of the 1999 IEEE Signal Processing Society Workshop, Neural Networks for Signal Processing IX*, 1999, pp. 41–48.
- [113] K. R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. on Neural Networks*, vol. 12, no. 2, pp. 181–201, 2001.

BIBLIOGRAPHY

- [114] Q. S. Liu, H.Q. Lu, and S. D. Ma, "Improving kernel fisher discriminant analysis for face recognition," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 42–49, 2004.
- [115] C. W. Hsu and C. J. Lin, "A Comparison of methods for multiclass support vector machines," *IEEE Trans. on Neural Network*, vol. 13, no. 2, pp. 415–425, 2002.
- [116] J. Friedman, "Another approach to polychotomous classification," Technical report, Department of Statistics, Stanford University, Stanford, CA, 1996.
- [117] J. C. Platt, N. Cristianini, and J. Shawe-Taylor, "Large margin DAGs for multiclass classification," *Advances in Neural Information Processing Systems*, vol. 12, pp. 547–553, 2000.
- [118] N. B. Karayiannis and M. M. R.-Gips, "On the construction and training of reformulated radial basis function neural networks," *IEEE Trans. on Neural Networks*, vol. 14, no. 4, pp. 835–846, 2003.
- [119] J. R. Quinlan, "Unknown attribute values in induction," in *Proceedings of the 6th International Workshop on Machine Learning*, Alberto Maria Segre, Ed. 1989, pp. 164–168, San Francisco, USA: Morgan Kaufmann.
- [120] J. R. Quinlan, *C4.5 Programs for Machine Learning*, San Mateo, CA: Morgan Kaufmann, 1992.
- [121] J. R. Quinlan, "C4.5: programs for machine learning," *Machine Learning*, vol. 16, no. 3, pp. 235–240, 1993.
- [122] P. Domingos and M. Pazzani, "On the optimality of the simple Bayesian classifier under zero-one loss," *Machine Learning*, vol. 29, no. 2-3, pp. 103–130, 1997.

BIBLIOGRAPHY

- [123] I. J. Good, *The Estimation of Probabilities: An Essay on Modern Bayesian Methods*, The MIT Press, 2003.
- [124] V. Vapnik, *The Nature of Statistical Learning Theory*, NK: Springer-Verlag, 1995.
- [125] C. Cortes and V. Vapnik, "Support vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [126] C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.
- [127] L. P. Wang, *Support Vector Machines: Theory and Applications*, Berlin: Springer, 2005.
- [128] I. Guyon, S. R. Gunn, A. Ben-Hur, and G. Dror, "Result analysis of the NIPS 2003 feature selection challenge," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, Vancouver, Canada. Available at: http://books.nips.cc/papers/files/nips17/NIPS2004_0194.pdf, 2004.
- [129] C. C. Chang and C. J. Lin, *LIBSVM: A Library for Support Vector Machines*, Available at: <http://www.csie.ntu.edu.tw/~cjlin/libsvm>, 2001.
- [130] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik, "Feature selection for SVMs," in *Advances in Neural Information Processing Systems*. 2000, vol. 13, pp. 668–674, Cambridge, Massachusetts: MIT Press.
- [131] J. Li and S. Halgamuge, "Improving training speed of support vector machines by creating exploitable trends of lagrangian variables: an application to DNA splice site detection," in *Frontiers in the Convergence of Bioscience and Information Technologies*, 2007, pp. 230–233.

