Mining minimal discriminative features sets and its applications to gene expression data analysis

Feng, Chu

2008

Feng, C. (2008). Mining minimal discriminative features sets and its applications to gene expression data analysis. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/40532>

<https://doi.org/10.32657/10356/40532>

Mining Minimal Discriminative Feature Sets And Its Applications To Gene Expression Data Analysis

Chu Feng

School of Electrical & Electronic Engineering

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

2008

Acknowledgments

First of all, I would like to express my gratitude to my supervisor, Associate Professor Lipo Wang, for his invaluable and continued guidance throughout my Ph.d study. With numerous discussions with him, I have not only gained a lot of professional knowledge but also learned the spirit of doing research. His constant suggestions, encouragement have greatly inspired me in facing and conquering difficult problems and finally reaching out satisfying solutions. Without his supervision, this dissertation could not have been accomplished.

Being a member in Professor Wang's group is one of most valuable and memorable experience in my life. A lot of work in this dissertation was inspired by the discussions with other members. I thank: Xie Wei, Jin Guosheng, Fu Xiuju, Shi Haixiang, Yan Leipo, Gu Wen, Zhou Nina, and Liu Wen.

Thanks also go towards Nanyang Technological University (NTU) for providing me the full financial aid in the form of research scholarship.

I would also like to thank all the staff and students in the Software Engineering Lab and the Intelligent System Research Lab of NTU, for their selfless help and suggestions.

I appreciate the valuable discussions with Assistant Professor Li Jinming from School of Biological Sciences of NTU about the microarray technologies.

I would like to thank the authors of the Online Biology Book (<http://www.emc.miaricopa.edu/faculty/farabee/BIOBK/BioBookPROTSYn.html>) to provide such a

wonderful textbook and make it freely available and reproducible for educational purposes.

I am grateful to my parents, Chu Hengqi and Cui Zizhen, for their unconditional love and support. I am also grateful to my wife, Liu Jing, for her patience, love, and encouragement.

Abstract

Feature subset selection has been an important problem in machine learning research. Recently, newly appeared data with very high dimensionality, such as microarray gene expression data and text classification data, drive feature subset selection techniques advance speedily.

In this dissertation, we mainly propose two very effective feature subset selection approaches. Since we mainly focus on classification/recognition problems here, we aim to find feature subsets with good discriminative power. We call such feature subset discriminative feature sets (DFSs). In addition, to reduce redundancy or noise contained by DFSs, we also hope the set size should be as small as possible. The first one is a partially exhaustive searching scheme, which searches all the feature combinations among some features with good discriminative power. We have applied this method to the cancer classification problem based on microarray gene expression data. Our simulations confirmed the effectiveness and efficiency of this method. Furthermore, we even found some very small gene combinations usually with two or three genes can achieve perfect separation for the data sets we tested. These small gene combinations even enable us to extract simple diagnostic rules.

In view of the major limitation of the partially exhaustive searching approach, i.e., its huge computing time, we propose the second method which we call active mining discriminative feature sets (AM-DFS). This method is similar with the pool

based active learning in its adaptive ability in the learning process. However, in the pool based active learning, the classifier adaptively obtain new **samples** to improve its accuracy. For the AM-DFS, the classifier adaptively obtain new **features** into the DFS to improve its accuracy. Through this approach, the AM-DFS is able to find very good DFSs efficiently. Because the AM-DFS uses a forward selection scheme, it is much faster than the support vector machine based recursive feature elimination (SVM-RFE) that is a widely-used and successful backward approach for feature subset selection. Moreover, because of its adaptive ability, the AM-DFS achieved comparable results with the SVM-RFE. We have also applied the AM-DFS to the cancer classification problems using gene expression data. These applications further substantiate its effectiveness.

Contents

Acknowledgments	ii
Abstract	iv
Table of Contents	v
List of Figures	x
List of Tables	xxiii
1 Introduction	1
1.1 Data Mining and Pattern Classification	1
1.2 Feature Subset Selection	1
1.3 Biological Background	4
1.4 Our Methodology	8
1.5 Major Contributions of the Dissertation	10
1.6 Organization of the Thesis	11
2 Classification Algorithms	13
2.1 Support Vector Machines	13
2.1.1 An Optimal Hyper-plane for Pattern Recognition	14
2.1.2 A Binary Nonlinear SVM Classifier	19
2.1.3 Multi SVM classifiers	22

2.2	A Fuzzy Neural Network	24
2.2.1	FNN Initialization	25
2.2.2	FNN Training	26
2.2.3	Rule Base Modification	27
3	Searching Algorithms and Feature Importance Ranking	30
3.1	Searching Algorithms for Feature Selection	31
3.1.1	Branch and Bound	31
3.1.2	Sequential Forward Selection	31
3.1.3	Generalized Sequential Forward Selection	32
3.1.4	Sequential Backward Selection	32
3.1.5	Generalized Sequential Backward Selection	32
3.1.6	Plus l - Take Away r Selection	33
3.1.7	Generalized Plus l - Take Away r Selection	33
3.1.8	Floating Search	33
3.1.9	Genetic Algorithm	34
3.2	Data Sets	34
3.2.1	The SRBCT Data Set	34
3.2.2	The Lymphoma Data Set	35
3.2.3	Wisconsin Diagnostic Breast Cancer Data Set	35
3.2.4	Ionosphere Data Set	35
3.3	PCA and SCs	36
3.3.1	Principal Component Analysis	36
3.3.2	Class-Separability Analysis for Gene Selection	36
3.3.3	Fisher-Ratio for Gene Selection	37
3.3.4	A T -Test-Based Gene Selection Approach	38
3.4	Experimental Results	39

3.5	Conclusions	40
4	A Partially Exhaustive Searching Approach	44
4.1	Introduction	44
4.2	Method	46
4.2.1	Step 1: Gene Importance Ranking	46
4.2.2	Step 2: Finding the Minimum Gene Subset	46
4.3	Results	48
4.3.1	Lymphoma Data	48
4.3.2	SRBCT Data	53
4.3.3	Liver Cancer Data	67
4.3.4	GCM Data	73
4.3.5	Results with Class Separability Importance Ranking	75
4.4	Discussion	82
5	Active Mining Discriminative Feature Sets	86
5.1	Introduction	86
5.2	Active Mining Discriminative Feature Set	88
5.2.1	T -Statistics for Selecting Seeds	91
5.2.2	Correction Score	91
5.2.3	Our Algorithm	92
5.2.4	Good DFS Selection	95
5.3	Experimental Results	97
5.3.1	SVM-RFE	97
5.3.2	Leukemia Data Set	98
5.3.3	Colon Data Set	105
5.3.4	Prostate Data Set	107
5.4	Discussion	117

6	Conclusions	120
6.1	Summary	120
6.1.1	Feature Importance Ranking	120
6.1.2	A Partially Exhaustive Searching Algorithm	121
6.1.3	Active Mining Discriminative Feature Sets	121
6.2	Limitations and Future Directions	122
	Author's Publications	124
A	Estimating Error-Dimensionality Relationship	127
A.0.1	Fisher Linear Discriminant	127
A.0.2	Relationship between Dimensionality and Classification Errors	128
	Bibliography	130

List of Figures

1.1	Proteins synthesis process. In step 1 (i.e., transcription), a strand of the DNA double helix is used as a template by the RNA polymerase (a macromolecular machine) to synthesize a messenger RNA (mRNA). After that, the mRNA moves from the nucleus to the cytoplasm. In step 2 (i.e., translation), another macromolecule called ribosome binds to the mRNA and serves as a protein factory and synthesizes the protein according to the genetic information coded in the mRNA. This figure is taken from the Online Biology Textbook (www.emc.maricopa.edu/faculty/farabee/BIOBK/BioBookTOC.html) that is free to use for educational purposes.	7

- 1.2 the process of using microarray chips to compare the expression profiles of the cells in two different conditions. First, some DNA fragments are attached at some fixed spots on a glass (or plastic, silicon) slide. Every single spot contains a huge number of identical DAN fragments and each of these fragments ideally should represent a specific gene. Since one chip usually contains thousands of spots, the expression profiles of these (different) genes could be measured in one experiment. Second, mRNAs from the cells in two different conditions (c.g., cells from a health body and their counterparts from a diseased body) are extracted and labeled with two different fluorescent dyes: a green dye for cells at condition 1 and a red dye for cells at condition 2. Third, liquid solution that contains mixed mRNAs from both conditions are incubated on the glass slide. The mRNAs then attach to their complementary sequences in the spots due to complementary matching. Finally, the glass slide is washed. The amount of samples bound to a spot can be measured by the level of fluorescence emitted when it is excited by a laser. If a spot contains more RNAs from the sample in condition 1, it will be green. If it contains more RNAs from the sample in condition 2, it will be red. If both are equal, a spot will be yellow. If neither are present, a spot appears black. Thus, from the color intensities of each spot, the relative gene expression levels in both samples can be estimated. 9

1.3	The quantifying process of microarray data. To obtain useable gene expression profiles, the raw data, i.e., the images files output by a microarray scanner, should undergo the following steps. First, the color intensity of each spot is measured and compared to the background using a software, which is called image quantization. Second, all the quantities that represent the same gene (no matter in the same array or in different arrays) have to be combined and re-scaled to make different arrays comparable. Finally, all the quantities will be put into a matrix, in which each row usually represents a gene and each column usually represents a condition or sample. .	10
2.1	An optimal hyper-plane for classification in a two dimensional case, for (a) linearly separable patterns and (b) linearly non-separable patterns.	15
2.2	The influence of C to the performance of the classifier. (a) a classifier with a large C (small margin); (b) an overfitting classifier; (c) a classifier with a small C (large margin); (d) a classifier with a proper C	20
2.3	The structure of a binary nonlinear SVM. There are three layers in the SVM, i.e., the input layer, the hidden layer that includes kernel functions, and the output layer. The bias and the weights between the hidden layer and the output are determined by the learning algorithm.	23
2.4	The structure of the FNN.	24

2.5	The initial membership function. Initially, the FNN uses a membership function with two triangles. During the training process, the membership function will also be changed as described in rule base is modification section.	27
2.6	The flow chart of the learning algorithm used by the FNN.	29
3.1	Comparison of classification results in the SRBCT data set using the 50 genes selected by the <i>CS</i> , the <i>TS</i> , Fisher ratio, and PCA, respectively.	41
3.2	Same as Fig. 3.1, for the lymphoma data.	41
3.3	Comparison of classification results in the WDBC data set using the 30 features ranked by the <i>CS</i> , the <i>TS</i> , Fisher ratio, and PCA, respectively.	42
3.4	Comparison of classification results in the Ionosphere data set using the 34 features ranked by the <i>CS</i> , the <i>TS</i> , Fisher ratio, and PCA, respectively.	42
4.1	The test accuracy achieved in the lymphoma data set using different values for η (0.1, 0.075, and 0.05) in the FNN.	49
4.2	The test accuracy achieved in the lymphoma data set using different values for γ (0.1, 0.01, and 0.001) and $C = 10$ in the SVM.	49
4.3	The test accuracy achieved in the lymphoma data set using different values for C (1, 10, and 100) and $\gamma = 0.01$ in the SVM.	50
4.4	The result achieved by the FNN in the lymphoma data set when the top ranked 50 genes were input one by one in the decreasing order of their TSs.	56

4.5	The procedure of the cross validation (CV) used in this dissertation. Here the classifier building and the parameter tuning are carried out during the CV using only the training samples. The testing samples are totally independent from the CV, hence the testing results obtained are unbiased.	57
4.6	Gene expression levels of 2-gene combinations that perfectly separate the 3 types of lymphoma sub-types, i.e., DLBCL, FL, and CLL: (a) (78, 52), (b) (78, 8), (c) (4, 87), (d) (4, 40). Here the genes are labelled according to their t-score (TS) ranks, for example, the two genes in (a) are ranked 78 and 52 according to their TSs. A total of 27 such 2-gene combinations are found by the FNN, which achieved 100% CV accuracy in the training process and also achieved 100% testing accuracy. These combinations are the best performed ones in Table 4.2. However, the performance of the classifier should only be estimated by the averaged accuracy of all the combinations in Table 4.2.	58
4.7	The expression profiles of the 2-gene combinations in lymphoma data set shown in Fig.4.6: (a) (78, 52), (b) (78, 8), (c) (4, 87), (d) (4, 40). Here all the genes are labelled according to their TS ranks. All four combinations can separate DLBCL, FL, and CLL. For example, in (d), gene 4 draws a boundary between DLBCL and other types; gene 40 draws a boundary between CLL and other types. Therefore, Gene 4 and Gene 40 jointly separate the three types.	59

4.8	Gene expression levels of 2-gene combinations that do not separate the 3 lymphoma sub-types: (a) (8, 40), (b) (4, 78). Although these genes have high TS ranks, their combinations cannot separate the 3 subtypes because of poor cooperation.	59
4.9	The procedures of the double cross validation (CV). In the flow, there are two loops, i.e. the inner loop and the outer loop. In the inner loop, genes are firstly ranked according to their TSs. After that, we use all the 2-gene combinations to build classifiers and do internal 10-fold CV. Then, we use the classifier that achieves the best CV accuracy in the internal loop to classify the testing samples in the outer loop. Repeat the process until all the samples are cross validated in the outer loop. Because the testing samples in the outer loop are totally isolated from the gene ranking and classifier building process conducted in the internal loop, the CV result obtained in the outloop are unbiased.	60
4.10	Same as Fig.4.16, for the lymphoma data set.	61

- 4.11 Matrix Winmax for the fold 1 to 5 in the lymphoma data set. Winmax: this matrix contains the best 10-fold CV results in the inner loop. For example, winmax will contain 5 gene combinations if these 5 different gene combinations all made no error for this fold. Since we only tested all the 2-gene combinations, the winmax has 3 columns. The first 2 columns contain the 2 genes. The third column contains the number of errors made by the 2-gene combination. For example, the row "1 6 0" means gene combination (1, 6) made 0 error when classifying the samples in this fold. Here 1 and 6 are TS ranks (ranked using only the internal CV data) of the two genes. The row written in bold and italic letters is the gene combination that we randomly selected to build the classifier. That is, we firstly generate a random number between 1 and the row number of the winmax, after that, we select the row corresponding to this number. The last row of a fold is the number of errors by the randomly selected two genes. 64
- 4.12 Same as Fig. 4.11, for the fold 6 to 10. 66
- 4.13 Three dimensional views of gene expression levels of 3-gene combination (5,12,30) that separates the 4 SRBCT sub-types, i.e., BL, EWS, RMS, NB: (a) a view in which RMS and NB can be seen to be separated from other sub-types, (b) a view in which BL can be seen to be separated from EWS. From the two views, the 4 sub-types are clearly separable. Here all the genes are labelled according to their TS ranks. Our FNN found only 1 such 3-gene combination that achieved 100% 10-fold CV accuracy. 69

4.14	Three dimensional views of gene expression levels of 3-gene combination (7, 13, 18) that separates the 4 SRBCT sub-types, i.e., BL, EWS, RMS, NB: (a) a view that includes only the training samples, (b) a view that includes only the testing samples. Although (b) is a perfect separation of the four sub-types, the combination (7, 13, 18) should be excluded from the evaluation of the classifier because it made one error for the training data and had a lower 10-fold CV accuracy than the combination (5, 12, 30).	70
4.15	Expression levels of Gene 1 and Gene 7 in the liver cancer data set. Here all the genes are labelled according to their TS ranks. The HCCs and the nontumor livers are clearly separable from the plot of (1, 7).	71
4.16	The 10 fold partition of the liver cancer data set. The first line is the No. of the fold, e.g., fold 3., means the 3rd fold. The second line is the samples in a fold. For example, fold 1 includes the samples of 119, 123, 113, 110, 136, etc. Here, these numbers is the No. of the sample in the liver cancer data set. For example, 119 indicates the 119th sample of the liver cancer data set. The third line is the label of the respective sample. Here, 1 stands for HCC; 2 stands for nontumor. The forth line are the labels predicted in the outer loop.	72
4.17	Same as Fig. 4.11, for the liver cancer data set.	73

4.18	Plots of gene combinations for binary classifiers 1-7. In each plot, the horizontal axis stands for gene 1 in Table 4.6 and the vertical axis stands for gene 2 in Table 4.6 for the corresponding classifier. We used 14 such binary classifiers, each responsible for classifying one of the 14 classes from all the rest. As shown in Figs. 4.18 and 4.19, we found a 2-gene combination for each of the 14 classifiers to best separate a given class from the rest. We then used the “one-against-all” scheme (introduced in Section 2.1.3) to obtain the overall classification results from results of the 14 binary classifiers, which leads to a slightly better cross-validation accuracy and the same testing accuracy compared to the best reported result, but with only 28 genes rather than all 16063 genes required by the existing approach.	76
4.19	Same as Fig.4.18, for binary classifiers 8-14.	77
4.20	Expression levels of three combinations after gene replacement. Here all the genes are labelled according to their TS ranks. Genes 71 and 36 are highly correlated with genes 4 and 40, respectively. The combination (4, 40) separates the three lymphoma sub-types very well as shown in Fig.4.6(b). The above plots show the three combinations, i.e., (71, 40), (71, 36), and (36, 4), which are closely correlated with (4, 40).	85

- 5.1 The working principle of an *AL*. Here the *AL* firstly uses the query function Q to search for an unlabelled sample and then assign a label to it using the mapping function F . This labelling is like a “bet” made by the *AL*. However, the *AL* should take the smallest “risk” to make this “bet”. For example, if F is an SVM classifier, the selected unlabelled sample should have a decision value very close to the assigned label and the SVM should be “safe” to assign the label to this sample. Secondly, the selected sample and the assigned label will be included into X to fit the mapping function F . Obviously, if the label is correctly assigned, the newly fitted F must be better or as good as previous because more correct information is used for the mapping. However, if the label is not correctly assigned, the newly fitted F could be worse than previous because more “noise” affects the mapping. 89

- 5.2 The working principle of an active miner. The algorithm starts with a feature with good discriminative ability to search for the DFS (This feature is called a seed as introduced in detail later). Then, the *AM* fits a mapping function, F , from this DFS to the label. After that, a query function Q is used to search for a new feature that can improve the accuracy of F . Here, we use three kinds of searching schemes as Q . The first scheme (A1) searches for the feature that aims to correct the errors made by the current DFS; in addition to the the routine of A1, the second scheme (A2) adds a step to to improve the prediction of “unreliable samples”; in addition to the the routine of A2, the third scheme (A3) adds a step to exclude “uncorrectable samples”. From this introduction, it is clear that our algorithm (including its three derivatives) are similar to active learning in its adaptability to “new knowledge”. The difference between active learning and our algorithm is the content of the “knowledge”. 90
- 5.3 The description of the *CS*. Here all the samples belong to two classes: squares with label “+1” and triangles with label “-1”. In the upper figure, we plot the samples that receive prediction values ≥ 0.8 or ≤ -0.8 ; In the figure below, we plot the samples that receive the prediction values between -0.8 to 0.8 together with the samples receiving wrong prediction. The *CS* is calculated to separate the samples in the under figure. 93

5.4	Four types of samples for a classification problem. Here all the “x” samples belong to -1 group and all the “+” samples belong to +1 group. The solid line between the -1 and +1 groups is the decision boundary $f(\mathbf{X}) = 0$. A: a <i>wrongly labelled sample</i> , that is, the decision boundary labels A wrongly; B: an <i>unreliable sample</i> , that is, the decision boundary labels B correctly. However, B has a decision value close to 0 (e.g., -0.1) and is very close to the decision boundary; C: an <i>uncorrectable sample</i> , that is, the decision boundary label C wrongly. Furthermore, the decision value for C is close to the wrong label +1 (e.g., +0.9); D: a <i>reliable sample</i> , that is, the decision boundary labels D. Furthermore, the decision value for D is close to the desired label -1 (e.g., -0.9).	96
5.5	Gene expression values for the two genes in the first DFS in the leukemia data set. (a) a plot includes only the training samples; (b) a plot includes all the training and testing samples.	103
5.6	The comparison of the averaged accuracies obtained by the SVM-RFE, AM-DFS (A1, A2, and A3), and Fisher’s linear discriminant in the leukemia data set in 50 random splits into 38 training samples and 34 testing samples.	105
5.7	The comparison of the averaged accuracies obtained by the SVM-RFE, AM-DFS (A1, A2, and A3), and Fisher’s linear discriminant in the colon data set in 50 random splits into 31 training samples and 31 testing samples.	107

5.8	The comparison of the averaged accuracies obtained by the SVM-RFE, AM-DFS (A1, A2, and A3), and Fisher's linear discriminant in the prostate data set in 50 random splits into 51 training samples and 51 testing samples.	115

List of Tables

4.1	Lymphoma gene importance ranking : 100 genes with the highest TSs, in the order of decreasing TSs (Gene ID is defined in [1]) . . .	54
4.2	2-gene combinations in the lymphoma data set that lead to 100% CV accuracy in the training set and their respective testing accuracy for the FNN. There are in total 174 such combinations. The complete table can be obtained in the Supplementary Information at http://www.ntu.edu.sg/home5/pg02317674	55
4.3	SRBCT gene importance ranking: 50 genes with the highest TSs, in the order of decreasing TSs.	68
4.4	Comparisons of results for the SRBCT data set obtained by different approaches. Here the results obtained in [2], [3], [4], and [5] are all biased, whereas our results for the FNN are unbiased.	69
4.5	General information for the GCM data set: the cancer types included, the number of training and testing samples for each type. .	78
4.6	The 10-fold cross-validation accuracy and the genes used by the 14 binary classifiers.	79
4.7	The confusion matrix for the 10-fold cross-validation results of the GCM data set. Rows delineate the predicted types and columns delineate the actual types.	80

4.8	The confusion matrix for the testing results of the GCM data set. Rows delineate the predicted types and columns delineate the actual types.	81
4.9	The K-means clustering result for the 100 selected genes in the lymphoma data set: the 20 clusters and the genes included in each cluster. The genes in each cluster may be highly correlated.	84
5.1	SVM-RFE for the leukemia data set with the top 500 features. The complete table can be obtained on the Supplementary Information webpage at http://www.ntu.edu.sg/home5/pg02317674	100
5.2	The DFSs obtained by the A2 for the leukemia data set with the top 500 features and the Seed 17. The runing time includes all the time spent on searching for DFSs (selecting seed, classification, computing correction scores, etc.). The complete table can be obtained on the webpage at www.ntu.edu.sg/home5 /pg02317674	101
5.3	The DFSs obtained by the A2 algorithm for the leukemia data with the top 500 features and their training and testing results. The runing time includes all the time spent on searching for DFSs (selecting seed, classification, computing correction scores, etc).	102
5.4	The comparison of the averaged accuracies obtained by the SVM-RFE, AM-DFS (A1, A2, and A3), and Fisher's linear discriminant in the leukemia data set in 50 random splits of the data set into 38 training samples and 34 testing samples. The numbers in the brackets are the corresponding standard deviations.	104
5.5	The testing accuracy for the 10-fold cross validation conducted in the colon data with our SVM-based AM-DFS. The numbers in brackets are the number of errors made in the corresponding fold. .	108

5.6	The training accuracy for the 10-fold cross validation conducted in the colon data using our SVM-based AM-DFS.	109
5.7	The validation accuracy for the 10-fold cross validation conducted in the colon data using our SVM-based AM-DFS.	109
5.8	The comparison of the errors in the colon data set with different approaches.	110
5.9	The comparison of the averaged accuracies obtained by the SVM-RFE, AM-DFS (A1, A2, and A3), and Fisher's linear discriminant in the colon data set in 50 random splits into 31 training samples and 31 testing samples. The numbers in the brackets are the corresponding standard deviations.	111
5.10	The testing accuracy for the 10-fold cross validation conducted in the prostate data with our SVM-based AM-DFS.	112
5.11	The validation accuracy for the 10-fold cross validation conducted in the prostate data with our SVM-based AM-DFS.	113
5.12	The training accuracy for the 10-fold cross validation conducted in the prostate data with our SVM based AM-DFS.	114
5.13	The comparison of the errors in the prostate data set with different approaches.	115
5.14	The comparison of computing times used for finding a 4-gene DFS both with the SVM-RFE and with the SVM-based AM-DFS in the colon data.	115

5.15 The comparison of the averaged accuracies obtained by the SVM-RFE, AM-DFS (A1, A2, and A3), and Fisher's linear discriminant in the prostate data set in 50 random splits into 51 training samples and 51 testing samples. The numbers in the brackets are the corresponding standard deviations.	116

Chapter 1

Introduction

1.1 Data Mining and Pattern Classification

We are now living in an “information era”. Tremendous amount of data are now stored in various types of databases. Furthermore, new data are created speedily and the total amount of the stored data are increasing exponentially [6]. Nowadays, a common problem we have is how to find some useful information (e.g., the present average height of female students in junior college in Singapore) from huge data resources. This is particularly obvious when the Internet becomes an easily accessible tool. Data mining is then an important technique for this task. In this dissertation, we will discuss one data mining problem, feature subset selection, in the context of pattern classification problems.

1.2 Feature Subset Selection

Providing a subset of features with good discriminative power (we call it discriminative feature set, or DFS) for a classifier has been an important topic in pattern recognition and machine learning research, especially when the dimensionality of the input space is high. Reducing the dimensionality of the input space will benefit a classifier because of the following aspects. (1) it can remove noise and irrelevant features from the classifier and therefore increase its accuracy; (2) it can reduce

1.2. Feature Subset Selection

---

the computation cost of the classifier.

Input dimensionality reduction approaches can be divided into two categories, i.e., feature extraction and feature selection. Feature extraction approaches transform the original input data into a reduced representation set of features such that they are expected to contain relevant information from the input data for performing a desired task. Popular feature extraction approaches (besides feature extraction, some of these methods could possibly be used for other purposes) include: principal components analysis (see, e.g., [7]), independent components analysis (see, e.g., [8]), partial least squares (see, e.g., [9]), etc. Feature selection approaches select a number of features from all features in the input space according to a selection criterion (SC), for example, t -statistic (see, e.g., [10,11]), f -statistic (see, e.g., [12,13]), correlation (see, e.g., [14,15]), separability-correlation measure [16], information gain (see, e.g., [17,18]), and so on. Although both feature extraction and feature selection reduce the dimensionality of a feature space, the former is more capable of capturing the characteristics of a feature space and is therefore more suitable for image processing [19], visualization [20], signal processing [8], and so on. In contrast, the latter is more capable of finding out proper features for a specific task, e.g., differentiating samples, and is therefore more suitable for classification problems.

Feature selection approaches can be further categorized into two models based on whether a classifier is involved. If the SC is classifier-independent, e.g., t -statistic, these approaches are called *filter*. If the approach selects a subset of features that directly optimize the performance of a classifier, these approaches are called *wrapper* [21]. A wrapper method usually yields a feature subset that leads to a model of better prediction accuracy and less redundancy than a filter method (see e.g., [17,22]). Be aware of this, a variety of wrapper approaches have been

1.2. Feature Subset Selection

---

proposed for various classifiers, e.g., neural networks [16, 23, 24], decision trees [25], linear discriminant analysis [26], and support vector machines (SVMs) [27, 28]. The major drawback for wrapper approaches is their much higher computational cost compared with filter methods.

During the process of feature subset selection, if a wrapper approach eliminates features from the original feature space repeatedly until a resulting subset is obtained, it is called a “backward” selection, on the contrary, if a wrapper adds features to a null subset repeatedly until a resulting subset is obtained, it is called a “forward” selection. Usually, a backward selection approach is better than its forward counterpart in terms of the quality of the selected feature subset [29]. Therefore, the backward selection, e.g., the widely used SVM-based recursive feature elimination (SVM-RFE) proposed in [27], becomes the mainstream for wrappers.

However, a major limitation of backward wrappers compared with forward wrappers is their high computational cost, especially when selecting feature subsets from a high-dimensional space. For example, using microarray gene expression data to differentiate tumors could be such a problem, which typically requires to select a few tens of features from thousands of candidates. In such cases, forward wrappers could be much faster than backward wrappers.

Considering the characteristics of both the filter and the wrapper, some hybrid approaches were proposed [30–33]. Such approaches usually first use a filter approach to select some candidate features from the original feature space and then use a forward or backward wrapper to obtain the final subset. These hybrid approaches reduce the computational cost through their filter step and consider the relationship among features through their wrapper step. Therefore, they usually achieve satisfactory results.

A good review of feature subset selection can be found in [34].

1.3. Biological Background

In this dissertation, we intend to develop powerful feature subset selection approaches. In general, our approaches can be categorized as hybrid feature selection. The searching strategy, i.e., how to use a classifier to search for good DFSs, plays a key role in feature selection approaches. In the beginning part of Chapter 3, we will review major searching schemes for feature selection. We will further apply the techniques we proposed to attack an important real world problem, i.e., cancer classification based on genomic data, or more specifically, microarray gene expression data.

1.3 Biological Background

Cancer is one of the most common and one of the most harmful diseases for human beings. For example, it is estimated that a total of 1,444,920 new cancer cases and 559,650 deaths occurred only in the United States in 2007, which means one in four deaths in the country is caused by cancer [35].

To fight against cancers, researchers have proposed a variety of diagnosis, detection, and screening approaches based on different types of clinical and histopathological information. For example, images, e.g., X-ray images, magnetic resonance images (MRI), electronic microscopy images, ultrasound images, computed tomography (CT) images, and so on, have been widely used in cancer diagnosis and have achieved good diagnosis results for some types of cancers [36–38]. Some other approaches test body fluids such as serum, plasma, sputum, saliva, etc., to detect cancers. Two commercialized examples approved by the U.S. Food and Drug Administration (FDA) are: (1) testing the level of prostate-specific antigen (a protein produced by cells of the prostate gland) to detect prostate cancer; and (2) measuring the level of CA125 (an intracellular protein) in serum to detect ovarian cancers [39,40]. Besides the above approaches and many other existing approaches,

1.3. Biological Background

---

a lot of new approaches and biomarkers have been proposed or found from time to time.

With the rapid advance of biological and medical sciences, researchers more and more rely on new techniques in cancer research. In the following several paragraphs, we briefly introduce some basic biological concepts closely related to this dissertation. Interested readers may refer to some biology textbooks, e.g. [41], for more information. After that, we introduce a newly developed technique, i.e., the microarray technology, which enables researchers to search for distinctions between different (sub)types of cancers at molecular level.

We start from cells that are fundamental working units of living organisms. A cell is a tiny but complex system that is invisible for naked eyes but is visible under a microscope. A human body contains approximately 6×10^{13} cells with different functionalities, such as skin cells, bone cells, muscle cells, etc. If we go further, a cell is formed by smaller molecules, that are usually not visible even through a electron microscope. A cell mainly consists of four types of molecules: proteins, DNAs (Deoxyribonucleic Acids), RNAs (Ribonucleic Acid), and some other small molecules. Proteins are formed by a string of amino acids that are “building blocks” of proteins. There are in total 20 different amino acids and the length of the string usually ranges from 100 to 1000 amino acids. Proteins are major functional units for a cell and perform varied cellular tasks, e.g., serving as structural components of a cell, sensing (changing shape as environmental factors changes), importing and exporting substances across the plasma membrane, working as enzymes, binding to a specific gene (see definition later) and turning it on and off, transferring signals/information among and within cells, etc.

DNA contains the genetic information that determines how, when, and where to produce each type of protein. The three dimensional structure of DNA is a

1.3. Biological Background

---

double helix, i.e., two long helical strands of nucleotides coiled around a common axis. Four different nucleotides, i.e., adenine (A), thymine (T), cytosine (C), and guanine (G), link end to end in a DNA strand. The nucleotides in the two strands has a *complementary* matching scheme: an A in one strand match a T in the other strand; and a C in one strand matches a G in the other. This complementary matching of the two strands is strong, e.g., if the two stands are separated, they will spontaneously zip back together in the right condition. Such *hybridization* is very important for detecting one strand using its complementary one. The linear sequence of nucleotides along a strand carries the genetic information of DNA. The portion that contains information in DNA is divided into discrete functional units (usually as long as 5000 to 100,000 nucleotides), *genes*. The protein synthesis process, i.e., how cells convert the genetic information in DNA into proteins, is illustrated in Fig. 1.1. Here if a gene in a DNA is used to construct protein, the gene is said to be *expressed*.

In 1995, microarray technology was invented, which make it possible to monitor expression profiles (which genes are expressed and the level of expression) of a large number of genes simultaneously [42]. This technology has become a widely accepted approach for biological and biomedical research and has been a great drive for new findings. As one of the major application areas for microarrays, cancer research benefits a lot from the gene expression data obtained from microarray experiments. For example, scientists can use microarrays to investigate:

- which genes are active in cancer cells compared with normal cells
- how cells respond to environmental stimuli (e.g., treatments or drugs)
- which genes are active in cancer cell reproduction (e.g., malignant tumor development).

1.3. Biological Background

Figure 1.1: Proteins synthesis process. In step 1 (i.e., transcription), a strand of the DNA double helix is used as a template by the RNA polymerase (a macromolecular machine) to synthesize a messenger RNA (mRNA). After that, the mRNA moves from the nucleus to the cytoplasm. In step 2 (i.e., translation), another macromolecule called ribosome binds to the mRNA and serves as a protein factory and synthesizes the protein according to the genetic information coded in the mRNA. This figure is taken from the Online Biology Textbook (www.emc.maricopa.edu/faculty/farabee/BIOBK/BioBookTOC.html) that is free to use for educational purposes.

1.4. Our Methodology

Figure 1.2 illustrates the process of using microarray chips to compare the expression profiles of the cells in two different conditions. Fig. 1.3 shows the process of quantifying the scanned images from microarray chips. A good recent review of microarray technology and its applications to cancer research can be found in [43].

1.4 Our Methodology

After microarray experiments, the next step is to analyze the obtained gene expression data, which is challenging due to the following three properties of these data. First, gene expression data are usually very high dimensional. The dimensionality usually ranges from several thousands to over ten thousands. Second, gene expression data sets usually contain relatively small numbers of samples, e.g., a few tens. If we treat this pattern recognition problem with supervised machine learning approaches, we need to deal with the shortage of training samples and high dimensional input features. Third, there are usually considerable noise in gene expression data sets.

To discover useful information from gene expression data, a lot of feature selection/extraction approaches and classification algorithms have been applied to this problem during the past decade. In Section 4.1, we will review the important algorithms and results in the literature. Although previous studies usually report results like this: the group of genes achieved $xx\%$ prediction accuracy, the following questions are still murky:

- Are these groups of genes the smallest gene sets who can achieve such accuracy? Is there still redundancy in these gene sets?
- Are these gene sets unique, i.e., are there any other gene sets that achieve similar performance?

1.4. Our Methodology

Figure 1.2: the process of using microarray chips to compare the expression profiles of the cells in two different conditions. First, some DNA fragments are attached at some fixed spots on a glass (or plastic, silicon) slide. Every single spot contains a huge number of identical DAN fragments and each of these fragments ideally should represent a specific gene. Since one chip usually contains thousands of spots, the expression profiles of these (different) genes could be measured in one experiment. Second, mRNAs from the cells in two different conditions (e.g., cells from a health body and their counterparts from a diseased body) are extracted and labeled with two different fluorescent dyes: a green dye for cells at condition 1 and a red dye for cells at condition 2. Third, liquid solution that contains mixed mRNAs from both conditions are incubated on the glass slide. The mRNAs then attach to their complementary sequences in the spots due to complementary matching. Finally, the glass slide is washed. The amount of samples bound to a spot can be measured by the level of fluorescence emitted when it is excited by a laser. If a spot contains more RNAs from the sample in condition 1, it will be green. If it contains more RNAs from the sample in condition 2, it will be red. If both are equal, a spot will be yellow. If neither are present, a spot appears black. Thus, from the color intensities of each spot, the relative gene expression levels in both samples can be estimated.

1.5. Major Contributions of the Dissertation

Figure 1.3: The quantifying process of microarray data. To obtain useable gene expression profiles, the raw data, i.e., the images files output by a microarray scanner, should undergo the following steps. First, the color intensity of each spot is measured and compared to the background using a software, which is called image quantization. Second, all the quantities that represent the same gene (no matter in the same array or in different arrays) have to be combined and re-scaled to make different arrays comparable. Finally, all the quantities will be put into a matrix, in which each row usually represents a gene and each column usually represents a condition or sample.

In our study, we aim to find the minimal discriminative gene sets (DGS) for high-dimensional cancer classification problems. First of all, we propose a partially exhaustive searching algorithm, which achieved high accuracy and significant redundancy reduction in our experiments. In addition, to overcome the difficulty of the partially exhaustive searching algorithm in finding good DGSs with more than five genes, we propose the active mining algorithm, which also achieved good results.

1.5 Major Contributions of the Dissertation

In this dissertation, my major contributions include the following points.

1. I proposed a novel partially exhaustive searching algorithm that is capable of finding very small DFS. We successfully applied this method to the cancer

1.6. Organization of the Thesis

---

classification problems using microarray gene expression data. In the gene expression data sets we tested, it is found that some gene combinations that contains only two or three genes are able to achieve perfect separation of different types of caners. As far as we know, these gene combinations are the smallest discriminative gene sets (DGSs) found in the similar studies, which help biologists focus on these genes and find the underlying relationship between the genes and the cancers.

2. In view of the major limitation of the partially exhaustive searching approach, i.e., its huge computing time, I proposed the second method called active mining discriminative feature set (AM-DFS). This method is similar with the pool based active learning (see, e.g., [44], and will be further introduced in Section 5.2) in its adaptive ability in the learning process. However, in the pool based active learning, a classifier adaptively obtains new samples to improve its accuracy. For the AM-DFS, a classifier adaptively obtains new features into the DFS to improve its accuracy. Because of this, the AM-DFS is able to find very good DFSs efficiently. Because the AM-DFS uses a forward selection scheme, it is much faster than the SVM-RFE proposed in [27], which is a widely-used and successful backward approach for feature subset selection. Moreover, because of its adaptive ability, the AM-DFS achieved comparable results with the SVM-RFE. We also applied the AM-DFS to cancer classification problems using gene expression data. These applications further substantiate its effectiveness.

1.6 Organization of the Thesis

The rest of the dissertation is organized as follow:

1.6. Organization of the Thesis

In Chapter 2, we introduce two classifiers, i.e., support vector machines (SVMs) and fuzzy neural networks (FNN). Our algorithms are based on these two classifiers. Therefore, we will use these two classifiers intensively in following chapters.

In Chapter 3, we first review the major searching schemes for feature selection. After that, we introduce and compare principal component analysis (PCA) and three feature importance ranking schemes, i.e., t -statistics, class separability measure, and Fish-ratios. According to this comparison, we will use some of these methods for primarily eliminating irrelevant features and also for seed selection later.

In Chapter 4, we propose a partially exhaustive searching algorithm and apply it to cancer classification using microarray gene expression data. We tested our algorithm in several well-known data sets and obtained some results that have not been reported before and may be interesting to biologists.

In Chapter 5, we propose another algorithm, i.e., active mining discriminative feature sets or AM-DFS, to overcome the huge computing time needed by the partially exhaustive searching when a DFS includes more than five features. The experiments in three microarray gene expression data sets prove the effectiveness and efficiency of this algorithm.

In Chapter 6, we summarize the dissertation, discuss the limitations of the proposed methods, and also recommend the future directions of this research.

Chapter 2

Classification Algorithms

In this dissertation, we mainly discuss support vector machines (SVMs) and fuzzy neural networks (FNNs) as classification algorithms. We choose the SVM because it is now regarded as one of the best classification algorithms and has a lot of successful applications [45–50]. Besides SVMs, we hope to choose another classification algorithm that is quite different with SVMs to test the feature subset selection approaches to be proposed later. Since some popular classification algorithms, e.g., radial basis function neural networks, use the kernel technique (the concept of kernel will be introduced later) similar with SVMs, we choose FNNs that is based on the fuzzy logic as another classification algorithm. To provide a solid ground for the approaches we are going to develop, these two algorithms are introduced in detail in this chapter.

2.1 Support Vector Machines

SVMs are a statistical learning algorithm pioneered by Vapnik [51–54]. The basic idea of SVMs is to find an optimal hyper-plane that can maximize the *margin* (a precise definition of margin will be given later) between two groups of samples. The vectors that are nearest to the optimal hyper-plane are called support vectors (vectors with a circle in Figure 2.1) and this algorithm is called support vector

2.1. Support Vector Machines

---

machine.

2.1.1 An Optimal Hyper-plane for Pattern Recognition Linearly Separable Patterns

Given l input vectors $\{\mathbf{x}_i \in R^n, i = 1, \dots, l\}$ that belong to two classes, with desired output $y_i \in \{-1, 1\}$, a hyper-plane can be defined as

$$\mathbf{w}^T \mathbf{x} + b = 0 \quad (2.1)$$

to separate two classes, i.e.,

$$\mathbf{w}^T \mathbf{x}_i + b \geq 0, \text{ if } y_i = +1, \forall i, \quad (2.2)$$

$$\mathbf{w}^T \mathbf{x}_i + b < 0, \text{ if } y_i = -1, \forall i, \quad (2.3)$$

then we say these patterns are *linearly separable*. Here $\mathbf{w}$ is weight vector and b is bias. By rescaling $\mathbf{w}$ and b properly, we can change the two equations above to:

$$\mathbf{w}^T \mathbf{x}_i + b \geq 1, \text{ if } y_i = +1, \forall i, \quad (2.4)$$

$$\mathbf{w}^T \mathbf{x}_i + b \leq -1, \text{ if } y_i = -1, \forall i, \quad (2.5)$$

Or

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1. \quad (2.6)$$

There are two parallel hyper-planes

$$H1: \mathbf{w}^T \mathbf{x} + b = 1, \quad (2.7)$$

2.1. Support Vector Machines

$$H2: \mathbf{w}^T \mathbf{x} + b = -1. \quad (2.8)$$

We define the distance ρ between $H1$ and $H2$ as the *margin* between the two classes (Fig.2.1a). According to the standard result of the distance between the origin and a hyper-plane, we can figure out that the distances between the origin and $H1$ and $H2$ are $|b - 1|/||\mathbf{w}||$ and $|b + 1|/||\mathbf{w}||$, respectively. Because $H1$ and $H2$ are parallel, the sum of these two distances, ρ , is

$$\rho = 2/||\mathbf{w}||. \quad (2.9)$$

Figure 2.1: An optimal hyper-plane for classification in a two dimensional case, for (a) linearly separable patterns and (b) linearly non-separable patterns.

Our objective is to maximize the margin between the two classes, i.e., to minimize $||\mathbf{w}||$. This objective is equivalent to minimizing the cost function:

$$\psi = \frac{1}{2}||\mathbf{w}||^2. \quad (2.10)$$

Then, this optimization problem subject to the constraint (2.6) can be solved using Lagrange multipliers. The Lagrangian function is

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

2.1. Support Vector Machines

---

where $\alpha_i, i = 1, 2, \dots, l$ are Lagrange multipliers. Differentiating this Lagrange function, we obtain

$$\frac{\partial L(\mathbf{w}, b, \alpha)}{\partial \mathbf{w}} = \mathbf{0}, \quad (2.12)$$

$$\frac{\partial L(\mathbf{w}, b, \alpha)}{\partial b} = 0. \quad (2.13)$$

Apply the Lagrangian function 2.11 to 2.12 and 2.13, respectively, we obtain

$$\mathbf{w} = \sum_{i=1}^l \alpha_i y_i \mathbf{x}_i, \quad (2.14)$$

and

$$\sum_{i=1}^l \alpha_i y_i = 0. \quad (2.15)$$

Assume $(\bar{\mathbf{w}}, \bar{b})$ is an optimal solution to this optimization problem, we take note that

$$\alpha_i [y_i (\bar{\mathbf{w}}^T \mathbf{x}_i + \bar{b}) - 1] = 0 \quad \text{for } i = 1, 2, \dots, l. \quad (2.16)$$

For the optimization problem defined in Eq. (2.10) with respect to the constraint of Eq. (2.6), it is possible to construct a *dual problem*. First, Eq. (2.11) can be expanded to

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2} \mathbf{w}^T \mathbf{w} - \sum_{i=1}^l \alpha_i y_i \mathbf{w}^T \mathbf{x}_i - b \sum_{i=1}^l \alpha_i y_i + \sum_{i=1}^l \alpha_i. \quad (2.17)$$

In view of the Eq. (2.15), the term $b \sum_{i=1}^l \alpha_i y_i$ will be equal to zero. Furthermore, from Eq.(2.14) we have

$$\mathbf{w}^T \mathbf{w} = \sum_{i=1}^l \alpha_i y_i \mathbf{w}^T \mathbf{x}_i = \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j. \quad (2.18)$$

2.1. Support Vector Machines

If we set the object function $L(\mathbf{w}, b, \alpha) = Q(\alpha)$, we can reach the following dual problem of the primal one:

$$\text{maximize: } Q(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j, \quad (2.19)$$

subject to:

$$\sum_{i=1}^l \alpha_i y_i = 0, \quad (2.20)$$

$$\alpha_i \geq 0. \quad (2.21)$$

This process of reforming the primal problem into the dual problem was originally proposed by Wolfe in [55] and is called Wolfe's Dual. Usually, this dual problem is solved numerically, e.g., through the sequential minimal optimization (SMO) method [56]. The procedure of the SMO is as follows. Each time gradually change only one coefficient and fix all the other coefficients to their initial values. Then set the value that leads to the maximum for the objective function to the coefficient being tuned. Repeat these steps to all the coefficients to be trained to get the final solution.