BIBLIOGRAPHY

- [132] S. Hua and Z. Sun, "A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach," *Journal of Molecular Biology*, vol. 308, pp. 397–407, 2001.
- [133] R. Saad, S. Halgamuge, and J. Li, "Polynomial kernel adaptation and extensions to the SVM classifier learning," *Neural Computing and Applications*, vol. 17, no. 1, pp. 19–25, 2008.
- [134] J. CP. Vert, "Introduction to support vector machines and applications to computational biology," *Seminar-SVM in Bioinformatics*, 2001.
- [135] V. Vapnik, *Statistical Learning Theory*, NK: Wiley, 1998.
- [136] C. W. Hsu, C. C. Chang, and C. J. Lin, "A practical guide to support vector classification," Technical report, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, 2003.
- [137] J. S. Denker H. Drucker I. Guyon L. D. Jackel Y. LeCun U. A. Muller E. Sackinger P. Simard L. Bottou, C. Cortes and V. Vapnik, "Comparison of classifier methods: a case study in handwriting digit recognition," *presented at the International Conference on Pattern Recognition*, vol. 2, no. 9-13, pp. 77–82, 1994.
- [138] S. Knerr, L. Personnaz, and G. Dreyfus, "Single-layer learning revisited: a stepwise procedure for building and training a neural network," *In Neurocomputing: Algorithms, Architectures and Applications*, 1990.
- [139] U. Krebell, *Pairwise Classification and Support Vector Machines*, Cambridge, MA, USA: MIT Press, 1999.

BIBLIOGRAPHY

- [140] K. Crammer and Y. Singer, "On the learnability and design of output codes for multiclass problems," in *Proceedings of the 13th Annual Conference on Computational Learning Theory*, 2000, pp. 35–46.
- [141] J. Weston and C. Watkins, "Multi-class support vector machines," in *Proceedings of the European Symposium on Artificial Neural Networks (ESANN)*, Brussels. Available at: http://platon.cs.rhbnc.ac.uk/research/compint/areas/comp_learn/sv/pub/report98-04.ps.gz, 1999.
- [142] V. Vapnik, *Statistical Learning Theory*, NK: Wiley, 1998.
- [143] C. J. Lin T.-F. Wu and R. C. Weng, "Probability estimates for multi-class classification by pairwise coupling," *Journal of Machine Learning Research*, vol. 5, pp. 975–1005, 2004.
- [144] R. Ruiz, J. S. Aguilar-Ruiz, and J. C. Riquelme, "SOAP: efficient feature selection of numeric attributes," *presented at the Ibero-American Conference on Artificial Intelligence (IBERAMIA)*, vol. 2527, no. 8, pp. 233–242, 2002.
- [145] S. Z. Li and A. K. Jain, *Handbooks of Face Recognition*, NK: Springer, 2005.
- [146] A. J. Goldstein, L. D. Harmon, and A. B. Lesk, "Identification of human faces," in *Proceedings of the IEEE*, 1971, vol. 59, pp. 748–760.
- [147] L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *Journal of the Optical Society of America*, vol. 3, no. 4, pp. 519–524, 1987.
- [148] K. I. Kim, K. C. Jung, and H. J. Kim, "Face recognition using kernel principal component analysis," *IEEE Signal Processing Letters*, vol. 9, no. 2, pp. 40–42, 2002.

BIBLIOGRAPHY

- [149] C. J. Liu, "Gabor-based kernel PCA with fractional power polynomial models for face recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 5, pp. 572–581, 2004.
- [150] J. W. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Face recognition using LDA-based algorithms," *IEEE Trans. on Neural Networks*, vol. 14, no. 1, pp. 195–200, 2003.
- [151] S. Vishnubhotla, "Face recognition," Project Work. Available at: <http://terpconnect.umd.edu/~srikanth/ENEE.633.Report.1.pdf>, 2005.
- [152] D. C. Tao, X. L. Li, X. D. Wu, and S. J. Maybank, "General tensor discriminant analysis and gabor features for gait recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 29, no. 10, pp. 1700–1715, 2007.
- [153] C. J. Liu and H. Wechsler, "Independent component analysis of Gabor features for face recognition," *IEEE Trans. on Neural Networks*, vol. 14, no. 4, pp. 919–928, 2003.
- [154] V. Perlibakas, "Distance measures for PCA-based face recognition," *Pattern Recognition Letters*, vol. 25, no. 6, pp. 711–724, 2004.
- [155] H. T. Zhao, P. C. Yuen, and J. T. Kwok, "A novel incremental principle component analysis and its application for face recognition," *IEEE Trans. on Sys. Man, and Cyber.-Part B: Cyber.*, vol. 36, no. 4, pp. 873–886, 2006.
- [156] I. Craw, D. Tock, and A. Bennett, "Finding face features," *Lecture Notes in Computer Science*, vol. 588, pp. 92–96, 1992.