The optimal Lagrange multipliers $(\alpha_{o,i})$ are therefore obtained. Hence the optimal weight vector $(\mathbf{w}_o)$ can be obtained by substituting $\alpha_{o,i}$ into Eq.(2.14):

$$\mathbf{w}_o = \sum_{i=1}^l \alpha_{o,i} y_i \mathbf{x}_i. \quad (2.22)$$

The optimal bias b_o can be obtained by considering a support vector on $H1$:

$$b_o = 1 - \mathbf{w}_o^T \mathbf{x}^s. \quad (2.23)$$

Here $\mathbf{x}^s$ is a vector on the hyperplane $H1$. With $\mathbf{w}_o$ and b_o known, the optimal hyperplane that maximize the margin between the two classes is determined.

2.1. Support Vector Machines

Linearly Non-separable Patterns

If the vectors $\{\mathbf{x}_i \in R^n, i = 1, \dots, l\}$ cannot be linearly separated, we would like to slack the constraints described by equation (2.6). Here we introduce a group of slack variables, i.e., ξ_i , that measure the deviation of a sample from the ideal condition of pattern separability. In Fig.2.1(b), ξ_1 and ξ_2 are two examples of these slack variables.

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \quad (2.24)$$

$$\xi_i \geq 0. \quad (2.25)$$

In fact, ξ_i is the distance between the training example $\mathbf{x}_i$ and the optimal hyper-plane (Fig.2.1b). For $0 \leq \xi_i \leq 1$, $\mathbf{x}_i$ falls on the region between the two hyper-planes, i.e., $H1$ and $H2$, but on the correct side of the optimal hyper-plane. However, for $\xi_i > 1$, $\mathbf{x}_i$ falls on the wrong side of the optimal hyper-plane.

Since we hope the optimal hyper-plane can maximize the margin between the two classes and minimize the errors, we rewrite the cost function from equation (2.10) to:

$$\psi = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^l \xi_i. \quad (2.26)$$

where C is a positive factor. This cost function must satisfy the constraints of (2.24) and (2.25). There is also a dual problem:

$$\text{maximize: } Q(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j, \quad (2.27)$$

subject to:

2.1. Support Vector Machines

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

$$C \geq \alpha_i \geq 0. \quad (2.29)$$

From this dual problem, the optimal weight vector ($\mathbf{w}_o$) and the optimal Lagrange multipliers ($\alpha_{o,i}$) of the optimal hyper-plane can be obtained. They are the same as their counterparts in (2.22), except that the constraints change to (2.28) and (2.29).

In general, C controls the trade-off between the two goals of the binary SVM, i.e., to maximize the margin between the two classes and to separate the two classes well. When C is small, the margin between the two classes is large, but it may make more mistakes in training patterns. In other words, when C is large, the SVM is likely to make fewer mistakes in training patterns. However, a model with a small margin makes the network vulnerable for overfitting. Fig.2.2 depicts the functionality of the parameter C which has a relatively large impact on the performance of the SVM. Usually, it is determined experimentally for a given problem.

2.1.2 A Binary Nonlinear SVM Classifier

According to [57], if a nonlinear transformation that can map an input feature space into a new feature space whose dimensionality is higher than the original input space, the classification problem is more likely to be linear in the new space (To be more accurate, we want to indicate that a nonlinear transformation without increasing dimensionality may also produce linear separability in some cases. However, in general, a classification problem mapped into higher dimensional space nonlinearly is more likely to be linearly separable than in the original feature space).

2.1. Support Vector Machines

Figure 2.2: The influence of C to the performance of the classifier. (a) a classifier with a large C (small margin); (b) an overfitting classifier; (c) a classifier with a small C (large margin); (d) a classifier with a proper C .

In view of this theorem, a *nonlinear* SVM algorithm can perform such a transformation to map an input feature space to a new space with higher dimensionality (In fact, some other learning algorithms, such as radial basis function (RBF) neural networks, also do such a transformation for the same reason). After the transformation, the features in the new space are classified using the optimal hyperplane we constructed in the previous sections. Therefore, using this nonlinear SVM to do classification includes the following two steps:

1. Mapping the input space into a much higher dimensional space with a nonlinear function.
2. Performing classification in the new high dimensional space by constructing an optimal linear hyperplane that is able to maximize the margin between the two classes.

2.1. Support Vector Machines

Combining the transformation and the linear optimal hyper-plane, we can formulate the mathematical descriptions of a nonlinear SVM as follows.

We are supposed to find the optimal values of weight vector $\mathbf{w}$ and bias b such that they satisfy the constraint:

$$y_i(\mathbf{w}^T \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \quad (2.30)$$

$$\xi_i \geq 0. \quad (2.31)$$

where $\phi(\mathbf{x}_i)$ is the function mapping i -th pattern vector to a potentially much higher dimensional feature space. The weight vector $\mathbf{w}$ and the slack variables ξ_i should be selected to minimize the cost function:

$$\psi = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^l \xi_i. \quad (2.32)$$

This optimization problem is very similar to the problem we have dealt with using a linear optimal hyper-plane. The only difference is that the input vectors $\mathbf{x}_i$ have been replaced by $\phi(\mathbf{x}_i)$.

To solve this optimization problem, we follow the similar procedure as before. Through constructing the Lagrange function and differentiating it, we obtain a dual problem as below:

$$\text{maximize: } Q(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j), \quad (2.33)$$

subject to:

$$\sum_{i=1}^l \alpha_i y_i = 0, \quad (2.34)$$

$$C \geq \alpha_i \geq 0. \quad (2.35)$$

2.1. Support Vector Machines

---

where $K(\mathbf{x}_i, \mathbf{x}_j)$ is the kernel function:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j). \quad (2.36)$$

From this dual problem, the optimal weight vector ($\mathbf{w}_o$) and the optimal Lagrange multipliers ($\alpha_{o,j}$) of the optimal hyper-plane can be obtained:

$$\mathbf{w}_o = \sum_{i=1}^l \alpha_{o,i} y_i \mathbf{x}_i. \quad (2.37)$$

The optimal hyper-plane that discriminates different classes is:

$$\mathbf{w}_o^T \phi(\mathbf{x}) + b = 0. \quad (2.38)$$

One of the most commonly used kernel functions is the polynomial kernel:

$$K(\mathbf{x}, \mathbf{x}_i) = (\mathbf{x}^T \mathbf{x}_i + 1)^p, \quad (2.39)$$

where $p > 0$ is a constant specified by users. Another kind of widely used kernel function is the radial basis function:

$$K(\mathbf{x}, \mathbf{x}_i) = e^{(-\gamma \|\mathbf{x} - \mathbf{x}_i\|^2)}, \quad (2.40)$$

where $\gamma > 0$ is also a constant specified by users. According to its mathematic description, the structure of a binary nonlinear SVM is shown in Fig.2.3.

2.1.3 Multi SVM classifiers

It is very common that there are more than two classes in the data set. Therefore, binary SVMs are usually not capable of solving multi-class classification problems. To do it, we should divide a whole problem into a number of binary classification problems. Usually, there are two approaches [58]. One is the “one against all” scheme and the other is the “one against one” scheme.

2.1. Support Vector Machines

Figure 2.3: The structure of a binary nonlinear SVM. There are three layers in the SVM, i.e., the input layer, the hidden layer that includes kernel functions, and the output layer. The bias and the weights between the hidden layer and the output are determined by the learning algorithm.

For the “one against all” scheme, if there are N classes in the entire data set, then N independent binary classifiers are built. Each binary classifier is in charge of picking out one specific class from all the other classes. For a specific sample, all the N classifiers are used to make a prediction. The sample is categorized to the class that receives the strongest prediction. The prediction strength is measured by the result of the decision function. For the “one against one” scheme, there must be one (and only one) classifier taking charge of the classification between any two classes. Therefore, for a data set with S classes, $S(S - 1)/2$ binary classifiers are used. To reach the ultimate result, a voting scheme is used. For every sample, all the classifiers give their votes and therefore $S(S - 1)/2$ votes are obtained. The sample is then designated to the class obtaining the highest number of votes. If an sample obtains highest votes for more than one class, it is randomly designated to one of them. In terms of performance, there is still no conclusion on which scheme is better for combining binary classifiers [59].

2.2. A Fuzzy Neural Network

Figure 2.4: The structure of the FNN.

2.2 A Fuzzy Neural Network

In this section, we review the structure and the algorithm of the FNN proposed by Frayman and Wang [60, 61]. The structure of the FNN is shown in Fig.2.4. The network consists of four layers, i.e., the input layer, the input membership function (MF) layer, the rule layer, and the output layer.

Input data are either numerical or categorical. The input membership function layer converts numerical values to categorical values.

Each rule node is connected to each input membership function node and each output node. Each rule node performs a product of its inputs. The input membership functions act as fuzzy weights between the input layer and the rule layer. Links between the rule layer, the output layer, and the input membership functions are tuned during the learning process. In the output layer, each node receives inputs from all the rule nodes connected to this output node and produces the actual output of the network.

2.2. A Fuzzy Neural Network

2.2.1 FNN Initialization

Firstly, we create n nodes for the input layer and m nodes for the output layer, where n and m are the numbers of the input variables (attributes) and the output variables (classes), respectively. The rule layer is empty, i.e., there are initially no rules in the rule base.

Two equally spaced triangular membership functions as shown in Fig.2.5 are put along the operating range of each input variable. Piecewise-linear triangular membership function is chosen for computational efficiency [62].

Then we create the initial rule base layer using the following form for rule i :

$$\text{Rule } i: \text{ if } x_1 \text{ is } A_{x_1}^i, \dots, \text{ and } x_n \text{ is } A_{x_n}^i, \text{ then } \mathbf{y}^i = [0, 0, \dots, 1, \dots, 0]^T,$$

where $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ is the input vector and $\mathbf{y}^i = [y_1^i, y_2^i, \dots, y_m^i]^T$ is the output vector of rule i , respectively. In the initialization stage, if a training sample belongs to class t , then the initial value of $\mathbf{y}^i$ is $[0, 0, \dots, 1, \dots, 0]^T$, i.e., $y_t^i = 1$ and all other numbers in $\mathbf{y}^i$ are 0. A_q^i ($q = x_1, x_2, \dots, x_n$) is the membership function of the antecedent part of rule i for node q in the input layer [63].

The strength of i -th rule, i.e., the weight ω_i is calculated with a fuzzy AND using the product operator:

$$\omega_i = A_{x_1}^i(x_1) \times A_{x_2}^i(x_2) \times \dots \times A_{x_n}^i(x_n) \quad (2.41)$$

The output y_l of the fuzzy inference is obtained using the weighted average [64]:

$$y_l = \frac{\sum_i \omega_i \times y_l^i}{\sum_i \omega_i}, \quad (2.42)$$

2.2. A Fuzzy Neural Network

2.2.2 FNN Training

The network is trained using the following general learning rule [65,66].

$$y_l^i(k+1) = y_l^i(k) - \eta \frac{\partial \varepsilon_l}{\partial y_l^i} \quad (2.43)$$

where η is the learning rate and k is the iteration number. The objective is to minimize the error function:

$$\varepsilon_l = \frac{1}{2} \times (y_l - y_{dl})^2 \quad (2.44)$$

where y_l is the current output, and y_{dl} is the target output.

We tune the learning rate η to improve the speed of convergence, as well as the learning performance (accuracy). If η is a small value, the convergence of the FNN will be guaranteed despite that the convergent speed of the FNN could be very slow. If η is too large, the FNN may not be able to reach convergence and may become unstable. To obtain an appropriate value for η , we update it according to the following heuristic rules:

1. If the error undergoes five consecutive reductions, increase η by 5%.
2. If the error undergoes three consecutive combinations of one increase and one reduction, decrease η by 5%.
3. Otherwise, keep η unchanged.

Here, the first rule will be effective if the convergent speed is too slow; the second rule will be effective if the FNN is not stable. Thanks to this dynamical updating strategy, the initial value of η is usually not critical for the FNN as long as it is not too large (because of the guaranteed convergency). The learning error ε_l is reduced towards zero or a pre-specified small value $\varepsilon_{def} > 0$ as the iteration number k increases.

2.2. A Fuzzy Neural Network

Figure 2.5: The initial membership function. Initially, the FNN uses a membership function with two triangles. During the training process, the membership function will also be changed as described in rule base modification section.

2.2.3 Rule Base Modification

An additional membership function is added to each input at the point of the maximum output error, following Higgins and Goodman [62]. One vertex of the additional membership function is placed at the point of the maximum output error, and it must have the membership value unity; the other two vertices lie at the centers of the two neighboring regions, respectively, and they have membership values zero. As the output of the network is not a binary 0 or 1, but a number ranging from 0 to 1, we can speed up the convergence of the network substantially by eliminating the error whose deviation from the target value is the greatest.

We then evaluate the accuracy and the simplicity of the rules generated above. We use a weighting parameter between accuracy and simplicity, which is the compatibility grade (CG) of each fuzzy rule. CG of rule j is calculated by the product operator as:

$$\mu_j(x) = \mu_{j1}(x_1) \times \mu_{j2}(x_2) \times \cdots \times \mu_{jn}(x_n) \quad (2.45)$$

when the system provides correct classification result.

Each rule whose CG falls below a pre-defined threshold is deleted. Elimination of rule nodes is rule by rule, i.e., when a rule node is deleted, its associated input

2.2. A Fuzzy Neural Network

---

membership nodes and links are deleted as well. By adjusting the CG threshold, the user is able to specify the degree of rule base compactness. The size of the rule base can thus be kept minimal. If the classification accuracy of the FNN after the elimination of rule nodes is below the requirement, we will add another rule as described above; otherwise we stop the process.

This FNN combines the powerful features of initial fuzzy model self-generation [63], parameter optimization [65], and rule-base simplification to achieve a good performance.

2.2. A Fuzzy Neural Network

Figure 2.6: The flow chart of the learning algorithm used by the FNN.

Chapter 3

Searching Algorithms and Feature Importance Ranking

In Chapter 1, I introduced feature extraction and feature selection and related concepts, e.g., filter, wrapper, and searching algorithms such as forward selection and backward selection, etc. In this Chapter, I will further review major searching algorithms for feature selection including branch and bound, sequential forward selection, generalized sequential forward selection, sequential backward selection, generalized sequential backward selection, plus l -take away r selection, generalized plus l -take away r selection, floating search, and genetic algorithm. Interested readers may refer to Chapter 9 of [67] for more details. In Chapter 4 and Chapter 5, we will propose a partially exhaustive searching algorithm and an active mining algorithm, respectively. Both of them can be regarded as new searching algorithms.

In the later part of this chapter, we compare three different SC s: (1) the class-separability that is equivalent to the f -statistics; (2) Fisher-ratio; and (3) t -statistics. In addition, we also include the principal components analysis (PCA) in this comparison because it is also a commonly used dimensionality reduction approach. When we propose our feature selection algorithms later, we will choose a good one from the SC s compared here.

3.1. Searching Algorithms for Feature Selection

3.1 Searching Algorithms for Feature Selection

3.1.1 Branch and Bound

A branch and bound approach uses a tree structure to cover all possible feature subsets. Each node on the tree represents a specific feature subset. The search algorithm starts from the node that represents the set including all features. Then it moves to different nodes on the tree by removing features recursively and test the feature subsets represented by these nodes. For each node (subset) being searched, the selection criterion function J (In terms of functionality, J is similar with the SC used by filters) is computed. Only at the promising node that obtain good J value, the algorithm goes further, i.e., removes features from the current subset.

A branch and bound approach works on condition that J satisfies the following *monotonicity property*: for two subsets X and Y ,

$$X \subset Y \Rightarrow J(x) < J(Y). \quad (3.1)$$

This requirement limits the usability of branch and bound approaches because the monotonicity property does not hold for a lot of real world problems.

3.1.2 Sequential Forward Selection

Sequential forward selection (SFS) was first used to feature subset selection by Whitney [68]. SFS starts the search from an empty set. Each step, a new feature, ξ_i , is added into the current feature set, X , such that the new feature set $X + \xi_i$ maximize the selection criterion function J , i.e., for $\forall k \neq i$ and $\xi_k \notin X$, we have $J(X + \xi_i) \geq J(X + \xi_k)$. This adding feature process repeats again and again until a stop criterion is satisfied.

The major disadvantage of SFS is that it does not remove a feature once it is added into the feature set, which is all called “nesting effect”.

3.1. Searching Algorithms for Feature Selection

3.1.3 Generalized Sequential Forward Selection

Generalized sequential forward selection (GSFS) is similar with SFS. The difference between the two approaches is that GSFS adds a group of (e.g., r) features each time. Suppose we have n features in total in a data set and X already includes d features. Therefore, each time there are C_{n-d}^r possible sets that could be added into X . Similar with SFS, the set that maximizes J is added.

Compared with SFS, GSFS is usually more computationally expensive. However, the advantage is that it takes into account the “cooperation” among candidate features to some degree and usually obtain better results than SFS.

3.1.4 Sequential Backward Selection

Sequential backward selection (SBS) starts the search from the complete feature set. Each step, a feature, ξ_i , is deleted from the current feature set, X , such that the new feature set $X - \xi_i$ the selection criterion function J , i.e., for $\forall k \neq i$ and $\xi_k \notin X$, we have $J(X - \xi_i) \geq J(X - \xi_k)$. This deleting feature process repeats again and again until a stop criterion is satisfied.

SBS is usually more computational expensive than SFS, especially when the feature set is high dimensional and more features to be deleted. However, SBS usually can obtain better resulting feature sets than its SFS counterpart, as indicated in [69].

3.1.5 Generalized Sequential Backward Selection

Similar as the change from SFS to GSFS, generalized sequential backward selection (GSBS) starts the search from the complete feature set and delete a group of features each step. Similarly, GSBS usually outperforms SBS but also more computationally expensive than SFS.

3.1. Searching Algorithms for Feature Selection

3.1.6 Plus l - Take Away r Selection

To overcome nesting effect, plus l take away r selection (PTA(l, r)) was proposed. If $l > r$, this search scheme adds l features using SFS and then delete the worst r features using SBS. If $l < r$, this search scheme delete r worst features using SBS and then adds l features using SFS.

3.1.7 Generalized Plus l - Take Away r Selection

The generalized plus l - take away r selection (GPTA(l, r)) uses GSFS and GSBS to replace SFS or SBS used in the PTA(l, r).

3.1.8 Floating Search

Floating search method proposed in [70] includes sequential forward floating selection (SFFS) and sequential backward floating selection (SBFS) and is an extension of PTA(l, r). Here floating means l and r are allowed to vary at each stage. The procedure of SFFS is as follows.

1. Using SFS to obtain a feature set with 2 features
2. Add a new feature, c.g., ξ_i , using SFS
3. Find a feature, e.g., ξ_k , such that ξ_k will be deleted if SBS is applied here. If $\xi_i = \xi_k$, go to step 1; otherwise, delete it from the feature set
4. Continue deleting features as long as J increases and the resulting feature set includes more than 2 features; otherwise, go to step 1.

Floating search approaches usually can obtain comparable results with GPTA(l, r). The main advantage of floating search is it usually computationally cheaper compared with GPTA(l, r).

3.2. Data Sets

3.1.9 Genetic Algorithm

Using genetic algorithm (GA) to feature subset selection usually includes the following steps:

1. Generate initial population, i.e., generate a pool of candidate feature subsets
2. Evaluate the fitness of each subset in the pool
3. Repeat

Select best-ranking subset to reproduce

Breed new generation through *crossover* and *mutation* (feature exchange) and give birth to offspring subsets

Evaluate the individual fitnesses of the offspring subsets

Replace worst ranked part of population with offspring subsets

4. Until the terminating condition is satisfied

For high dimensional feature selection problems, GA is usually a possible option, which can lead to satisfactory solutions. However, GA is usually computationally expensive.

3.2 Data Sets

In the following sections, we will experimentally compare three different *SCs* and PCA using two gene expression data sets and two data sets from UCI machine learning repository [71].

3.2.1 The SRBCT Data Set

The small round blue cell tumor (SRBCT) data set [2] can be obtained from the website (<http://research.nhgri.nih.gov/microarray/Supplement/>). The entire data

3.2. Data Sets

---

set includes the expression data of 2308 genes. There are in total 63 training samples and 25 testing samples, five of the testing samples being not SRBCTs. The 63 training samples contain 23 Ewing family of tumors (EWS), 20 rhabdomyosarcoma (RMS), 12 neuroblastoma (NB), and 8 Burkitt lymphomas (BL). The 20 SRBCTs testing samples contain 6 EWS, 5 RMS, 6 NB, and 3 BL.

3.2.2 The Lymphoma Data Set

The lymphoma data set [72] can be obtained from the website (<http://lmpp.nih.gov/lymphoma>). In this data set, there are 42 samples derived from diffuse large B-cell lymphoma (DLBCL), 9 samples from follicular lymphoma (FL), and 11 samples from chronic lymphocytic lymphoma (CLL). The entire data set includes the expression data of 4026 genes. In this data set, a small part of data is missing. A k-nearest neighbor algorithm was applied to fill those missing values [73].

3.2.3 Wisconsin Diagnostic Breast Cancer Data Set

The Wisconsin Diagnostic Breast Cancer (WDBC) data set is obtained from the UCI machine learning repository [71]. The data set consists of 569 breast cases and each case is described by 30 features [74]. Among these cases, 357 of them are benign breast and 212 of them are malignant breast cancer.

3.2.4 Ionosphere Data Set

This ionosphere data set is also obtained from the UCI machine learning repository [71]. These radar data are categorized into: “Good” radar returns that show evidence of structure in the ionosphere; and “Bad” returns that show nothing. The data are described by 34 features. In total, there are 351 returns in this data set, in which 225 are “good” and the rest 126 are “bad”.

3.3. PCA and SCs

3.3 PCA and SCs

In this section, three *SCs* together with the *PCAs* are introduced in detail. After that, they are tested in microarray gene expression data sets.

3.3.1 Principal Component Analysis

One of the most widely used technique for dimensionality reduction in gene expression analysis is principal component analysis (PCA) (see, e.g., Chapter 8 of [75]). The basic idea of PCA is transforming the input space into a new space described by principal components (PCs). All the PCs are orthogonal to each other and are ordered according to the absolute values of their eigenvalues. The k -th PC is the eigenvector with the k -th largest eigenvalue. By leaving out the eigenvectors with small eigenvalues, the dimensionality of the input space is reduced. Because PCA chooses vectors with the largest eigenvalues, it covers the directions with the largest vector variations in the input space. In the directions determined by the vectors with small eigenvalues, the vector variations are very small. In a word, PCA intends to capture the most informative directions.

3.3.2 Class-Separability Analysis for Gene Selection

Another frequently used method for gene selection is to measure the class-separability (*CS*) [76]. *CS* of gene i is defined as:

$$CS_i = SB_i / SW_i, \quad (3.2)$$

where

$$SB_i = \sum_{i=1}^K (\bar{x}_{ik} - \bar{x}_i)^2, \quad (3.3)$$

$$SW_i = \sum_{k=1}^K \sum_{j \in C_k} (x_{ij} - \bar{x}_{ik})^2, \quad (3.4)$$

3.3. PCA and SCs

$$\bar{x}_{ik} = \sum_{j \in C_k} \bar{x}_{ij} / n_k \quad (3.5)$$

$$\bar{x}_i = \sum_{j=1}^n x_{ij} / n \quad (3.6)$$

Here SB_i is the sum of squares of between-class distances (the distances between samples of different classes). SW_i is the sum of squares of with-in class distances (the distances of samples within the same class). In the whole data set, there are K classes. C_k refers to class k that includes n_k samples. x_{ij} is the expression value of gene i in sample j . $\bar{x}_{ik}$ is the mean expression value in class k for gene i . n is the total number of samples. $\bar{x}_i$ is the general mean expression value for gene i . A CS is calculated for each gene. A larger CS indicates a larger ratio of the distances between different classes to the distances within one specific class. Therefore, CSs can be used to measure the capability of genes to separate different classes.

3.3.3 Fisher-Ratio for Gene Selection

Fisher ratio (see, e.g., [77]) is also a ratio of between-class distances to with-in class distances. If there are two classes in a data set, the Fisher ratio (F) for gene i is:

$$F_i = \frac{(\bar{x}_{i1} - \bar{x}_{i2})^2}{s_{i1}^2 + s_{i2}^2}. \quad (3.7)$$

where

$$s_{i1}^2 = \frac{1}{n_1} \sum_{j \in C_1} (x_{ij} - \bar{x}_{i1})^2 \quad (3.8)$$

$$s_{i2}^2 = \frac{1}{n_2} \sum_{j \in C_2} (x_{ij} - \bar{x}_{i2})^2. \quad (3.9)$$

If there are more than two classes in the data set,

3.3. PCA and SCs

$$F_i = \sum_{p=1}^K \sum_{q=1}^K \frac{(\bar{x}_{ip} - \bar{x}_{iq})^2}{K(K-1)(s_{ip}^2 + s_{iq}^2)}, \quad p \neq q. \quad (3.10)$$

where

$$s_{ik}^2 = \frac{1}{n_k} \sum_{j \in C_k} (x_{ij} - \bar{x}_{ik})^2. \quad (3.11)$$

The main differences between the CS and the F are as follows. (a) When extending F from two-class version (Eq. (3.7)) to multi-class version (Eq. (3.10)), the multi-class version averages all the F between any two different classes; however, CS can be directly calculated for multi-class data sets. (b) When calculating the between-class distance, the CS adds up all the distances from the mean of each class ($\bar{x}_{ik}$) to the overall mean ($\bar{x}_i$); however, F directly calculates the distances between the means of any two different classes and then adds them up.

3.3.4 A T -Test-Based Gene Selection Approach

T -test is a statistical method proposed by Welch [11]. It is used to measure how large the difference is between the distributions of two groups of samples. For a specific gene, if it shows larger distinctions between 2 groups, it is more important for the classification of the two groups. To find the genes that contribute most to the classification, t -test has been used in gene selection in recent years [78].

The TS of gene i is defined as follows [79].

$$TS_i = \left| \frac{\bar{x}_{c1} - \bar{x}_{c2}}{s_{pi} \sqrt{1/n_1 + 1/n_2}} \right| \quad (3.12)$$

where

$$\bar{x}_{c1} = \sum_{j \in C_1} \bar{x}_{ij} / n_1 \quad (3.13)$$

$$\bar{x}_{c2} = \sum_{k \in C_2} \bar{x}_{ik} / n_2 \quad (3.14)$$

3.4. Experimental Results

$$s_{pi}^2 = \frac{\sum_{j \in C_1} (x_{ij} - \bar{x}_{c1})^2 + \sum_{k \in C_2} (x_{ik} - \bar{x}_{c2})^2}{n_1 + n_2 - 2} \quad (3.15)$$

There are 2 classes, i.e., C_1 and C_2 , which include n_1 and n_2 samples, respectively. x_{ij} and x_{ik} are the expression values of gene i in C_1 and C_2 , respectively. $\bar{x}_{c1}$ and $\bar{x}_{c2}$ are the mean expression values of C_1 and C_2 . s_{pi} is the pooled standard deviation of gene i .

The major limitation of the above definition is that it is only able to measure the difference between two groups. When the number of classes is more than two, we need to modify the standard t -test. In this case, we use t -test to calculate the degree of difference between one specific class and the centroid of all the classes. Hence, the definition of TS for gene i can be described like this:

$$TS_i = \max\left\{\left|\frac{\bar{x}_{ik} - \bar{x}_i}{m_k s_i}\right|, k = 1, 2, \dots, K\right\} \quad (3.16)$$

$$s_i^2 = \frac{1}{n - K} \sum_k \sum_{j \in C_k} (x_{ij} - \bar{x}_{ik})^2 \quad (3.17)$$

$$m_k = \sqrt{1/n_k - 1/n} \quad (3.18)$$

Here $\max\{y_k, k = 1, 2, \dots, K\}$ is the maximum of all $\{y_k, k = 1, 2, \dots, K\}$. s_i is the pooled within-class standard deviation for gene i .

3.4 Experimental Results

We tested the three SC s and PCA using the SRBCT data and the lymphoma data. For each data set, we ranked the genes with the three SC s and PCA and then picked out the top 50 genes. After that, we input these genes into a SVM classifier and the results are shown in Fig. 3.1 and Fig. 3.2, respectively. For the SRBCT data, we follow its original partition, i.e., 63 samples for training and ranking genes; the rest 20 samples for testing. From Fig. 3.1, we found that all

3.5. Conclusions

---

the other three *SCs* outperformed PCA obviously. Furthermore, *TS* was slightly better than *CS* and Fisher ratio, especially when the number of genes was small.

For the lymphoma data, there are no ready-to-use partitions from its creator. Therefore, we randomly divided the data into two parts: 31 samples for training and ranking genes, the rest 31 samples for testing. In Fig. 3.2, we found again that all the other three *SCs* outperformed PCA obviously. Furthermore, *TS* and *CS* are slightly better than Fisher ratio, especially when 5 to 25 genes are used for classification.

Besides gene expression data, we also tested the three *SCs* and PCA using the WDBC data and the Ionosphere data from the UCI machine learning repository [71]. Similarly, we used a SVM classifier to do these tests. For the WDBC data, we randomly divided the data set into two parts: 341 sample for training and ranking features; the rest 228 sample for testing. The results is shown in Fig. 3.3. From this result, we found that except that PCA obtained an inferior result, the other three *SCs* obtained comparable results.

For the Ionosphere data, we randomly divided the data set into two parts: 177 sample for training and ranking features; the rest 174 sample for testing. The results is shown in Fig. 3.4. From this result, we found that all the other three *SCs* outperformed PCA obviously. Furthermore, *TS* and *CS* are slightly better than Fisher ratio.

3.5 Conclusions

In this chapter, we tested the performance of ranking feature for PCA and three other *SCs*, i.e., the class separability analysis, the *t*-test, and the Fisher ratio. In general, PCA is inferior to the other three *SCs* in terms of accuracy in all the

3.5. Conclusions

Figure 3.1: Comparison of classification results in the SRBCT data set using the 50 genes selected by the *CS*, the *TS*, Fisher ratio, and PCA, respectively.

Figure 3.2: Same as Fig. 3.1, for the lymphoma data.

3.5. Conclusions

Figure 3.3: Comparison of classification results in the WDBC data set using the 30 features ranked by the *CS*, the *TS*, Fisher ratio, and PCA, respectively.

Figure 3.4: Comparison of classification results in the Ionosphere data set using the 34 features ranked by the *CS*, the *TS*, Fisher ratio, and PCA, respectively.

3.5. Conclusions

---

four data sets. In fact, this is understandable because PCA does not use the label information for dimensionality reduction. Furthermore, the TS and the SC are slightly better than the Fisher ratio in these tests. Therefore, we will use TS or SC in the following chapters when a good SC is needed in our approaches.

Chapter 4

A Partially Exhaustive Searching Approach

4.1 Introduction

As introduced in Sections 1.3 and 1.4, gene expression profiles obtained from microarray experiments shines some light on cancer classification problems. In this context, different kinds of machine learning and statistical methods such as artificial neural network [2], evolutionary algorithms [4], nearest shrunken centroids [3], etc., have been used to analyze gene expression data [58]. In [27], a support vector machine based recursive feature elimination method (SVM-RFE) was proposed. Although this method takes advantages from the classifier, i.e., the SVM, it is a greedy backward searching scheme that requires a large amount of computing time. In [17], a method called *Markov blanket* was used to reduce the redundancy in DGSs. In [30], Wang *et al.* used clustering methods to identify the redundancy in DGSs and then reduce the redundancy by “collapsing dense clusters”.

In 2003, Tibshirani *et al.* [80] successfully classified lymphoma data set [1] with only 48 genes by using a statistical method called nearest shrunken centroids with an accuracy of 100%. For the SRBCT data, Khan *et al.* [2] classified all 20 testing samples with 96 genes. They used a very simple two-layered linear neural network. In 2002, Tibshirani *et al.* [3] applied nearest shrunken centroids to the SRBCT

4.1. Introduction

---

data set. They obtained 100% accuracy with 43 genes. For the method of nearest shrunken centroids, it categorizes each sample to the class whose centroid is nearest to the sample. The difference between standard nearest centroids and nearest shrunken centroids is that the latter uses only some important genes rather than all the genes to calculate the centroids. In 2003, Deutsch [4] reduced the number of genes required to correctly classify the four cancer sub-types in the SRBCT data set to 12 genes. At the same year, Lee and Lee [5] also obtained 100% accuracy in this data set with a SVM classifier and the separability based gene importance ranking [81]. They used at least 20 genes to obtain this result. At the same time, they generated 3 principal components (PCs) from the 20 top genes. Their SVM can also obtain 100% accuracy with these 3 principal components. In a liver cancer analysis, Chen *et al.* [82] used 3180 genes (represented by 3964 cDNA) to classify HCC and the nontumor samples. (We will describe this liver cancer data set in detail in Section 4.3.3, where we first use this data set to test our algorithm.)

In this chapter, we propose a simple yet very effective method that leads to accurate cancer classification using expressions of only very few genes. This method exhaustively searches good discriminative gene sets (DGSs) among some genes with high importance ranking.

This chapter is organized as follows. We first introduce our procedure to find the minimum gene combinations. After that, we test our method in the lymphoma data, the SRBCT data (these two data sets are described in Section 3.2 where they are used to compare the performance of feature importance ranking schemes), the liver cancer data, and the GCM data (the liver cancer data and the GCM data will be described in Section 4.3.3 and Section 4.3.4, respectively, where we first use these data sets to test our algorithm). The numerical results of four data sets demonstrate the effectiveness of our approach. In the final part, we discuss the

4.2. Method

---

results and related findings.

4.2 Method

The proposed method comprises of two steps. In step 1, we rank all genes in the training data set using a scoring scheme. Then we retain the genes with high scores. In step 2, we test the classification capability of all simple combinations among the genes selected in step 1 using a good classifier. We note that this chapter proposes neither new feature ranking measures nor new classifiers. We shall describe two mechanisms for each step.

4.2.1 Step 1: Gene Importance Ranking

In step 1, we compute the importance ranking of each gene using the t -test introduced in Section 3.3.4 or the class separability introduced in Section 3.3.2. We then retain only the most important genes for step 2.

4.2.2 Step 2: Finding the Minimum Gene Subset

After selecting some top genes in the importance ranking list, we attempt to classify the data set with only one gene. We input each selected gene into a classifier. If no good accuracy is obtained, we go on classifying the data set with all the possible 2-gene combinations within the selected genes. If still no good accuracy is obtained, we repeat this procedure with all the 3-gene combinations and so on, until we obtain a good accuracy.

The number of top genes selected for forming combinations is experimentally determined with the following scheme. First of all, we select a number of top genes (e.g., the top 50 or 100 genes) and test all the combinations (e.g., 2-gene combinations) among them. After that, we add a new batch (10 or 20) of genes into the previous group and then test the new combinations brought by the newly

4.2. Method

---

added genes. If good combinations (i.e., combinations that lead to good accuracy) are found after the inclusion of the new batch of genes, we continue to add another batch of genes. If no new good combinations are found after adding the new batches of genes, we stop adding genes. The objective of adding new genes is to find (more) combinations that lead to high classification accuracy. For example, among the top 50 genes, a gene (e.g., the gene ranked 8) did not organize a good combination. When the 10 genes ranked from 51 to 60 are added, the gene may organize a good combination (e.g., gene combinations (8, 55)). However, adding new genes does not increase the size of combinations (the combination (8, 55) is still a two-gene combination). In this dissertation, we used the following two classifiers to test gene combinations.

Fuzzy Neural Network (FNN)

We apply an FNN proposed earlier in [83, 84]. This FNN combines the features of initial fuzzy model self-generation, partition validation, parameter optimization and rule-base simplification. A detailed introduction to this FNN can be found in Section 3.3.2.

Support Vector Machines (SVMs)

In addition to the FNN, we also used nonlinear SVMs (introduced in Section 2.1.2). The kernel function used in the SVMs is the radial basis function kernel described in Eq. (2.40). The parameters (C and γ) of SVMs are defined in Eq. (2.35) and Eq. (2.40), respectively.

For both the FNN and the SVMs, we carried out 10-fold cross-validation (CV) in the training data set to tune their parameters. The FNN has one parameter, i.e., the learning rate (η) that needs to be tuned. At the beginning of the tuning process, we assign an initial value η_0 to η (e.g., 0.1). After that, we build up

4.3. Results

---

an FNN and then test the FNN using 10-fold CV in the training data set. We subsequently adjust η a little and build then cross-validate the FNN again. We select the η value that leads to the smallest CV error. In Fig.4.1, we show the test accuracy achieved in the lymphoma data set using three different values for η (0.1, 0.75, and 0.05). From this figure, we found that the FNN obtained better results when $\eta = 0.05$ than $\eta = 0.75$ or $\eta = 0.1$.

For the SVM, the two parameters (C and γ) are selected similarly, i.e., different combinations of C and γ are tested in the training data using 10-fold CV and the combination that leads to the smallest CV error is selected as the parameter for the SVM. In Fig. 4.2, we show the test accuracy achieved in the lymphoma data set using three different values for γ (0.1, 0.01, and 0.001) and $C = 10$. From this figure, we found that the SVM obtained better results when $\gamma = 0.001$ than $\gamma = 0.1$ and $\gamma = 0.01$. In Fig. 4.3, we show the test accuracy achieved in the lymphoma data set using three different values for C (1, 10, and 100) and $\gamma = 0.01$. From this figure, we found that the SVM obtained better results when $C = 100$ than $C = 1$ and $C = 10$.

4.3 Results

4.3.1 Lymphoma Data

The lymphoma data set is described in detail in Section 3.2.2. In the first step, we randomly divided the 62 samples into 2 parts, 31 samples for training, 31 samples for testing. We ranked all the 4026 genes according to their t-scores (TSs) in the training data set. Then we selected the 100 genes with the highest TSs shown in Table 4.1. In this dissertation, every gene is labelled after its importance rank. For example, gene 4 means the gene ranked 4 in Table 4.1. Through its ID in the microarray (for example GENE1622X), the real name of each gene can be found

4.3. Results

Figure 4.1: The test accuracy achieved in the lymphoma data set using different values for η (0.1, 0.075, and 0.05) in the FNN.

Figure 4.2: The test accuracy achieved in the lymphoma data set using different values for γ (0.1, 0.01, and 0.001) and $C = 10$ in the SVM.

4.3. Results

Figure 4.3: The test accuracy achieved in the lymphoma data set using different values for C (1, 10, and 100) and $\gamma = 0.01$ in the SVM.

on the web page of the lymphoma data set.

We applied our FNN to classify the lymphoma microarray data set. At first, we added the selected 100 genes one by one to the network according to their TS ranks starting with the gene ranked 1 in Table 4.1. That is, we first used only a single gene that is ranked 1 as the input to the network. We trained the network with the training data set, and subsequently tested the network with the test data set. We repeated this process with the first 2 genes in Table 4.1, then 3 genes, and so on. We found that the FNN performed very well: it reached 100% 10-fold CV accuracy for the training data with only the first 8 genes in Table 4.1. And its corresponding testing accuracy was 96.8%.

The excellent performance of our FNN motivated us to search for the smallest gene subsets that can ensure highly accurate classification for the entire data set. We first attempted to classify the data set using only 1 gene. We fed each of the 4026 genes in the lymphoma data set into our FNN. The best 10-fold CV accuracy was 90.32% and the best testing accuracy was 80.65%. Second, we tested

4.3. Results

---

all possible combinations of 2 genes within the 100 genes in Table 4.1. Figure 4.5 shows the CV procedure used here. This procedure totally excluded the testing samples from the classifier building process and hence excluded the influence of bias. Other CVs conducted in this dissertation also used similar scheme. To our pleasant surprise, among all 4950 such possible 2-gene combinations, the CV accuracy for the training data reached 100% for the FNN in 174 combinations. The corresponding testing accuracies for these combinations varied from 80.6% (6 errors in 31 testing samples) to 100%. The averaged accuracy was 93.85%. Table 4.2 shows the detailed results.

Since only 2 genes were required for classification, it became possible for us to visualize the gene profiles with respect to the distinct lymphoma sub-types. Here we picked out some combinations that also reached 100% testing accuracy for visualization. We take note that these combinations only represent some best performed combinations we found, which are used for visually presenting our findings and related discussions. However, the performance of the classifier should only be estimated by the averaged accuracy of all the combinations in Table 4.2, i.e. 93.85%.

Figure 4.6 shows four plots of 2-gene combinations (78, 52), (78, 8), (4, 87), (4, 40). In each of the 4 plots, the clusters of DLBCL, CLL, FL are very clear and the boundaries can be easily drawn. Figure 4.7 shows the corresponding expression profiles of the four combinations.

In recent years, the method of objectively evaluating classifiers in the context of gene expression based cancer classification has been discussed in the literature [69,85–88]. A *double cross validation* or *nested cross validation* scheme was applied in [86–88]. To examine our results more completely, we also applied the double CV to the lymphoma data set. Fig.4.9 is the flow chart of the double CV algorithm that

4.3. Results

---

we used. Compared to the double CV in [86], the procedure we used is different in the following two aspects.

1. Besides the CV in the outer loop, we also conducted a CV in the inner loop to evaluate the classifier. However, in [86], Freyhult et al only used 1 fold of the samples to test the classifiers in the inner loop. Considering the small number of samples in an inner fold, e.g. only 5 or 6 samples for the lymphoma data set, our scheme is more likely to find out promising gene combinations.
2. Our method used all the inner CV data for gene ranking, and greatly reduced the computing time. Otherwise, if we wanted to totally exclude the testing samples from gene ranking in the inner CV, we had to rank genes and search for gene combinations for each fold. The computing time would be nearly 10 times of our scheme for a 10-fold inner CV. Although our inner CV accuracy included the “selection bias” as indicated in [69], we found such inner results were accurate enough for picking out promising gene combinations as shown in the following results. More importantly, such scheme would never change the independence of the testing samples in the outer CV.

For the lymphoma data set, we used an SVM as the classifier and searched for gene combinations among the top 100 genes (TSs). The double CV achieved 93.55% accuracy, i.e. 4 errors in 62 samples. To facilitate easy verification of our double CV process and results, we include all the results on here. Figure 4.10 describes how the data set was randomly divided into 10 folds. It also includes the actual and the predicted labels for each sample. Figure 4.12 contains the winmax matrix for each fold. In addition, it also indicates the gene combination that was randomly picked out.

However, simply combining genes with high ranks does not ensure high classification accuracy. The cooperation of genes is also of great importance. To illustrate

4.3. Results

---

this point, we plot gene combinations (8, 40) and (4, 78) versus the cancer types in Fig.4.8. Figure 4.6c and Fig.4.8b both show that gene 4 is very capable of classifying DLBCL, because DLBCL can be picked out using the X-axis (gene 4) value alone. Similarly, Figs 4.6d and 4.8a show that gene 40 has good capacity to classify CLL. Through the cooperation of gene 4 and gene 40, DLBCL, FL and CLL are totally classified (Fig.4.6d). However, if gene 40 works with a gene that is not good at separating FL from CLL, for example gene 8 (Fig.4.8a), the cooperation between these two genes will be poor and the accuracy will be low (only 63.675%). The poor cooperation between gene 4 and gene 78 (both with high ranks) shown in Fig.4.8b also substantiates the importance of cooperation between genes.

Furthermore, the expression profile in Fig.4.7 can also support this idea of appropriate gene combinations. In Fig.4.7, the expression value of gene 4 shows obvious difference between DLBCL and the other two types, i.e., FL and CLL. There is a clear boundary in the figure. This difference tells us that gene 4 is good at classifying DLBCL. Similarly, it can be found that gene 40 has a good capability of classifying CLL. Gene 4 and gene 40 are able to jointly classify the three lymphoma types with 100% accuracy.

4.3.2 SRBCT Data

The SRBCT data set is described in detail in Section 3.2.1. In the first step, we ranked the entire 2308 genes according to their TSs [3,89] in the training data set. Then we picked out the 50 genes with the highest TSs as shown in Table 4.3. We applied our FNN as the classifier to the SRBCT microarray data set.

We used the expression data of the gene ranked 1 in Table 4.3 to train and then test the FNN. We repeated this process with the top 2 genes in Table 4.3, then top 3 genes, and so on. The testing error decreased to 0 when the top 16 genes were input into the FNN.

4.3. Results

Table 4.1: Lymphoma gene importance ranking : 100 genes with the highest TSs, in the order of decreasing TSs (Gene ID is defined in [1])

Rank	ID	Gene Description
1	GENE653X	(Lactate dehydrogenase A; Clone=686889)
2	GENE651X	*tubulin-beta; Clone=325880
3	GENE698X	*BUB1=putative mitotic checkpoint protein ser/thr kinase; Clone=1341540
4	GENE1622X	*CD63 antigen (melanoma 1 antigen); Clone=769861
5	GENE2244X	*Unknown UG Hs.15458 ESTs; Clone=1354083
6	GENE639X	*hepatoma-derived growth factor; Clone=700360
7	GENE542X	*actin=cytoskeletal gamma-actin; Clone=1240822
8	GENE537X	(B-actin,1099-1372; Clone=143)
9	GENE735X	*RAD51=Recombination/repair strand exchange protein=Replication protein A; Clone=1059434
10	GENE642X	*nm23-H1=NDP kinase A=Nucleoside diphosphate kinase A; Clone=176482
11	GENE771X	*Thioredoxin reductase; Clone=700768
12	GENE654X	(dystrobrevin B DTN-B2=dystrophin-associated protein A0; Clone=1303618)
13	GENE1636X	*Fibronectin 1; Clone=139009
14	GENE708X	*Ki67 (long type); Clone=100
15	GENE655X	(GRSF-1=cytoplasmic G-rich mRNA sequence binding factor; Clone=365837)
16	GENE728X	(ribonucleotide-diphosphate reductase M1 subunit; Clone=1352103)
17	GENE563X	(Similar to proteasome subunit p112; Clone=1371333)
18	GENE622X	(CD73=5' nucleotidase; Clone=1341245)
19	GENE703X	*pLK=homologue of Drosophila polo serine/threonine kinase; Clone=1352275
20	GENE769X	*14-3-3 epsilon; Clone=266106
...	...	...
...	...	...
90	GENE1730X	(sgk=putative serine/threonine protein kinase transcriptionally modified during anisotonic and isotonic alterations of cell volume; Clone=1288868)
91	GENE1647X	*cathepsin B; Clone=261517
92	GENE1702X	*Mig=Hmig=chemokine targeting T cells; Clone=503617
93	GENE2224X	*Unknown UG Hs.32615 ESTs; Clone=1353075
94	GENE767X	(Bleomycin hydrolase; Clone=1357250)
95	GENE2397X	(Unknown UG Hs.124299 ESTs; Clone=1371968)
96	GENE506X	(putative oral tumor suppressor protein (doc-1); Clone=1288538)
97	GENE2084X	*Unknown UG Hs.119415 ESTs; Clone=1369392
98	GENE894X	(Similar to Cyclin B2; Clone=416184)
99	GENE624X	*RPA = replication protein A; Clone=1186177
100	GENE646X	*nm23-H2=NDP kinase B=Nucleoside diphosphate kinase B; Clone=1352394

4.3. Results

Table 4.2: 2-gene combinations in the lymphoma data set that lead to 100% CV accuracy in the training set and their respective testing accuracy for the FNN. There are in total 174 such combinations. The complete table can be obtained in the Supplementary Information at <http://www.ntu.edu.sg/home5/pg02317674>.