BIBLIOGRAPHY

- [157] R. Johnson and K. Bonsor, "How facial recognition systems work," Available at: <http://computer.howstuffworks.com/facial-recognition.htm>, 2005.
- [158] L. Wiskott, J.-M. Fellous, N. Kuiger, and C. von der Malsburg, "Face recognition by elastic bunch graph matching," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 775–779, 1997.
- [159] R. Bellman, *Adaptive Control Processes: A Guided Tour*, USA: Princeton University Press, 1961.
- [160] K. F. Duerinck, Available at: <http://www.duerinck.com/snp.html>, 2001.
- [161] R. Lippert R. Schwartz F. M. De La Vega A. G. Clark B. Halldrsson, V. Bafna and S. Istrail, "Optimal haplotype blockfree selection of tagging snps for genome-wide association studies," *Genome Research*, vol. 14, no. 8, pp. 1633–1640, 2004.
- [162] E. Halperin, G. Kimmel, and R. Shamir, "Tag SNP selection in genotype data for maximizing SNP prediction accuracy," *Bioinformatics*, vol. 21, no. suppl 1, pp. 195–203, 2005.
- [163] R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu, "Diagnosis of multiple cancer types by shrunken centroids of gene expression," in *Proceedings of the National Academy of Sciences (PNAS)*, USA, 2002, vol. 99, pp. 6567–6572.
- [164] S. Wright, "The interpretation of population structure by F-statistics with special regard to systems of mating," *Evolution*, vol. 19, no. 3, pp. 395–420, 1965.
- [165] B. S. Weir and C. C. Cockerham, "Estimating f-statistics for the analysis of population structure," *Evolution*, vol. 38, pp. 1358–1370, 1984.

BIBLIOGRAPHY

- [166] B. S. Weir and W. G. Hill, "Estimating f-statistics," *Annual Review of Genetics*, vol. 36, pp. 721–750, 2002.
- [167] J. Jaeger, R. Sengupta, and W. L. Ruzzo, "Improved gene selection for classification of microarrays," in *Proceedings of Pacific Symposium on Bio-computing*, 2003, vol. 8, pp. 53–64.
- [168] J. Devore and R. Peck, *Statistics: the Exploration and Analysis of Data*, CA: Duxbury Press, 1997.
- [169] Y. Su, T. M. Murali, V. Pavlovic, M. Schaffer, and S. Kasif, "Rank Gene: identification of diagnostic genes based on expression data," *Bioinformatics*, vol. 19, no. 12, pp. 1578–1579, 2003.
- [170] D. Fishman W. McMurray G. Mor K. Stone D. Ward K. Williams B. Wu, T. Abbott and H. Zhao, "Comparison of statistical methods for classification of ovarian cancer using mass spectrometry data," *Bioinformatics*, vol. 19, no. 13, pp. 1636–1643, 2003.
- [171] I. Levner, "Feature selection and nearest centroid classification for protein mass spectrometry," *BMC Bioinformatics*, vol. 6, no. 68, 2005.
- [172] W. M. Trochim, "The Research Methods Knowledge Base," Atomic Dog Publishing, 2 edition. Available at: <http://trochim.human.cornell.edu/kb/index.htm>, 2004.
- [173] M. Stephens, N. J. Smith, and P. Donnelly, "A new statistical method for haplotype reconstruction from population data," *American Journal of Human Genetics*, vol. 68, no. 4, pp. 978–989, 2001.

BIBLIOGRAPHY

- [174] E. Halperin and R. Karp, “The minimum-entropy set cover problem,” *Theoretical Computer Science*, vol. 348, no. 2, pp. 240–250, 2005.
- [175] D. Branza and A. Zelikovsky, “2SNP: scalable phasing based on 2-SNP haplotypes,” *Bioinformatics*, vol. 22, no. 3, pp. 371–373, 2006.
- [176] A. Gusev, I. I. Mandoiu, and B. Pasaniuc, “Highly scalable genotype phasing by entropy minimization,” *IEEE/ACM Trans. on Computational Biology and Bioinformatics (TCBB)*, vol. 5, no. 2, pp. 252–261, April-June 2008.