Gene 1	Gene 2	10-fold CV error in Training	Number of Mistakes in Testing
1	62	0	3
1	78	0	1
2	4	0	1
2	24	0	1
2	37	0	1
2	78	0	0
2	83	0	1
2	89	0	2
2	92	0	1
3	24	0	4
3	31	0	4
3	37	0	3
3	78	0	1
3	89	0	3
4	5	0	1
...	...	...	...
...	...	...	...
78	84	0	1
78	86	0	2
78	88	0	2
78	94	0	2
78	99	0	2
79	92	0	3
84	92	0	4
86	92	0	3
87	92	0	1
92	97	0	6
Averaged Number of Errors			1.908
Averaged Accuracy			93.85%

4.3. Results

Figure 4.4: The result achieved by the FNN in the lymphoma data set when the top ranked 50 genes were input one by one in the decreasing order of their TSs.

To further reduce the number of genes required for accurate classification of the 4 cancer sub-types, we tested all possible combinations of 1 gene and 2 genes within the 50 selected genes. None of them can lead to 100% CV accuracy for the training data.

Then we tested all possible combinations of 3 genes within the 50 selected genes. Among all 19600 such 3-gene combinations, we found the 10-fold CV accuracy for the training data reached 100% with the combination of Gene 5, Gene 12 and Gene 30. The testing accuracy of this combination is 95% (1 error for the 20 testing samples). Because this combination only contains 3 genes, it is still possible for us to visualize it. Figure 4.13 shows two different views of the 3-D plot of this 3-gene combination. In the view of Fig.4.13a, RMS and NB are well separated from other types. In the view of Fig.4.13b, BL and EWS are well separated with a clear boundary. From these two views, it is clear that the four SRBCT sub-types are separated from one another.

Except (5, 12, 30), no other gene combinations obtained 100% CV accuracy for

4.3. Results

Figure 4.5: The procedure of the cross validation (CV) used in this dissertation. Here the classifier building and the parameter tuning are carried out during the CV using only the training samples. The testing samples are totally independent from the CV, hence the testing results obtained are unbiased.

4.3. Results

Figure 4.6: Gene expression levels of 2-gene combinations that perfectly separate the 3 types of lymphoma sub-types, i.e., DLBCL, FL, and CLL: (a) (78, 52), (b) (78, 8), (c) (4, 87), (d) (4, 40). Here the genes are labelled according to their t-score (TS) ranks, for example, the two genes in (a) are ranked 78 and 52 according to their TSs. A total of 27 such 2-gene combinations are found by the FNN, which achieved 100% CV accuracy in the training process and also achieved 100% testing accuracy. These combinations are the best performed ones in Table 4.2. However, the performance of the classifier should only be estimated by the averaged accuracy of all the combinations in Table 4.2.

4.3. Results

Figure 4.7: The expression profiles of the 2-gene combinations in lymphoma data set shown in Fig.4.6: (a) (78, 52), (b) (78, 8), (c) (4, 87), (d) (4, 40). Here all the genes are labelled according to their TS ranks. All four combinations can separate DLBCL, FL, and CLL. For example, in (d), gene 4 draws a boundary between DLBCL and other types; gene 40 draws a boundary between CLL and other types. Therefore, Gene 4 and Gene 40 jointly separate the three types.

Figure 4.8: Gene expression levels of 2-gene combinations that do not separate the 3 lymphoma sub-types: (a) (8, 40), (b) (4, 78). Although these genes have high TS ranks, their combinations cannot separate the 3 subtypes because of poor cooperation.

4.3. Results

Figure 4.9: The procedures of the double cross validation (CV). In the flow, there are two loops, i.e. the inner loop and the outer loop. In the inner loop, genes are firstly ranked according to their TSSs. After that, we use all the 2-gene combinations to build classifiers and do internal 10-fold CV. Then, we use the classifier that achieves the best CV accuracy in the internal loop to classify the testing samples in the outer loop. Repeat the process until all the samples are cross validated in the outer loop. Because the testing samples in the outer loop are totally isolated from the gene ranking and classifier building process conducted in the internal loop, the CV result obtained in the outloop are unbiased.

4.3. Results

fold 1						
44	13	25	2	48	59	62
2	1	1	1	2	3	3
2	1	1	1	2	3	3
fold 2						
51	10	52	23	27	34	40
2	1	3	1	1	1	1
2	1	3	1	1	1	1
fold 3						
28	55	61	32	17	53	
1	3	3	1	1	3	
1	3	2	1	1	3	
fold 4						
49	1	22	20	7	8	
2	1	1	1	1	1	
2	2	1	1	1	1	
fold 5						
47	12	46	29	58	33	
2	1	2	1	3	1	
2	1	2	1	3	1	
fold 6						
43	6	14	45	60	30	
2	1	1	2	3	1	
1	1	1	1	3	1	
fold 7						
56	5	37	11	24	38	
3	1	1	1	1	1	
3	1	1	1	1	1	
fold 8						
18	31	26	15	35	16	
1	1	1	1	1	1	
1	1	1	1	1	1	
fold 9						
3	19	36	41	50	42	
1	1	1	1	2	1	
1	1	1	1	2	1	
fold 10						
57	21	54	9	39	4	
3	1	3	1	1	1	
3	1	3	1	1	1	

Figure 4.10: Same as Fig.4.16, for the lymphoma data set.

4.3. Results

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
1 6 0	1 7 0	1 8 0	1 22 0	1 6 0
1 11 0	1 9 0	1 21 0	1 27 0	1 34 0
1 28 0	1 10 0	1 41 0	1 32 0	1 35 0
1 30 0	1 21 0	1 48 0	1 64 0	1 37 0
1 35 0	1 23 0	1 63 0	1 69 0	1 41 0
1 37 0	1 29 0	1 67 0	1 75 0	1 59 0
1 55 0	1 35 0	1 70 0	1 95 0	1 64 0
1 68 0	1 41 0	1 80 0	1 96 0	1 74 0
1 76 0	1 51 0	1 90 0	2 10 0	1 91 0
1 95 0	1 55 0	3 8 0	2 22 0	1 96 0
3 30 0	1 67 0	3 28 0	2 27 0	4 34 0
3 35 0	1 69 0	3 41 0	2 32 0	4 35 0
3 37 0	1 73 0	3 48 0	2 43 0	4 41 0
3 55 0	1 78 0	3 63 0	2 64 0	4 59 0
3 68 0	7 8 0	3 67 0	2 69 0	4 91 0
3 76 0	7 12 0	3 70 0	2 75 0	4 96 0
6 39 0	7 18 0	3 80 0	2 95 0	5 18 0
11 20 0	7 29 0	4 78 0	3 10 0	5 34 0
11 39 0	7 30 0	5 40 0	3 64 0	5 42 0
11 60 0	7 38 0	5 64 0	3 95 0	6 42 0
11 74 0	7 51 0	5 85 0	6 22 0	6 78 0
11 96 0	7 69 0	6 59 0	6 27 0	7 11 0
11 98 0	7 96 0	7 8 0	6 32 0	7 35 0
14 37 0	8 21 0	7 41 0	6 64 0	7 41 0
14 76 0	8 23 0	7 48 0	6 75 0	7 43 0
22 39 0	8 29 0	7 63 0	7 10 0	7 91 0
23 39 0	8 35 0	7 67 0	7 18 0	7 96 0
30 39 0	8 51 0	7 70 0	7 22 0	9 87 0
30 88 0	8 55 0	7 80 0	7 27 0	9 100 0
35 39 0	8 69 0	7 90 0	7 32 0	11 42 0
35 62 0	10 15 0	8 17 0	7 43 0	12 42 0
37 59 0	10 30 0	14 78 0	7 64 0	27 42 0
37 98 0	13 30 0	17 18 0	7 69 0	27 78 0
39 50 0	14 50 0	17 25 0	7 75 0	27 87 0
39 55 0	15 21 0	17 28 0	7 82 0	34 42 0
39 63 0	15 23 0	17 41 0	7 95 0	37 42 0
39 68 0	15 29 0	17 48 0	8 15 0	41 42 0
39 76 0	15 42 0	17 50 0	9 22 0	42 49 0
39 95 0	15 43 0	17 58 0	9 27 0	42 59 0
59 76 0	15 51 0	17 67 0	9 32 0	42 74 0
0	15 68 0	17 70 0	9 34 0	42 85 0
	15 69 0	17 80 0	9 48 0	42 89 0
	15 73 0	17 87 0	9 64 0	42 95 0

(a) First Part

4.3. Results

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
	16 30 0	28 32 0	9 75 0	42 96 0
	17 21 0	28 40 0	10 13 0	74 78 0
	17 34 0	28 86 0	10 21 0	78 91 0
	18 34 0	32 48 0	10 25 0	0
	21 22 0	32 73 0	10 42 0	
	21 25 0	32 76 0	10 61 0	
	21 30 0	42 94 0	10 68 0	
	21 50 0	48 55 0	11 25 0	
	21 88 0	48 59 0	13 22 0	
	21 92 0	~~48~~ 75 0	13 43 0	
	21 93 0	48 83 0	13 75 0	
	23 25 0	48 94 0	13 95 0	
	24 30 0	63 99 0	14 22 0	
	25 29 0	64 90 0	14 24 0	
	25 35 0	67 86 0	15 22 0	
	25 51 0	67 94 0	15 27 0	
	25 69 0	70 86 0	15 32 0	
	25 84 0	70 95 0	15 64 0	
	26 30 0	80 94 0	15 75 0	
	28 30 0	90 94 0	15 95 0	
	29 30 0	1	17 32 0	
	30 31 0		17 64 0	
	30 35 0		18 65 0	
	30 36 0		18 70 0	
	30 45 0		21 22 0	
	30 51 0		21 24 0	
	30 63 0		21 32 0	
	30 65 0		~~21~~ 40 0	
	30 71 0		21 43 0	
	30 73 0		21 64 0	
	30 74 0		21 69 0	
	30 77 0		21 73 0	
	30 78 0		21 79 0	
	30 83 0		21 82 0	
	30 84 0		21 95 0	
	30 97 0		22 25 0	
	40 69 0		22 39 0	
	41 56 0		22 42 0	
	41 57 0		22 51 0	
	61 71 0		22 61 0	
	71 88 0		22 68 0	
	71 93 0		22 78 0	
	73 88 0		25 27 0	
	0		25 32 0	

(b) Second Part

4.3. Results

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
			25 34 0	
			25 37 0	
			25 41 0	
			25 64 0	
			25 69 0	
			25 75 0	
			25 95 0	
			27 42 0	
			27 61 0	
			32 39 0	
			32 61 0	
			32 68 0	
			37 44 0	
			37 95 0	
			39 69 0	
			39 75 0	
			39 95 0	
			40 61 0	
			42 69 0	
			42 75 0	
			42 95 0	
			43 68 0	
			52 61 0	
			61 64 0	
			61 69 0	
			61 75 0	
			61 82 0	
			61 95 0	
			61 96 0	
			63 71 0	
			63 80 0	
			64 68 0	
			64 80 0	
			65 75 0	
			68 69 0	
			68 95 0	
			69 78 0	
			69 80 0	
			70 95 0	
			76 82 0	
			1	

(c) Third Part

Figure 4.11: Matrix Winmax for the fold 1 to 5 in the lymphoma data set. Winmax: this matrix contains the best 10-fold CV results in the inner loop. For example, winmax will contain 5 gene combinations if these 5 different gene combinations all made no error for this fold. Since we only tested all the 2-gene combinations, the winmax has 3 columns. The first 2 columns contain the 2 genes. The third column contains the number of errors made by the 2-gene combination. For example, the row "1 6 0" means gene combination (1, 6) made 0 error when classifying the samples in this fold. Here 1 and 6 are TS ranks (ranked using only the internal CV data) of the two genes. The row written in bold and italic letters is the gene combination that we randomly selected to build the classifier. That is, we firstly generate a random number between 1 and the row number of the winmax, after that, we select the row corresponding to this number. The last row of a fold is the number of errors by the randomly selected two genes.

4.3. Results

Fold 6	Fold 7	Fold 8	Fold 9	Fold 10
2 24 0	1 14 0	2 6 0	1 30 0	1 3 0
2 43 0	1 33 0	2 22 0	1 32 0	1 26 0
2 52 0	1 72 0	2 25 0	1 38 0	1 33 0
2 62 0	1 74 0	2 30 0	1 46 0	1 36 0
2 70 0	1 77 0	2 34 0	1 49 0	1 47 0
3 17 0	2 14 0	2 50 0	1 65 0	1 56 0
3 24 0	2 33 0	2 65 0	1 70 0	1 83 0
3 43 0	2 37 0	2 79 0	1 77 0	1 84 0
3 62 0	2 41 0	2 87 0	1 91 0	1 86 0
3 70 0	2 72 0	4 22 0	1 97 0	3 4 0
4 66 0	2 74 0	4 25 0	2 32 0	3 13 0
5 45 0	2 77 0	4 30 0	2 72 0	3 70 0
5 82 0	2 79 0	4 34 0	2 95 0	3 80 0
6 13 0	4 14 0	4 65 0	4 7 0	4 26 0
7 86 0	4 72 0	4 93 0	5 30 0	4 33 0
9 12 0	4 74 0	6 17 0	5 32 0	4 36 0
9 45 0	4 77 0	8 17 0	5 65 0	4 47 0
10 39 0	10 23 0	13 99 0	5 70 0	4 51 0
12 43 0	10 82 0	17 22 0	9 20 0	4 56 0
12 62 0	11 17 0	17 23 0	9 33 0	4 63 0
12 70 0	15 64 0	17 30 0	9 95 0	4 84 0
13 38 0	16 23 0	17 33 0	12 75 0	6 13 0
13 86 0	23 25 0	17 34 0	13 33 0	6 43 0
13 97 0	23 26 0	17 36 0	13 84 0	13 25 0
14 45 0	23 33 0	17 40 0	13 95 0	13 26 0
17 36 0	23 58 0	17 50 0	16 33 0	13 30 0
19 43 0	23 72 0	17 65 0	16 74 0	13 33 0
24 45 0	23 77 0	17 75 0	17 24 0	13 35 0
25 45 0	23 90 0	17 76 0	17 29 0	13 37 0
31 38 0	26 59 0	17 78 0	20 30 0	13 47 0
32 43 0	33 50 0	17 93 0	20 32 0	13 53 0
32 70 0	33 80 0	20 23 0	20 46 0	13 56 0
36 43 0	33 82 0	20 25 0	20 70 0	13 57 0
36 62 0	33 95 0	20 30 0	20 77 0	13 84 0
36 70 0	42 95 0	20 34 0	21 33 0	13 86 0
40 45 0	48 92 0	20 65 0	21 94 0	18 26 0
44 97 0	64 81 0	0	21 95 0	18 30 0
45 52 0	72 85 0		23 30 0	18 36 0
45 62 0	74 95 0		23 32 0	18 41 0
45 70 0	77 82 0		23 46 0	18 47 0
45 72 0	77 85 0		23 70 0	18 51 0
45 86 0	0		24 64 0	18 63 0
45 90 0			25 33 0	18 83 0

(a) First Part

4.3. Results

Fold 6	Fold 7	Fold 8	Fold 9	Fold 10
66 97 0			28 33 0	18 84 0
82 97 0			29 71 0	19 88 0
93 94 0			30 95 0	22 43 0
2			32 33 0	26 43 0
			32 95 0	29 43 0
			33 46 0	29 70 0
			33 68 0	29 94 0
			33 70 0	36 59 0
			33 77 0	43 86 0
			33 80 0	61 88 0
			33 91 0	70 86 0
			33 97 0	75 84 0
			39 75 0	80 86 0
			45 75 0	86 94 0
			46 95 0	0
			50 84 0	
			60 64 0	
			65 95 0	
			70 95 0	
			80 95 0	
			95 97 0	
			0	

(b) Second Part

Figure 4.12: Same as Fig. 4.11, for the fold 6 to 10.

4.3. Results

---

the training data. Although some combinations achieved 100% testing accuracy (e.g. the combination (7, 13, 18) in Fig. 4.14), they made a few errors for the training data. Such combinations should be excluded from the evaluation of the classifier. Otherwise, the result would be biased because it took advantage of the testing data.

We also used the combination (5, 12, 30) to predict the five testing samples that are not SRBCT, i.e., the testing sample Nos. 1, 3, 13, 20, and 22. These five samples were labelled as BL, BL, NB, RMS, and RMS, respectively.

In Table 4.4, we compared the numbers of genes required in different methods for the SRBCT data set. This comparison clearly points out that our method can significantly reduce the number genes required for accurate classification.

We did not randomly divide the SRBCT data for evaluation because the provider of the SRBCT data had already divided the data into training and testing sets [2] and this data partition has been used by other authors [3–5]. Hence we also followed this partition for comparison.

4.3.3 Liver Cancer Data

The liver cancer data set [82] (<http://genome-www.stanford.edu/hcc/>) has two classes, i.e. the nontumor liver and HCC. The data set contains 156 samples and the expression data of 1648 important genes. Among them, 82 are HCCs and the other 74 are nontumor livers. We randomly divided the data into 78 training samples and 78 testing samples. In this data set, there are some missing values. We also used the k-nearest neighbor method to fill those missing values [73].

In this data set, we followed the same steps as what we did in the lymphoma and the SRBCT data sets. Firstly, we chose 100 important genes in the training data set. Then we tested all possible 1-gene and 2-gene combinations within the 100 important genes. Among all the 5050 combinations, we found that combination

4.3. Results

Table 4.3: SRBCT gene importance ranking: 50 genes with the highest TSs, in the order of decreasing TSs.

Rank	Gene ID	Image ID	Gene Description
1	GENE842	810057	cold shock domain protein A
2	GENE1955	784224	fibroblast growth factor receptor 4
3	GENE187	296448	insulin-like growth factor 2 (somatomedin A)
4	GENE1389	770394	Fc fragment of IgG, receptor, transporter, alpha
5	GENE509	207274	Human DNA for insulin-like growth factor II (IGF-2); exon 7 and additional ORF
6	GENE2046	244618	ESTs
7	GENE2050	295985	ESTs
8	GENE255	325182	cadherin 2, N-cadherin (neuronal)
9	GENE2198	212542	Homo sapiens mRNA; cDNA DKFZp586J2118 (from clone DKFZp586J2118)
10	GENE246	377461	caveolin 1, caveolae protein, 22kD
11	GENE819	41591	meningioma (disrupted in balanced translocation) 1
12	GENE742	812105	transmembrane protein
13	GENE1003	796258	sarcoglycan, alpha (50kD dystrophin-associated glycoprotein)
14	GENE2022	204545	ESTs
15	GENE851	563673	antiquitin 1
16	GENE1764	44563	growth associated protein 43
17	GENE1319	866702	protein tyrosine phosphatase, non-receptor type 13 (APO-1/CD95 (Fas)-associated phosphatase)
18	GENE1	21652	catenin (cadherin-associated protein), alpha 1 (102kD)
19	GENE1954	814260	follicular lymphoma variant translocation 1
20	GENE129	298062	troponin T2, cardiac
...	...	...	...
...	...	...	...
40	GENE1194	859359	quinone oxidoreductase homolog
41	GENE153	383188	recoverin
42	GENE1093	812965	v-myc avian myelocytomatosis viral oncogene homolog
43	GENE603	42558	glycine amidinotransferase (L-arginine:glycine amidinotransferase)
44	GENE1723	813841	plasminogen activator, tissue
45	GENE1896	789253	presenilin 2 (Alzheimer disease 4)
46	GENE1662	377048	Homo sapiens incomplete cDNA for a mutated allele of a myosin class I, myh-1c
47	GENE2	25725	farnesyl-diphosphate farnesyltransferase 1
48	GENE1422	32299	Homo sapiens myo-inositol monophosphatase 2 mRNA, complete cds
49	GENE1916	80109	major histocompatibility complex, class II, DQ alpha 1
50	GENE1750	233721	insulin-like growth factor binding protein 2 (36kD)

4.3. Results

Figure 4.13: Three dimensional views of gene expression levels of 3-gene combination (5,12,30) that separates the 4 SRBCT sub-types, i.e., BL, EWS, RMS, NB: (a) a view in which RMS and NB can be seen to be separated from other sub-types, (b) a view in which BL can be seen to be separated from EWS. From the two views, the 4 sub-types are clearly separable. Here all the genes are labelled according to their TS ranks. Our FNN found only 1 such 3-gene combination that achieved 100% 10-fold CV accuracy.

Method	Accuracy	Number of genes required
MLP neural network [2]	100% (biased)	96
Nearest shrunken centroids [3]	100% (biased)	43
Evolutionary algorithm [4]	100% (biased)	12
SVM [5]	100% (biased)	20 or 3 PCs
Our FNN	95% (unbiased)	3

Table 4.4: Comparisons of results for the SRBCT data set obtained by different approaches. Here the results obtained in [2], [3], [4], and [5] are all biased, whereas our results for the FNN are unbiased.

4.3. Results

Figure 4.14: Three dimensional views of gene expression levels of 3-gene combination (7, 13, 18) that separates the 4 SRBCT sub-types, i.e., BL, EWS, RMS, NB: (a) a view that includes only the training samples, (b) a view that includes only the testing samples. Although (b) is a perfect separation of the four sub-types, the combination (7, 13, 18) should be excluded from the evaluation of the classifier because it made one error for the training data and had a lower 10-fold CV accuracy than the combination (5, 12, 30).

4.3. Results

Figure 4.15: Expression levels of Gene 1 and Gene 7 in the liver cancer data set. Here all the genes are labelled according to their TS ranks. The HCCs and the nontumor livers are clearly separable from the plot of (1, 7).

(1,7) and (2,82) reached 100% CV accuracy for the training data. Their corresponding test accuracy are 100% and 96.2%, respectively. The averaged accuracy is 98.1%, which can be used to estimate the performance of the classifier.

The clone IDs of gene 1 and 7 are IMAGE: 301122 and IMAGE: 128461, respectively. We plot the gene expression levels of this combination in Fig.4.15. From this plot, we found that the nontumor liver samples and HCC samples were separated quite well.

We also conducted 10-fold double CV to the whole liver cancer data set. The double CV accuracy is 96.15%, i.e., 6 errors in 156 samples. The detailed results are given in Fig.4.16 and Fig.4.17.

For the liver cancer data set, Chen *et al.* [82] used 3180 genes (represented by 3964 cDNA) to classify HCC from the nontumor samples. In comparison with Chen *et al.*'s work [82], our method also greatly reduced the number of genes required to obtain an accurate result.

4.3. Results

Fold 1															
119	123	113	110	136	76	68	137	152	81	108	55	156	84	57	94
2	2	2	2	2	1	1	2	2	1	2	1	2	2	1	2
2	2	2	2	2	1	1	2	2	1	2	1	2	1	1	2
Fold 2															
15	5	153	14	129	130	140	1	106	112	23	78	50	96	26	114
1	1	2	1	2	2	2	1	2	2	1	1	1	2	1	2
1	1	2	1	2	2	2	1	2	2	1	1	1	2	1	2
Fold 3															
111	39	28	21	125	105	99	35	51	142	19	9	134	61	82	3
2	1	1	1	2	2	2	1	1	2	1	1	2	1	1	1
2	1	1	2	2	2	2	1	1	2	1	1	2	1	1	1
Fold 4															
53	97	147	101	86	128	74	4	30	132	17	48	90	10	27	146
1	2	2	2	2	2	1	1	1	2	1	1	2	1	1	2
1	2	2	2	2	2	1	1	1	2	1	1	2	1	1	2
Fold 5															
52	144	64	72	148	16	122	49	67	13	43	25	33	103	42	44
1	2	1	1	2	1	2	1	1	1	1	1	1	2	1	1
1	2	1	1	2	1	2	1	1	1	1	1	1	1	1	1
Fold 6															
133	71	12	104	38	56	149	139	117	91	70	121	107	115	120	75
2	1	1	2	1	1	2	2	2	2	1	2	2	2	2	1
2	1	1	2	1	1	2	2	2	2	2	2	2	2	2	1
Fold 7															
102	65	45	34	24	124	77	20	37	138	80	11	66	100	31	
2	1	1	1	1	2	1	1	1	2	1	1	1	2	1	
2	1	1	1	1	2	1	1	1	2	2	1	1	2	1	
Fold 8															
58	22	8	7	89	62	116	36	59	151	92	93	95	85	60	
1	1	1	1	2	1	2	1	1	2	2	2	2	2	1	
1	1	1	1	2	1	2	1	1	2	2	2	2	2	1	
Fold 9															
54	32	131	69	141	155	98	2	18	29	154	118	40	135	46	
1	1	2	1	2	2	2	1	1	1	2	2	1	2	1	
1	1	2	1	2	2	2	1	1	1	2	2	1	2	1	
Fold 10															
126	63	47	88	145	127	87	79	41	143	83	73	6	109	150	
2	1	1	2	2	2	2	1	1	2	2	1	1	2	2	
2	1	1	2	2	2	2	2	1	2	2	1	1	2	2	

Figure 4.16: The 10 fold partition of the liver cancer data set. The first line is the No. of the fold, e.g., fold 3., means the 3rd fold. The second line is the samples in a fold. For example, fold 1 includes the samples of 119, 123, 113, 110, 136, etc. Here, these numbers is the No. of the sample in the liver cancer data set. For example, 119 indicates the 119th sample of the liver cancer data set. The third line is the label of the respective sample. Here, 1 stands for HCC; 2 stands for nontumor. The forth line are the labels predicted in the outer loop.

4.3. Results

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
1 13 0	1 22 1	1 29 1	1 2 1	1 72 0
13 40 0	1 39 1	1 42 1	1 29 1	1
1	1 51 1	1 56 1	1 40 1	
	1 56 1	1 60 1	1 57 1	
	1 77 1	1 83 1	1 59 1	
	1 94 1	11 42 1	1 74 1	
	7 67 1	1	1 86 1	
	46 66 1		1 91 1	
	0		1 95 1	
			2 49 1	
			5 56 1	
			5 86 1	
			0	

(a) First Part

4.3.4 GCM Data

The above 3 data sets, i.e., the lymphoma, the SRBCT, and the liver cancer data are relatively small data sets. They include 3, 4, and, 2 (sub)types of cancers, respectively. In such small data sets, exhaustive search for 2-gene or 3-gene subsets are possible because it does not require very long computing time. Let us suppose that one gene subset needs 0.5 second to process. Then, it will cost 22.46 hours to

Fold 6	Fold 7	Fold 8	Fold 9	Fold 10
1 9 0	1 21 0	1 10 0	1 14 0	1 30 0
1	1 38 0	0	1 58 0	1 52 0
	1 81 0		0	2 39 0
	1 82 0			22 30 0
	5 54 0			1
	31 81 0			
	0			

(b) Second Part

Figure 4.17: Same as Fig. 4.11, for the liver cancer data set.

4.3. Results

---

search for all possible 3-gene subsets from the 100 genes obtained by gene ranking and selection. However, if we search for all possible 5-gene subsets in the same data, it will cost 435.7 days. Therefore, exhaustive search is only effective for searching subsets with a small number of genes.

However, this limitation does not mean that exhaustive search cannot deal with large and complex data sets. In fact, for data sets with more classes, we can divide the whole classification problem into a group of binary classification problems with “one-against-one” or “one-against-all” schemes [90]. For each binary classification problem, we then use the proposed 2-step exhaustive search. Using this “divide-and-conquer” method, we can greatly reduce the computation time required.

To test the effectiveness of our method for more complex problems, we applied it to the GCM data set that include 14 types of cancers [91] (<http://www-genome.wi.mit.edu/mpr/GCM.html>). This data set contains expression data of 16306 genes and the total 198 samples have already been divided into 2 parts, i.e., 144 for training and the other 54 for testing. Table 4.5 gives the general information of the data set.

We used the “one-against-all” scheme [90] to process the GCM data set. That is, we built 14 binary SVM classifiers and each binary SVM classifier output a probability that a sample belongs to a cancer type. Finally, a sample was categorized to the type of the largest probability. For each binary classifier, we ranked all the genes according to their TSs and then tested all the 2-gene combinations within the top 50 genes. After that, we picked out the combination that achieved best 10-fold cross-validation accuracy in the training data set. Table 4.6 summarizes the genes used by each binary SVM and the best 10-fold cross-validation accuracy it achieved. Using the 14 binary SVMs jointly, we carried out 10-fold cross-validation in the 144 training samples. The SVMs made 22 errors and the

4.3. Results

---

confusion matrix is shown in Table 4.7. When we predicted the 54 testing samples, the SVMs made 12 errors and the confusion matrix is shown in Table 4.8. Since all the 14 binary SVMs used only 2 genes, we plotted the 14 2-gene combinations in Figs. 4.18 and 4.19.

For the GCM data set, the best CV accuracy reported is 81.25% (27 errors in 144 samples) and the best testing accuracy reported is 77.8% (12 errors in 54 samples) with all the 16063 genes [91]. Compared with these results, our 14 2-gene binary SVMs jointly reached 84.7% CV accuracy in the training data (22 errors in 144 samples) and 77.8% testing accuracy with only the 28 genes listed in Table 4.6.

We did not randomly divide the GCM data set for evaluation because the provider of the GCM data had already divided the data into training and testing sets [91]. Hence we also followed this partition for comparison.

4.3.5 Results with Class Separability Importance Ranking

To find out how the gene importance ranking scheme influences the classification result, we also used another ranking scheme, i.e., the class separability (CS) [81]. In the SRBCT data set, we tested the combinations within the top 100 genes selected by the new ranking scheme. We also found the only 3-gene combination that leads to 100% accuracy with TS. However, the 3 genes rank (23, 40, 22) in CS rather (5, 12, 30) in TS ranking [3, 89]. In the liver cancer data set, we also found the only 2-gene (ranked (1, 7) in TS but (1, 16) in CS) combination with 100% accuracy. In the lymphoma data set, we found 216 2-gene combinations within the top 100 genes of the new rank that reached 100% CV accuracy. These 216 combinations included most of the combinations found with the TS [3, 89] importance ranking.

4.3. Results

Figure 4.18: Plots of gene combinations for binary classifiers 1-7. In each plot, the horizontal axis stands for gene 1 in Table 4.6 and the vertical axis stands for gene 2 in Table 4.6 for the corresponding classifier. We used 14 such binary classifiers, each responsible for classifying one of the 14 classes from all the rest. As shown in Figs. 4.18 and 4.19, we found a 2-gene combination for each of the 14 classifiers to best separate a given class from the rest. We then used the “one-against-all” scheme (introduced in Section 2.1.3) to obtain the overall classification results from results of the 14 binary classifiers, which leads to a slightly better cross-validation accuracy and the same testing accuracy compared to the best reported result, but with only 28 genes rather than all 16063 genes required by the existing approach.

4.3. Results

Figure 4.19: Same as Fig.4.18, for binary classifiers 8-14.

4.3. Results

Table 4.5: General information for the GCM data set: the cancer types included, the number of training and testing samples for each type.

Type Number	Cancer Name	Abbr.	Number of Training Samples	Number of Testing Samples
1	Breast	B	8	3
2	Prostate	P	8	2
3	Lung	L	8	3
4	Colorectal	CR	8	3
5	Lymphoma	Ly	16	6
6	Bladder	BL	8	3
7	Melanoma	M	8	2
8	Uterus	U	8	2
9	Leukemia	Le	24	6
10	Renal	R	8	3
11	Pancreas	PA	8	3
12	Ovary	Ov	8	3
13	Mesothelioma	Ms	8	3
14	Brain	C	16	4

4.3. Results

Table 4.6: The 10-fold cross-validation accuracy and the genes used by the 14 binary classifiers.

Classifier Number	Cancer Type Concerned	Gene 1 (Accession)	Gene 2 (Accession)	10-fold Cross Validation Accuracy
1	Breast	RCAA397825_at	RC_AA195229_s_at	97.22%
2	Prostate	X07730_at	RC_AA176975_s_at	99.31%
3	Lung	M24461_at	X02419_rna1_s_at	97.92%
4	Colorectal	X83228_at	U51095_at	99.31%
5	Lymphoma	M27394_s_at	RC_AA233620_at	100%
6	Bladder	RC_AA135095_at	M92449_at	97.92%
7	Melanoma	RC_AA176812_at	U65092_at	98.61%
8	Uterus	M61906_at	U19718_at	98.61%
9	Leukemia	S72024_s_at	RC_AA410338_at	100%
10	Renal	U18088_s_at	M64082_at	98.61%
11	Pancreas	U31449_at	J04040_at	98.61%
12	Ovary	RC_AA456055_at	X54667_s_at	97.22%
13	Mesothelioma	RC_AA419609_at	M62402_at	99.31%
14	Brain	AA093923_at	N76496_at	100%

4.3. Results

Table 4.7: The confusion matrix for the 10-fold cross-validation results of the GCM data set. Rows delineate the predicted types and columns delineate the actual types.

	Actual Type															<i>n</i>
	B	P	L	CR	Ly	BL	M	U	Le	R	PA	Ov	Ms	C		
B	6						1					1			8	
P	1	6	1												8	
L	1		6									1			8	
CR				8											8	
Ly					16										16	
BL						6	1	1							8	
M						1	6					1			8	
U	1							6		1					8	
Le									24						24	
R			2						1	5					8	
PA						1					6	1			8	
Ov						1		1		1		5			8	
Ms							1						7		8	
C						1								15	16	

4.3. Results

Table 4.8: The confusion matrix for the testing results of the GCM data set. Rows delineate the predicted types and columns delineate the actual types.

	Actual Type														<i>n</i>
	B	P	L	CR	Ly	BL	M	U	Le	R	PA	Ov	Ms	C	
B	3							1							4
P		5	1												6
L			4												4
CR				4											4
Ly					6										6
BL	2					1									3
M							2								2
U	1							1							2
Le	1				1				4						6
R	1						1			1					3
PA						1					2				3
Ov						2						2			4
Ms													3		3
C														4	4

4.4. Discussion

4.4 Discussion

For our purpose of finding the smallest gene subsets for accurate cancer classifications, both TS and CS are highly effective ranking schemes, whereas both SVM and FNN are sufficiently good classifiers. Many other gene importance ranking schemes and classifiers may also be used in our approach.

As we have known from the results in the lymphoma data set, the gene combination that gives good separation may not be unique. Furthermore, it is found that highly correlated genes may undertake very similar tasks in a combination. Therefore, if a gene in a combination is replaced by another highly correlated gene, the newly obtained combination may still lead to good classification. Clustering important genes selected in Step 1 will help us discover such important correlations.

In the lymphoma data set, we clustered the 100 selected genes into 20 groups using the K-means method (see e.g., [92]). The clustering result is summarized in Table 4.9. As we mentioned above, (4, 40) is a good combination for the lymphoma data set. We also note that genes 71 and 36 are in the same clusters with genes 4 and 40, respectively, and the two clusters are very small. Therefore, we replaced genes 4 and 40 with genes 71 and 36, respectively. From the plots of these combinations, we found (71, 36) and (4, 36) are good combinations. Furthermore, (71, 40) also roughly separates the three classes.

In addition, we take note that although these gene combinations achieved good separation in the above data sets (experiments), they are not necessarily useful biomarkers in the current form due to a few reasons. First of all, the good separation obtained by some of these gene combinations could possibly be a “transient” phenomena rather than stable biological relationships. To confirm the differentiating ability of these gene combinations, they should be tested in more (and even other types of biological) experiments. Next, the microarray technology is still in

4.4. Discussion

---

its early stage and has limitations on accuracy, sensitivity, specificity, and reproducibility [93]. Even the choice of threshold has a great impact on the findings obtained from microarray experiments [94]. Therefore, we should consider the inconsistency arisen from microarray technique itself when using its findings.

Following the idea of testing small gene combinations, we solved this problem with much smaller numbers of genes compared with the previously published methods. In fact, after finding out the optimal gene combinations, this problem becomes a comparably easy pattern recognition problem, which can be seen from the plots of gene combinations.

In the SRBCT data set, the 3 genes 5, 12, 30 are the genes IGF2, AF1q, and CD99, respectively. It has been reported that CD99 is a marker for EWS [95, 96]. In addition, IGF2 is also reported to be indispensable for the formation of medulloblastoma and RMS [97, 98]. As for the genes where no relations between them and the related cancers were reported, our work calls for the further investigation of such relations.

To sum up, we proposed a simple yet very effective 2-step method for gene selection and gene expression data classification. We applied our method to four well-known microarray data sets, i.e., the lymphoma, the SRBCT, the liver cancer, and the GCM data sets. The results in all the data sets indicate that our method can find minimum gene subsets that can ensure very high prediction accuracy. In addition, although the TS [3, 89] and the CS [81] based approaches have been proven to be effective in selecting important genes for reliable prediction, they are not perfect. To find minimum gene subsets that ensure accurate predictions, we must also consider the cooperations between genes.

4.4. Discussion

Table 4.9: The K-means clustering result for the 100 selected genes in the lymphoma data set: the 20 clusters and the genes included in each cluster. The genes in each cluster may be highly correlated.

Cluster Number	Total Number of Genes in the Cluster	Genes Included in the Cluster
1	12	5, 10, 23, 26, 38, 39, 49, 50, 51, 52, 70, 77
2	2	85, 87
3	1	83
4	3	2, 58, 68
5	3	4, 71, 75
6	7	25, 33, 53, 56, 93, 95, 97
7	6	7, 8, 21, 32, 36, 40
8	3	24, 73, 80
9	3	5, 29, 35
10	2	89, 91
11	1	90
12	1	82
13	1	13
14	1	37
15	13	3, 15, 19, 22, 30, 41, 42, 44, 47, 57, 60, 65, 79
16	37	6, 11, 12, 15, 16, 17, 18, 20, 27, 28, 31, 34, 43, 45, 46, 48, 54, 59, 61, 62, 63, 64, 66, 67, 69, 72, 74, 76, 79, 84, 86, 88, 94, 96, 98, 99, 100
17	1	81
18	1	92
19	1	78
20	1	55

4.4. Discussion

Figure 4.20: Expression levels of three combinations after gene replacement. Here all the genes are labelled according to their TS ranks. Genes 71 and 36 are highly correlated with genes 4 and 40, respectively. The combination (4, 40) separates the three lymphoma sub-types very well as shown in Fig.4.6(b). The above plots show the three combinations, i.e., (71, 40), (71, 36), and (36, 4), which are closely correlated with (4, 40).

Chapter 5

Active Mining Discriminative Feature Sets

5.1 Introduction

In the previous chapter, we proposed a partially exhaustive searching algorithm that is used to find good DFSs. Although it has achieved very good results even in the very large GCM data set, the huge computing time is still a major limitation to this method. Here we still use the GCM data set to illustrate this. As shown in the previous chapter, we jointly used 14 2-gene combinations to classify the whole data set. However, if a 2-gene combination is not enough to classify a specific type in the GCM data, for example, we have to use five or even more genes for a type. Then even the “divide-and-conquer” scheme cannot solve the problem due to the computing time.

To overcome this limitation, we propose a wrapper feature subset selection method called active mining discriminative feature sets (AM-DFS). The AM-DFS proposed here is a forward selection approach. Although some forward selection approaches have been proposed in the literature [67, 99, 100], it is a least studied approach because of its mediocre performance. For example, a forward selection approach under-performed its backward counterpart in [29]. In this dissertation, our AM-DFS improved the performance of forward selection and achieved com-

5.1. Introduction

---

parable results with the SVM-RFE proposed by Guyon *et al.* [27], which is a well-known and successful backward selection approach. In the meantime, our AM-DFS still shares the advantage of forward selection: its speediness. The AM-DFS achieved such performance because of its ability to obtain features according to the “needs” of current features, which is similar to the adaptability used in the pool-based active learning (see, e.g., [44]). The major difference between the AM-DFS and the pool-based active learning is as follows. The AM-DFS uses its adaptability to search for discriminative **features**. In other words, the objective of the AM-DFS is to maximize the discriminating ability of the DFS. From functionality point of view, this adaptability makes the AM-DFS an error-correction process. In Section 5.3.2, we use an example to show this error-correction process. In contrast to the AM-DFS, the pool-based active learning uses its adaptability to search for “unlabelled” data **samples**. In other words, the objective of the pool-based active learning is to obtain more and quality samples for the training process.

The AM-DFS combines the error correction ability of our AM paradigm and the speediness of the forward wrapper. Furthermore, since the working principle of the AM-DFS does not rely on a specific classifier model, the AM-DFS can be easily adapted to different classifiers based on the tasks and the characteristics of the data.

We also conducted comparison studies between the AM-DFS and the SVM-RFE that combines the classification power of SVMs and the selection power of wrapper model and hence is one of the best-performed feature subset selection approaches. A detailed introduction to the SVM-RFE is given in Section 5.3.1. Our AM-DFS is similar with the SVM-RFE in that: (1) it is also a recursive wrapper; (2) it also uses SVMs as the classifier (although other classifiers can also

5.2. Active Mining Discriminative Feature Set

---

be fitted into this framework). The major difference between the AM-DFS and the SVM-RFE is that the former is a forward selection approach but the later is a backward approach. In addition, our AM-DFS borrows the adaptability from the *active learning* paradigm.

The rest of this chapter is organized as follows. In section 5.2, our AM-DFS method, its key components (e.g., the seed, the correction score), and the detailed pseudo code are introduced. In section 5.3, we test our method in three data sets, i.e., the leukemia data set, the colon data set, and the prostate data set. In addition, we also compared the performance of the AM-DFS with the SVM-RFE in this section. In section 5.4, we discuss the results, recommend the future work, and conclude the chapter.

5.2 Active Mining Discriminative Feature Set

Recently, *active learning* has attracted much attention in machine learning field [44, 101]. In general, active learning is a supervised learning algorithm that is able to choose a group of (unlabelled) samples and then label them to create a training set. In this process, if a pool of labelled samples are available to the learner besides the unlabelled samples, this special case of active learning is *pool-based* active learning. The paradigm of pool-based active learning is like this: an active learner (AL) contains three components $\{X, F, Q\}$. X is an input matrix. F is a mapping function from input space to output space that realizes the objective (or the function) of the AL . Q is a query function that is used to search and add unlabelled samples into the X . In other words, the AL has the ability to obtain “new knowledge” that will “benefit” its learning. Compared to passive learners that only contain X and F but no Q , ALs are able to select data for itself based on the learner’s present status and therefore has the potential to obtain a better

5.2. Active Mining Discriminative Feature Set

learning result. The working procedure of an *AL* is described in Figure 5.1. Some applications of active learning [44, 102] have proved its usefulness in improving the performance of a learner.

Figure 5.1: The working principle of an *AL*. Here the *AL* firstly uses the query function Q to search for an unlabelled sample and then assign a label to it using the mapping function F . This labelling is like a “bet” made by the *AL*. However, the *AL* should take the smallest “risk” to make this “bet”. For example, if F is an SVM classifier, the selected unlabelled sample should have a decision value very close to the assigned label and the SVM should be “safe” to assign the label to this sample. Secondly, the selected sample and the assigned label will be included into X to fit the mapping function F . Obviously, if the label is correctly assigned, the newly fitted F must be better or as good as previous because more correct information is used for the mapping. However, if the label is not correctly assigned, the newly fitted F could be worse than previous because more “noise” affects the mapping.

The difference between active learning and our algorithm is the difference in the content of the “knowledge”. In almost all the active learning approaches proposed previously, the function Q is used to search for unlabelled *samples* (or *observations*) to be learned by the *AL*, i.e., the “new knowledge” to be obtained refers to the unlabelled samples. Here we will propose a learning algorithm with a query function $\tilde{Q}$ that is used to search for features according to the current state of the learner, i.e., the “new knowledge” in our algorithm refers to features. Figure 5.2 shows the working principle of our algorithm. Seeing this difference, we call our

5.2. Active Mining Discriminative Feature Set

algorithm *active mining* as opposed to *active learning*.

Figure 5.2: The working principle of an active miner. The algorithm starts with a feature with good discriminative ability to search for the DFS (This feature is called a seed as introduced in detail later). Then, the *AM* fits a mapping function, F , from this DFS to the label. After that, a query function Q is used to search for a new feature that can improve the accuracy of F . Here, we use three kinds of searching schemes as Q . The first scheme (A1) searches for the feature that aims to correct the errors made by the current DFS; in addition to the the routine of A1, the second scheme (A2) adds a step to to improve the prediction of “unreliable samples”; in addition to the the routine of A2, the third scheme (A3) adds a step to exclude “uncorrectable samples”. From this introduction, it is clear that our algorithm (including its three derivatives) are similar to active learning in its adaptability to “new knowledge”. The difference between active learning and our algorithm is the content of the “knowledge”.

In this dissertation, we focus on classification problems. Therefore, the mapping function F is a classifier and the outputs of the learning process are DFSs. We use a ranking scheme called correction scores (introduced in Section 5.2.2) to add features (i.e., the “new knowledge”) that intend to correct errors into DFSs. This new feature adding scheme based on correction scores acts as the query function $\tilde{Q}$. The whole process of our algorithm is therefore called active mining discriminative

5.2. Active Mining Discriminative Feature Set

---

feature sets (AM-DFS) and it can be divided into the following steps.

1. Assign a feature as the first member of a DFS.
2. Fit a classifier using the current DFS.
3. Based on the classification result of the current DFS, add a new feature into the DFS to improve its performance.

Steps 2 and 3 are repeated until a required number of features are included into the DFS. Each of the three steps will be introduced in detail hereafter.

5.2.1 T -Statistics for Selecting Seeds

According to the definition of a DFS, it is obvious that a feature with greater discriminative power is more likely to be included in good DFSs. Therefore, we first rank all the features according to their discriminating capability. Then a number of features with the highest ranks (e.g., the top 20 features) are selected as the first features in DFSs, i.e., the *seeds* of these DFSs. Here we use T -statistic (TS) to do this ranking.

Although a seed with a high TS rank is more likely to lead to a good DFS, it does not mean that the No.1 seed will lead to the best DFS. This is because the t -statistic is only an univariate measure and cannot represent multivariate space property. Therefore, using a number of top features (e.g., top 20 features) as seeds to search for DFSs will greatly increase our algorithm's possibility to find the best DFSs.

We use nonlinear SVMs introduced in Chapter 2.1 as the classifier.

5.2.2 Correction Score

We define a ranking scheme, which we call correction score (CS), to measure a feature's ability to separate the samples that are misclassified (or the prediction

5.2. Active Mining Discriminative Feature Set

---

is “*unreliable*”) by the DFS obtained in the previous round of training. (Here we define a process of picking out a feature and adding it into a DFS as a *round* of training.) Figure 5.3 shows a description of the *CS*. The *CS* of feature i is defined as:

$$CS_i = S_{bi}/S_{wi} \quad (5.1)$$

where

$$S_{bi} = \sum_{j \in C_1} (e_{ij} - \bar{x}_{c1})^2 + \sum_{k \in C_2} (e_{ik} - \bar{x}_{c2})^2 \quad (5.2)$$

$$S_{wi} = \sum_{j \in C_1} (e_{ij} - \bar{x}_{c2})^2 + \sum_{k \in C_2} (e_{ik} - \bar{x}_{c1})^2 \quad (5.3)$$

where e_{ij} and e_{ik} are the values of misclassified samples (or together with unreliable samples) in C_1 and C_2 , respectively. $\bar{x}_{c1}$ and $\bar{x}_{c2}$ are defined in Eq.3.13 and Eq.3.14. S_{bi} is the sum of squares of the inter-class distances (the distances between samples of different classes) among the misclassified samples. S_{wi} is the sum of squares of the intra-class distances (the distances of samples within the same class) among the misclassified samples. A larger *CS* indicates a greater ratio of the inter-class distance to the intra-class distance, and therefore a higher ability of a feature to separate the selected samples.

5.2.3 Our Algorithm

Figure 5.4 shows an example of two classes being classified by a SVM. There are four types of samples, i.e., *reliable samples*, *unreliable samples*, *wrongly labelled samples*, *uncorrectable samples*, are introduced in this figure. Based on the different characters of these samples, our algorithm has three derivatives. For the first derivative (we call it algorithm 1, or *A1*), only wrongly labelled samples are used for searching the next feature to be added into the DFS. In other words, *A1*

5.2. Active Mining Discriminative Feature Set

Figure 5.3: The description of the CS . Here all the samples belong to two classes: squares with label “+1” and triangles with label “-1”. In the upper figure, we plot the samples that receive prediction values ≥ 0.8 or ≤ -0.8 ; In the figure below, we plot the samples that receive the prediction values between -0.8 to 0.8 together with the samples receiving wrong prediction. The CS is calculated to separate the samples in the under figure.

5.2. Active Mining Discriminative Feature Set

---

only tries to correct the errors made in the previous round of training. For the second derivative (we call it algorithm 2, or A2), both wrongly labelled samples and unreliable samples are used for searching the next features to be added into the DFS. In other words, A2 not only tries to correct the errors but also tries to improve the reliability of the prediction made in the previous round of training. For the third derivative (we call it algorithm 3, or A3), uncorrectable samples are deleted in the process of searching the next features of the DFS. In other words, A3 tries to exclude the influence of “possible noise”. The whole process of A1, A2, and A3 is given below.

Algorithm: SVM-based AM-DFS

Inputs:

Training samples: $\mathbf{X}_{tr} = [\mathbf{x}_{tr1}, \mathbf{x}_{tr2}, \dots, \mathbf{x}_{trl}]^T$, testing samples $\mathbf{X}_{test}$

Class labels for training, and testing samples: $\mathbf{Y}_{tr} = [y_{tr1}, y_{tr2}, \dots, y_{trl}]^T$, $\mathbf{Y}_{test}$

The number of seeds to be used for searching for DFSs: M

The number of features to be included in a DFS: N

Initialize:

Set the initial values for the parameters for defining different types of samples as described in Fig. 5.4 (wrongly labelled, unreliable, uncorrectable, and reliable samples.).

Initialize a DFS to an empty matrix: $\text{DFS} = []$.

Choose a seed:

Calculate the TS for each feature in $\mathbf{X}_{tr}$.

for($m = 1$; until $m < M$; $m++$)

{

Select a feature with the m -th largest TS as the seed (S).

$S \rightarrow \text{DFS}$.

5.2. Active Mining Discriminative Feature Set

Repeat until: N features are left in the DFS:

{

    Train an SVM with DFS then obtain E_{tr} .

    Pick out the misclassified samples $\mathbf{X}_e = [\mathbf{x}_{e1}, \mathbf{x}_{e2}, \dots, \mathbf{x}_{et}]^T$.

{Pick out the UNRELIABLE SAMPLES and put them into $\mathbf{X}_e$ if they are not already there. /*this step is only used in the A2 and A3*/}

{Pick out the POSSIBLE NOISE SAMPLES and delete them from $\mathbf{X}_e$ /*this step is only used in the A3*/}

    Calculate CS for each feature in $\mathbf{X}_e$.

    Pick out the feature with the largest CS and put it into the DFS.

}

}

Output

DFSs

5.2.4 Good DFS Selection

Since each seed leads to a different DFS, our algorithm usually outputs a number of DFSs (e.g., 20 DFSs when we use 20 seeds for searching). When a DFS is used as a predictor, one needs to select a DFS with good generalization capability from all the obtained DFSs. In this dissertation, we use 10-fold cross validation (CV) to do this selection. That is, for all the data sets in test, we carry out 10-fold CV for the training samples for each DFS and then select the DFS that achieves the highest accuracy to build the predictor. If more than one DFSs achieve the highest CV accuracy, we select the DFS that has the smallest $||\mathbf{w}||^2$ value.

Another DFS selection method we tested is based on an estimation of error-dimensionality relationship ($\hat{P}\hat{M}$) introduced in Appendix A. We used the ($\hat{P}\hat{M}$) to estimate the possible errors of DFSs and then select the DFS that achieves

5.2. Active Mining Discriminative Feature Set

Figure 5.4: Four types of samples for a classification problem. Here all the “x” samples belong to -1 group and all the “+” samples belong to +1 group. The solid line between the -1 and +1 groups is the decision boundary $f(\mathbf{X}) = 0$. A: a *wrongly labelled sample*, that is, the decision boundary labels A wrongly; B: an *unreliable sample*, that is, the decision boundary labels B correctly. However, B has a decision value close to 0 (e.g., -0.1) and is very close to the decision boundary; C: an *uncorrectable sample*, that is, the decision boundary label C wrongly. Furthermore, the decision value for C is close to the wrong label +1 (e.g., +0.9); D: a *reliable sample*, that is, the decision boundary labels D. Furthermore, the decision value for D is close to the desired label -1 (e.g., -0.9).

5.3. Experimental Results

---

smallest error for prediction. This criterion works well for some (but not all) of data sets. In the future, we plan to do more test on this criterion.

5.3 Experimental Results

We tested the proposed methods in three microarray gene expression data sets, i.e., the leukemia data set, the colon data set, and the prostate data set. In addition, we also compared the results achieved by the SVM-based AM-DFS with the results achieved by the SVM-RFE.

5.3.1 SVM-RFE

As we mentioned previously, the SVM-RFE algorithm [27] is one of the most popular approaches for feature subset selection nowadays. It is also regarded as one of the best approaches to this problem. In this study, we compare the performance our algorithms with the SVM-RFE.

The basic idea of the SVM-RFE is recursively removing the feature that has the least impact on the margin of a SVM, i.e., $(1/2)\|\mathbf{w}\|^2$, until a predefined number of features are left in the DFS. The ranking criterion for feature k (i.e., R_k) is as follows.

$$R_k = \left| \|\mathbf{w}\|^2 - \|\mathbf{w}^{(k)}\|^2 \right| = \frac{1}{2} \sum_{i,j} \left| \alpha_i^* \alpha_j^* y_i y_j \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) - \alpha_i^{*(k)} \alpha_j^{*(k)} y_i y_j \phi(\mathbf{x}_i^{(k)})^T \phi(\mathbf{x}_j^{(k)}) \right| \quad (5.4)$$

where the factors with $^{*(k)}$ in the upper right corner are calculated with the k -th feature being removed. In other words, R_k describes how much the variation caused to the margin, i.e. $(1/2)\|\mathbf{w}\|^2$, if the k -th feature is removed. The SVM-RFE each time removes the feature with the smallest R_k until a required number of features are left in the DFS. In addition, $\alpha_i^{*(k)}$ is supposed to be equal to α_i^*

5.3. Experimental Results

---

here for the purpose of simplification.

5.3.2 Leukemia Data Set

The leukemia data set [103] can be obtained at (<http://www-genome.wi.mit.edu/cancer/>). The samples in this data set belong to two types of leukemia, acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL). Golub *et al.* divided the data into two parts: 38 of them are used for training; other 34 independent samples are for testing. Among the 38 training samples, there are 27 ALL samples and 11 AML samples. Among the 34 testing samples, there are 20 ALL samples and 14 AML samples. The entire leukemia data set contains the expression values of 7129 genes. We normalized this data set by subtracting the mean and dividing the standard deviation across each sample (rather than each gene). In the erratum for [27] (<http://www.clopinet.com/isabelle/Papers/RFE-erratum.html>), Guyon *et al.* pointed out that preprocessing (e.g., normalization) across the genes would introduce bias. Therefore, we normalized the data set across the samples.

Table 5.1 shows the results obtained by the SVM-RFE. The DFS with 8 features selected by the SVM-RFE achieved the testing accuracy of 97.06% (i.e., 1 error in the 34 testing samples).

After the experiments of SVM-RFE, we tested our algorithm A2 using the top 500 features. We used the top 20 features as seeds to carry out search. Because of the limitation of space, we do not show the results for all the 20 seeds. Table 5.2 shows the DFSs originated from the No. 17 seed and their corresponding training and testing accuracies. In Table 5.3, we show the results jointly achieved by the 20 seeds. Since we used 20 seeds for searching, we obtained 20 DFSs with a specific number (e.g., 5) of genes, e.g., the DFSs originated from the No. 1 seed, No. 2 seed, etc. Then, we selected a DFS that achieved the best 10-fold CV in the training data for prediction. In this table, a 8-feature DFS also achieved 97.06%

5.3. Experimental Results

---

testing accuracy.

In this data set, the SVM-RFE and our A2 algorithm both achieved 97.06% testing accuracy with a 8-feature DFS. For the DFSs with other numbers of features, the two approaches also achieved comparable results as shown in Tables 5.1 and 5.3, respectively.

Besides the above experiments, we also randomly divided the 72 samples into 38 training samples and 34 testing samples for 50 times. We tested our AM-DFS and the SVM-RFE using this 50 partitions. In addition, we also tested a method using standard forward selection as searching scheme and Fisher linear discriminant (see, e.g., [104]) as classifier, which was applied to the leukemia data set and the colon data set in [69]. Table 5.4 shows the averaged accuracies and standard deviations obtained by the AM-DFS, SVM-RFE, and Fisher linear discriminant in the 50 different partitions. Figure 5.6 plots these results.

In [69], the Fisher linear discriminant classifier were used to demonstrate **highly biased** classification results in the leukemia and colon data sets, which were obtained by using all the samples (including both training and testing samples) for feature selection.

From our experiments in the leukemia data set, we also found that both the SVM-RFE and the AM-DFS achieved better results than the Fisher linear discriminant. In terms of accuracy, the SVM-RFE and the AM-DFS achieved comparable results. However, a significant advantage of the AM-DFS over the SVM-RFE is the much reduced the computational cost.

From these results, we found the AM-DFS is an error correction process and very effective and efficient in finding good DFSs. Let us use the first DFS in Table 5.3 to illustrate this. Since this DFS only contain 2 genes (*U50136-rna1-at* and *X17042-at*), we plot this gene combination in Fig.5.5. In the first round training,

5.3. Experimental Results

Number of Features	Training Accuracy	Testing Accuracy	Running Time (S)	The DFS
2	100%	88.24%	1561.3	1834, 6539
4	100%	94.12%	1550.4	4847, 1834, 6201, 6539
8	100%	97.06%	1549.9	3320, 4847, 1834, 6201, 6539, 1829, 6055, 5598
16	100%	82.35%	1567.6	3320, 4847, 5039, 1834, 461, 6201, 6539, 1829, 2301, 6055, 4951, 2078, 5596, 3140, 2131
32	100%	88.24%	1564.2	see Supplementary Information
64	100%	91.18%	1563.3	see Supplementary Information
128	100%	91.18%	1450.7	see Supplementary Information

Table 5.1: SVM-RFE for the leukemia data set with the top 500 features. The complete table can be obtained on the Supplementary Information webpage at <http://www.ntu.edu.sg/home5/pg02317674>.

5.3. Experimental Results

No. of Features	Training Accuracy	Testing Accuracy	Running Time (S)	The DFS
2	86.84%	88.24%	7.27	2121 , 3012
4	92.11%	91.18%	7.93	2121 , 3012, 2995, 4438
8	100%	97.06%	8.45	2121 , 3012, 2995, 4438, 7044, 6642, 5056, 668
16	100%	97.06%	10.21	2121 , 3012, 2995, 4438, 1898, 4232, 5056, 1256, 3504, 400, 5432, 2179, 997, 6002, 2830, 1924
32	100%	94.12%	13.63	See Supplementary Information
64	100%	82.36%	20.81	See Supplementary Information
128	100%	88.24%	36.90	See Supplementary Information

Table 5.2: The DFSs obtained by the A2 for the leukemia data set with the top 500 features and the Seed 17. The running time includes all the time spent on searching for DFSs (selecting seed, classification, computing correction scores, etc.). The complete table can be obtained on the webpage at www.ntu.edu.sg/home5/pg02317674.

5.3. Experimental Results

No. of Features	Training Accuracy	Testing Accuracy	CV Accuracy	The Seed	Running Time (S)
2	100%	82.35%	94.74%	3	41.32
4	94.74%	91.18%	94.74%	7	53.51
8	100%	97.06%	97.37%	2	75.2
16	100%	94.12%	97.37%	10	106.1
32	100%	73.25%	100%	7	161.2
64	100%	82.36%	100%	17	284.0
128	100%	94.12%	100%	8	542.2

Table 5.3: The DFSs obtained by the A2 algorithm for the leukemia data with the top 500 features and their training and testing results. The running time includes all the time spent on searching for DFSs (selecting seed, classification, computing correction scores, etc).

only the seed, i.e., gene *U50136_rna1_at* was input to the SVM. Because gene *U50136_rna1_at* has a high *TS*, the SVM misclassified only three samples that were indicated with arrows. In the second round training, the algorithm selected the gene that had the best capability to separate the three misclassified samples, i.e., gene *X17042_at*. We found in Fig.5.5(a) that gene *X17042_at* “dragged” the misclassified samples far away from the classes they mistakenly assigned to in the previous round of training. Therefore, with the help of gene *X17042_at*, The DFS increased its training accuracy from 92.11% to 100%. The 38 training samples were perfectly separated by this DFS. We also plotted the 34 testing sample in Fig.5.5(b). The testing accuracy of the DFS is 82.35%.

Although some papers has claimed 100% accuracy (e.g., in leave-one-out cross-validation) obtained in the leukemia data set (see e.g., [27] [30]), it is widely accepted that the leukemia data contains at least one mislabelled sample [105]. One possible reason for the 100% accuracy in the leukemia data set could be the se-

5.3. Experimental Results

Figure 5.5: Gene expression values for the two genes in the first DFS in the leukemia data set. (a) a plot includes only the training samples; (b) a plot includes all the training and testing samples.

lection bias or the bias introduced in the preprocessing steps [29]. Quite a few reported results failed to obtain 100% test accuracy. In [106], the method called the nearest shrunken centroids obtained 94.1% testing accuracy. In [105], an SVM using a kernel gene selection method also obtained 94.1% testing accuracy. In [107], a method for mining emerging patterns obtained 91.2% testing accuracy (i.e., 3 errors in the 34 testing samples).

5.3. Experimental Results

Table 5.4: The comparison of the averaged accuracies obtained by the SVM-RFE, AM-DFS (A1, A2, and A3), and Fisher's linear discriminant in the leukemia data set in 50 random splits of the data set into 38 training samples and 34 testing samples. The numbers in the brackets are the corresponding standard deviations.

No. of Features	SVM-RFE	A1	A2	A3	Fisher
2	83.2% (4.06%)	81.9% (3.73%)	84.5% (4.01%)	83.6% (3.76%)	81.1% (4.13%)
4	85.5% (4.17%)	88.7% (4.11%)	90.3% (4.93%)	84.7% (2.98%)	82.5% (3.82%)
8	88.3% (4.41%)	91.6% (3.85%)	93.7% (3.99%)	94.1% (4.22%)	87.6% (5.02%)
16	90.6% (3.92%)	93.7% (4.17%)	92.9% (5.28%)	95.2% (4.44%)	90.3% (4.38%)
32	92.1% (3.83%)	92.9% (4.91%)	94.2% (4.26%)	92.2% (4.20%)	91.3% (3.84%)
64	93.5% (4.12%)	94.1% (3.92%)	91.3% (3.25%)	89.7% (4.74%)	92.1% (4.51%)
128	94.5% (3.91%)	90.4% (2.98%)	93.3% (4.21%)	91.3% (5.08%)	93.0% (4.72%)

5.3. Experimental Results

Figure 5.6: The comparison of the averaged accuracies obtained by the SVM-RFE, AM-DFS (A1, A2, and A3), and Fisher’s linear discriminant in the leukemia data set in 50 random splits into 38 training samples and 34 testing samples.

5.3.3 Colon Data Set

The colon data set [108] (<http://www.molbio.princeton.edu/colondata/>) contains 22 normal samples and 40 colon tumor samples. In this data set, there are expression values of 2000 genes. Because Alon *et al.* did not divide the data set into training and testing samples, we need to divide the data set by ourselves. Here we used 10-fold cross validation (CV) to test our method. We randomly divided the whole data set into 10 parts, among which 2 parts contained 7 samples each and the rest 8 parts contained 6 samples each. We cross-validated each part, i.e., we assigned each part as testing samples once and trained the SVM using the rest 9 parts. After that, we combined the testing results for each part together. To overcome overfitting, in the 9 parts held out from the testing part, we only used 7 parts for training and used the rest 2 parts for validation. For each testing part, we used the top 10 genes with the largest TS s (calculated only using the training samples in the 7 parts) as the seeds and then obtained 10 DFSs for each testing part. We chose the DFS that achieved the best validation accuracy as the final

5.3. Experimental Results

DFS that we used to test the testing part. If more than one DFSs had the same validation accuracies, we further checked their training accuracies and chose the one with the best training accuracy. If there were still more than one DFSs with the same training accuracy, we randomly chose one from these DFSs.

The testing, the validation, and the training results are shown in Table 5.5, Table 5.7, and Table 5.6, respectively. From these results, we found that the AM-DFS made 7 errors in the 10-fold CV test. The 10 DFSs used for testing samples totally contains 38 genes, i.e., each DFS being chosen contains 3.8 genes on average.

To evaluate our results in the colon data set, we also selected some gene groups (e.g., top 5, 10, 15 genes and so on) with other three most widely used gene selection approaches, i.e., t -statistics, f -statistics (FS), and the SVM-RFE. To make the comparison more objective, we used the same 10-fold partition and the same training, validation, and testing assignment for those parts as the ones we used for testing SVM based AM-DGS. Similarly, we also used 7 parts to obtain the TS , FS , and SVM-RFE ranking results and used 2 parts to tune the parameters of the SVM. The comparison results are listed in Table 5.8. From Table 5.8, we found that the fewest errors made for the TS , FS , and SVM-RFE based approaches are 10, 11, and 11, respectively. And they used 15 genes, 10 genes, and 10 genes to obtain these results. In contrast, the SVM based AM-DFS made only 7 errors by using the DFSs with 3.8 genes on average.

Similar with what we did for the leukemia data set, we also randomly divided the 62 colon samples into 31 training samples and 31 testing samples for 50 times. Then, we tested our AM-DFS, the SVM-RFE, and Fisher linear discriminant using this 50 partitions. Table 5.9 shows the averaged accuracies and standard deviations. Figure 5.7 plots these results with respect to the number of genes.

From these experiments, we also found that both the SVM-RFE and the AM-

5.3. Experimental Results

Figure 5.7: The comparison of the averaged accuracies obtained by the SVM-RFE, AM-DFS (A1, A2, and A3), and Fisher's linear discriminant in the colon data set in 50 random splits into 31 training samples and 31 testing samples.

DFS achieved better results than the Fisher linear discriminant. In terms of accuracy, the SVM-RFE and the AM-DFS achieved comparable results. However, a significant advantage of the AM-DFS over the SVM-RFE is the greatly reduced the computational cost.

5.3.4 Prostate Data Set

The prostate data set [109] (<http://www-genome.wi.mit.edu/MPR/prostate.>) contains 50 nontumor samples and 52 prostate tumor samples. In this data set, there are expression values of 12600 genes. Considering the huge amount of genes included in the data set, we firstly ranked all the genes according to their TS s and then picked out the 1000 genes with the largest TS s for further processing. We followed the exactly same procedures as what we did in the colon data set to process this data set, i.e., we carried out 10-fold CV to the data using the SVM based AM-DFS. Tables 5.10, 5.11, and 5.12 show the testing, validation, and training results, respectively. The SVM-based AM-DFS achieved 87.3% 10-fold CV accuracy

5.3. Experimental Results

Table 5.5: The testing accuracy for the 10-fold cross validation conducted in the colon data with our SVM-based AM-DFS. The numbers in brackets are the number of errors made in the corresponding fold.

Fold No.	1	2	3	4	5	6	7	8	9	10
Fold Size	7	7	6	6	6	6	6	6	6	6
Accuracy for Seed 1 (%)	85.7 (1)	71.4 (2)	83.3 (1)	33.3 (4)	66.7 (2)	100 (0)	83.3 (1)	100 (0)	83.3 (1)	83.3 (1)
Accuracy for Seed 2 (%)	100 (0)	100 (0)	83.3 (1)	50 (3)	66.7 (2)	100 (0)	83.3 (1)	100 (0)	100 (0)	66.7 (2)
Accuracy for Seed 3 (%)	85.7 (1)	100 (0)	100 (0)	33.3 (4)	66.7 (2)	100 (0)	83.3 (1)	100 (0)	83.3 (1)	66.7 (2)
Accuracy for Seed 4 (%)	100 (0)	100 (0)	83.3 (1)	66.7 (2)	83.3 (1)	100 (0)	83.3 (1)	83.3 (1)	83.3 (1)	66.7 (2)
Accuracy for Seed 5 (%)	85.7 (1)	85.7 (1)	83.3 (1)	50 (3)	66.7 (2)	100 (0)	83.3 (1)	83.3 (1)	66.7 (2)	50 (3)
Accuracy for Seed 6 (%)	57.1 (3)	100 (0)	66.7 (2)	50 (3)	66.7 (2)	83.3 (1)	83.3 (1)	83.3 (1)	83.3 (1)	50 (3)
Accuracy for Seed 7 (%)	85.7 (1)	71.4 (2)	83.3 (1)	50 (3)	83.3 (1)	50 (3)	83.3 (1)	100 (0)	83.3 (1)	66.7 (2)
Accuracy for Seed 8 (%)	71.4 (2)	71.4 (2)	83.3 (1)	16.7 (5)	83.3 (1)	83.3 (1)	83.3 (1)	100 (0)	100 (0)	83.3 (1)
Accuracy for Seed 9 (%)	85.7 (1)	71.4 (2)	83.3 (1)	33.3 (4)	83.3 (1)	83.3 (1)	83.3 (1)	100 (0)	83.3 (1)	50 (3)
Accuracy for Seed 10 (%)	100 (0)	85.7 (1)	66.7 (2)	50 (3)	83.3 (1)	83.3 (1)	66.7 (2)	100 (0)	100 (0)	83.3 (1)

5.3. Experimental Results

Table 5.6: The training accuracy for the 10-fold cross validation conducted in the colon data using our SVM-based AM-DFS.

Fold No.	1	2	3	4	5	6	7	8	9	10
Accuracy for Seed 1 (%)	85.7	85.7	85.7	88.1	85.7	85.7	88.1	85.7	85.7	90.5
Accuracy for Seed 2 (%)	85.7	85.7	85.7	88.1	85.7	83.3	83.3	83.3	92.9	90.5
Accuracy for Seed 3 (%)	78.6	81.0	81.0	88.1	81.0	85.7	81.0	88.1	81.0	83.3
Accuracy for Seed 4 (%)	83.3	83.3	90.5	90.5	81.0	78.6	73.8	85.7	78.6	95.2
Accuracy for Seed 5 (%)	81.0	78.6	81.0	83.3	88.1	76.1	88.1	81.0	88.1	81.0
Accuracy for Seed 6 (%)	76.2	76.2	76.2	85.7	76.2	73.8	78.6	76.1	78.6	78.6
Accuracy for Seed 7 (%)	81.0	81.0	78.6	92.9	88.1	66.7	76.2	78.6	76.2	81.0
Accuracy for Seed 8 (%)	85.7	85.7	85.7	81.0	85.7	88.1	88.1	88.1	85.7	92.9
Accuracy for Seed 9 (%)	78.6	78.6	78.6	83.3	85.7	85.7	92.9	85.7	78.6	95.2
Accuracy for Seed 10 (%)	83.3	78.6	83.3	90.5	95.2	83.3	73.8	85.7	83.3	81.0

Table 5.7: The validation accuracy for the 10-fold cross validation conducted in the colon data using our SVM-based AM-DFS.

Fold No.	1	2	3	4	5	6	7	8	9	10
Accuracy for Seed 1 (%)	76.9	84.6	78.6	92.9	78.6	78.6	78.6	78.6	78.6	78.6
Accuracy for Seed 2 (%)	100	100	85.7	85.7	92.9	78.6	100	78.6	100	85.7
Accuracy for Seed 3 (%)	100	100	100	100	85.7	100	92.9	85.7	100	100
Accuracy for Seed 4 (%)	100	100	100	100	100	100	92.9	85.7	92.9	92.9
Accuracy for Seed 5 (%)	92.3	92.3	92.9	92.9	78.6	92.9	100	85.7	78.6	92.9
Accuracy for Seed 6 (%)	84.6	69.2	78.6	92.9	85.7	78.6	92.9	85.7	100	85.7
Accuracy for Seed 7 (%)	92.3	92.3	85.7	100	92.9	85.7	92.9	100	100	92.9
Accuracy for Seed 8 (%)	76.9	76.9	71.4	85.7	92.9	92.9	78.6	100	100	85.7
Accuracy for Seed 9 (%)	84.6	92.3	78.6	85.7	92.9	92.9	100	92.9	78.6	92.9
Accuracy for Seed 10 (%)	76.9	84.6	92.9	92.9	85.7	92.9	71.4	92.9	92.9	92.9

5.3. Experimental Results

Table 5.8: The comparison of the errors in the colon data set with different approaches.

No. of Genes	1	2	5	10	15	20	30	50	100	200	500
t-statistics	17	11	15	11	10	13	14	17	16	17	21
f-statistics	17	10	15	11	11	13	14	16	16	17	21
SVM-RFE	18	12	14	11	16	16	21	22	22	22	22
SVM-based AM-DFS	7 errors using DFSs with 3.8 genes on average.										

(i.e., 13 errors in the 102 samples) using the DFSs with 5 genes on average. In addition, we also compared our methods with the t -statistic, f -statistic, and the SVM-RFE in Table 5.13. We found that TS -based method achieved 87.3% with at least 15 genes. It also achieved a slightly better accuracy, i.e., 88.2% (12 errors in the 102 samples), with 200 genes. The FS -based method achieved 87.3% accuracy with at least 20 genes and 88.2% accuracy with 200 and 500 genes, respectively. The best accuracy for this data, i.e., 89.2% (11 errors in the 102 samples), was achieved by the SVM-RFE-based method with 100 genes. It also achieved 87.3% accuracy with 30 genes. Although the SVM-based AM-DFS did not achieved the best accuracy in this data set, it obtained the result very close to the best one with much smaller DFSs.

We also randomly divided the 102 prostate samples into 51 training samples and 51 testing samples for 50 times. Then, we tested our AM-DFS, the SVM-RFE, and Fisher linear discriminant using these 50 partitions. Table 5.15 shows the averaged accuracies and standard deviations. Figure 5.8 plots these results with respect to the number of genes.

5.3. Experimental Results

Table 5.9: The comparison of the averaged accuracies obtained by the SVM-RFE, AM-DFS (A1, A2, and A3), and Fisher's linear discriminant in the colon data set in 50 random splits into 31 training samples and 31 testing samples. The numbers in the brackets are the corresponding standard deviations.

No. of Features	SVM-RFE	A1	A2	A3	Fisher
2	69.3% (5.32%)	71.3% (5.12%)	68.6% (3.71%)	68.9% (4.26%)	66.5% (4.92%)
4	73.5% (6.21%)	73.3% (4.83%)	76.2% (4.21%)	72.8% (3.98%)	69.2% (4.88%)
8	72.8% (4.17%)	77.2% (5.58%)	75.8% (3.96%)	76.5% (4.17%)	71.7% (4.58%)
16	75.9% (4.13%)	78.6% (4.96%)	80.1% (4.07%)	78.4% (4.33%)	75.0% (5.10%)
32	77.3% (5.92%)	77.8% (5.31%)	78.9% (6.03%)	77.2% (4.11%)	74.2% (4.92%)
64	79.6% (3.86%)	76.5% (4.92%)	79.3% (4.57%)	79.1% (3.97%)	77.3% (4.37%)
128	78.5% (4.91%)	77.8% (4.87%)	80.2% (5.23%)	79.4% (4.65%)	76.5% (5.01%)

5.3. Experimental Results

Table 5.10: The testing accuracy for the 10-fold cross validation conducted in the prostate data with our SVM-based AM-DFS.

Fold No.	1	2	3	4	5	6	7	8	9	10
Fold Size	11	11	10	10	10	10	10	10	10	10
Accuracy for Seed 1 (%)	72.7 (3)	63.6 (4)	80 (2)	90 (1)	100 (0)	90 (1)	90 (1)	90 (1)	90 (1)	80 (2)
Accuracy for Seed 2 (%)	72.7 (3)	63.6 (4)	90 (1)	100 (0)	100 (0)	90 (1)	80 (2)	90 (1)	100 (0)	90 (1)
Accuracy for Seed 3 (%)	90.9 (1)	63.6 (4)	90 (1)	90 (1)	100 (0)	90 (1)	70 (3)	100 (0)	100 (0)	80 (2)
Accuracy for Seed 4 (%)	90.9 (1)	72.7 (3)	90 (1)	80 (2)	90 (1)	80 (2)	90 (1)	90 (1)	90 (1)	90 (1)
Accuracy for Seed 5 (%)	90.9 (1)	63.6 (4)	70 (3)	80 (2)	80 (2)	70 (3)	80 (2)	90 (1)	90 (1)	90 (1)
Accuracy for Seed 6 (%)	72.7 (3)	81.8 (2)	80 (2)	80 (2)	80 (2)	90 (1)	80 (2)	90 (1)	100 (0)	80 (2)
Accuracy for Seed 7 (%)	90.9 (1)	81.8 (2)	100 (0)	90 (1)	100 (0)	70 (3)	90 (1)	80 (2)	90 (1)	90 (1)
Accuracy for Seed 8 (%)	90.9 (1)	81.8 (2)	80 (2)	100 (0)	100 (0)	90 (1)	90 (1)	90 (1)	100 (0)	90 (1)
Accuracy for Seed 9 (%)	81.8 (2)	100 (0)	80 (2)	50 (5)	50 (5)	70 (3)	80 (2)	100 (0)	80 (2)	70 (3)
Accuracy for Seed 10 (%)	90.9 (1)	63.6 (4)	70 (3)	80 (2)	80 (2)	80 (2)	90 (1)	80 (2)	90 (1)	80 (2)

5.3. Experimental Results

Table 5.11: The validation accuracy for the 10-fold cross validation conducted in the prostate data with our SVM-based AM-DFS.

Fold No.	1	2	3	4	5	6	7	8	9	10
Accuracy for Seed 1 (%)	81.0	90.5	77.3	86.4	81.8	81.8	86.4	77.3	72.7	72.7
Accuracy for Seed 2 (%)	81.0	90.5	77.3	95.5	77.3	95.5	86.4	81.8	81.8	90.9
Accuracy for Seed 3 (%)	85.7	90.5	86.4	77.3	95.5	86.4	86.4	86.4	86.4	86.4
Accuracy for Seed 4 (%)	81.0	90.5	81.8	77.3	86.4	81.8	81.8	86.4	95.5	77.3
Accuracy for Seed 5 (%)	76.2	90.5	81.8	77.3	81.8	90.9	77.3	77.3	72.7	77.3
Accuracy for Seed 6 (%)	81.0	85.7	86.4	81.8	77.3	90.9	95.5	72.7	81.8	86.4
Accuracy for Seed 7 (%)	90.5	95.2	86.4	81.8	72.7	72.7	86.4	86.4	77.3	86.4
Accuracy for Seed 8 (%)	81.0	85.7	86.4	77.3	86.4	95.5	86.4	95.5	86.4	95.5
Accuracy for Seed 9 (%)	90.5	81.0	90.9	81.8	81.8	90.9	81.8	77.3	90.9	86.4
Accuracy for Seed 10 (%)	71.4	90.5	81.8	81.8	81.8	77.3	86.4	90.9	95.5	90.9

5.3. Experimental Results

Table 5.12: The training accuracy for the 10-fold cross validation conducted in the prostate data with our SVM based AM-DFS.

Fold No.	1	2	3	4	5	6	7	8	9	10
Accuracy for Seed 1 (%)	88.6	95.7	94.3	90	94.3	91.4	88.6	88.6	90	92.9
Accuracy for Seed 2 (%)	87.1	94.3	94.3	94.3	90	94.3	95.7	85.7	84.3	95.7
Accuracy for Seed 3 (%)	87.1	94.3	87.1	92.9	90	97.1	92.9	95.7	94.3	88.6
Accuracy for Seed 4 (%)	91.4	91.4	91.4	94.3	92.9	80	94.3	84.3	85.7	94.3
Accuracy for Seed 5 (%)	90	90	87.1	94.3	87.1	87.1	87.1	97.1	90	91.4
Accuracy for Seed 6 (%)	82.9	85.7	85.7	80	80	94.3	94.3	95.7	84.3	88.6
Accuracy for Seed 7 (%)	92.9	92.9	92.9	84.3	88.6	91.4	84.3	87.1	78.6	84.3
Accuracy for Seed 8 (%)	94.3	94.3	94.3	91.4	82.9	88.6	91.4	91.4	84.3	91.4
Accuracy for Seed 9 (%)	91.4	91.4	91.4	78.6	77.1	84.3	97.1	92.9	82.9	88.6
Accuracy for Seed 10 (%)	91.4	94.3	80	74.3	74.3	95.7	94.3	82.9	91.4	87.1

5.3. Experimental Results

Number of Genes	1	2	5	10	15	20	30	50	100	200	500
t-statistics	24	18	15	14	13	13	17	15	14	12	13
f-statistics	24	19	15	14	14	13	16	14	14	12	12
SVM-RFE	30	24	14	17	15	14	13	14	11	14	15
SVM-based AM-DFS	13 errors using DFSs with 5 genes on average.										

Table 5.13: The comparison of the errors in the prostate data set with different approaches.

Method	SVM-RFE	SVM-based AM-DFS
Time Used (s)	386.6250	1.2204

Table 5.14: The comparison of computing times used for finding a 4-gene DFS both with the SVM-RFE and with the SVM-based AM-DFS in the colon data.

Figure 5.8: The comparison of the averaged accuracies obtained by the SVM-RFE, AM-DFS (A1, A2, and A3), and Fisher's linear discriminant in the prostate data set in 50 random splits into 51 training samples and 51 testing samples.

5.3. Experimental Results

Table 5.15: The comparison of the averaged accuracies obtained by the SVM-RFE, AM-DFS (A1, A2, and A3), and Fisher's linear discriminant in the prostate data set in 50 random splits into 51 training samples and 51 testing samples. The numbers in the brackets are the corresponding standard deviations.

No. of Features	SVM-RFE	A1	A2	A3	Fisher
2	74.2% (3.91%)	73.6% (3.88%)	75.2% (4.22%)	75.2% (4.10%)	74.9% (3.98%)
4	76.8% (4.77%)	75.9% (4.26%)	77.4% (3.96%)	78.1% (4.07%)	76.0% (4.08%)
8	80.2% (4.08%)	79.2% (3.87%)	79.3% (4.10%)	80.1% (4.17%)	78.5% (4.13%)
16	82.5% (4.03%)	81.7% (5.03%)	81.8% (4.67%)	78.8% (4.92%)	81.3% (4.98%)
32	79.2% (4.90%)	81.9% (3.93%)	81.6% (5.01%)	80.9% (4.85%)	79.2% (4.87%)
64	81.7% (4.62%)	78.9% (4.32%)	80.5% (4.17%)	81.2% (4.84%)	81.7% (5.04%)
128	79.9% (3.87%)	82.0% (4.58%)	81.9% (4.58%)	78.7% (4.98%)	79.9% (4.93%)

5.4. Discussion

5.4 Discussion

Our experiments show that our AM-DFS algorithm is able to achieve the results comparable with the SVM-RFE. The major advantage of the AM-DFS over the SVM-RFE is the reduced computing time and is therefore more suitable for large data sets. For example, for the leukemia data set, the SVM-RFE used 1561.3 seconds to obtain a 2-feature DFS in the 500-features input space. However, our algorithm A2 used only 7.27 seconds (including selecting seeds, classification, and computing correction scores) to obtain a 2-feature DFS with the Seed 17. Even we used the top 20 features as seeds, the total computing time is only 41.32 seconds (including selecting seeds, classification, and computing correction scores), which is also much shorter than that of the SVM-RFE. In the colon data set that contains 2000 genes, the AM-DFS used only 4 rounds of training to obtain a 4-gene DFS. In contrast, the SVM-RFE used 1996 rounds of training to reach a 4-gene DFS. In addition, SVM-RFE usually spends more time on a round of training than the SVM-based AM-DFS because SVM-RFE needs to process higher-dimensional data compared to the SVM based AM-DFS. Table 5.14 compares the computing time of searching for 4-gene DFSs both with the SVM-RFE and with the SVM-based AM-DFS in the colon data set. We used the same training data and the same computer to do the simulation.

In addition, since the SVM-RFE spends most of time on calculating R_i , its computing time is roughly in portion to d^2 . Here d is the number of features to be eliminated. Therefore, the time required by the SVM-RFE will increase exponentially when more features are to be eliminated. In contrast, the computing time for the AM-DFS is linear with the number of features to be included into a DFS, which makes it superior to the SVM-RFE for high dimensionality reduction problems, such as microarray data analysis.

5.4. Discussion

Another major difference between the AM-DFS and the SVM-RFE is the former usually obtains a number of DFSs originated from a number of seeds. Therefore, we use the 10-fold CV to select a “best” DFS to build the predictor. In fact, this step is a model selection procedure. It is a major challenge for the AM-DFS because predicting the performance of a DFS is a very challenging task [110]. Compared with complicated approaches (e.g., nested CV), the 10-fold CV is fast and easy to use. However, the trade-off is its inaccuracy in selecting good DFSs. For example, the 10-fold CV criterion (plus the $\frac{1}{2}||\mathbf{w}'||^2$ criterion) usually did not pick out the DFS that achieved the best testing accuracy from the 20 candidates for the leukemia data. Certainly, the great noise and imbalance of the leukemia data partially caused the problem. The model selection scheme is also an important factor to be improved. In the future, we will try other approaches in the future that may improve the final performance of the AM-DFS.

In our experiments, we found that the DFSs that achieved best accuracy were usually not unique. Sometimes very different DFSs can obtain same good results. This is to some degree coincide with the conclusions in [111], although the gene sets were used to predict survival rate in [111]. From this point of view, the AM-DFS is attractive in enabling its users to visit a number of different DFSs originated from different seeds. In contrast, the SVM-RFE only leads to one DFS.

Moreover, the AM-DFS is a framework that is not dedicated to SVM classifiers. This framework can be fitted in by other classifiers. In addition, the criterion for choosing “unreliable samples” can also be replaced by other evaluation standards, for example, mutual information.

In conclusion, this chapter proposes a feature subset selection approach, i.e. the AM-DFS, that has the adaptability similar with the scheme used in the pool-based active learning. As a forward selection approach, the AM-DFS achieved

5.4. Discussion

---

comparable results in the three experimental studies with the SVM-RFE, which is a backward selection approach and one of the best performed ones. Although a model selection step is needed for the AM-DFS (which is unnecessary for the SVM-RFE), its speediness is a big advantage over the SVM-RFE. Furthermore, the AM-DFS and its derivatives are open framework that can be adapted to other classifiers. Simulations in the leukemia, colon, and the prostate data sets prove that our algorithm leads to accurate classifications with very small feature sets.

Chapter 6

Conclusions

In this Chapter, we summarize the methods proposed in this dissertation. In addition, we also discuss some limitations of these methods. After that, we will recommend some possible future works.

6.1 Summary

In this dissertation, our research is focused on feature subset selection problem. Furthermore, our study is mainly carried out in the context of microarray gene expression data analysis.

6.1.1 Feature Importance Ranking

We firstly studied some feature importance ranking schemes, for example, t -statistic, class separability measure, Fisher-ratio. Although there are some other ranking approaches can be found in the literature, these three ranking schemes are the most commonly used methods and among the most successful ranking schemes. Our study shows that these methods usually achieve comparable results when they are used as gene selection methods. In the data sets we tested the t -statistic obtained a slightly better results than the other methods. Although this slight superiority possibly comes from the specific data sets that favor t -statistic, we mainly use the t -statistic as the ranking scheme because these schemes are only used as a

6.1. Summary

---

preliminary selection step.

6.1.2 A Partially Exhaustive Searching Algorithm

If we do not consider the computing time as an obstacle, exhaustive search must be the best feature subset selection method because it visits all the possible feature combinations. However, if the computing time is included in consideration, the exhaustive search becomes an unfeasible or even useless method. In this dissertation, we find that exhaustive search can be a good method in gene expression data analysis if it is used in a proper way. First of all, we do not need to search for DFSs among all the features. On the contrary, we only search among some features with top importance ranks, for example, the top 50 genes ranked by t -statistics. Secondly, we find very small gene subsets usually with 2 or 3 genes that can ensure highly accurate classification in almost all the data sets we tested. Considering these two points, our partially exhaustive searching algorithm becomes an attractive approach for such problems. Furthermore, some biological studies have confirmed the relationship between cancers and the genes included in our very small DGSs, which further proves the usefulness of our partially exhaustive searching scheme.

6.1.3 Active Mining Discriminative Feature Sets

In the experiments conducted for the partially exhaustive searching algorithm, we found the cooperation among features in a DFS has a critical impact on the performance of the DFS. Therefore, we extend this idea by proposing the AM-DFS algorithm. The AM-DFS emphasizes the cooperation among features by using an adaptive learning procedure borrowed from the pool-based active learning. We applied the AM-DFS to the cancer classification problems that is a special case of feature subset selection. The AM-DFS obtained very good results comparable

6.2. Limitations and Future Directions

---

with the results achieved by the SVM-RFE. However, the AM-DFS is much faster than the SVM-RFE because it is a forward searching approach but SVM-RFE is a backward searching approach.

6.2 Limitations and Future Directions

A major limitation of this study is the lack of links between the results obtained by our algorithms and the real-world biological facts. For example, our algorithms find a number of DFSs that are able to classify cancer A and cancer B in the simulation. However, do these DFSs truly reflect the biological differences between A and B? In fact, it is very likely that only a very small portion of the DFSs reflect the underlying stable biological fact. The others only capture some “transient states”. To overcome this limitation, a possible solution is repeating the biological experiments using a few genes in the DFSs found by our algorithms several times and check whether they still can differentiate A and B.

Another obstacle to find biologically stable DFSs is the great noise usually contained by microarray gene expression data. In fact, reproducing the process of microarray experiments is very difficult [93]. Therefore, our research also calls for improved microarray platforms that can produce accurate and stable results.

We used 10-fold cross validation to select a good DFS from some of them in our AM-DFS algorithm. However, this criterion is not accurate sometimes. Therefore, we considered the second approach of estimating the relationship between accuracy and dimensionality. In the future, we will test this approach more and try to make it suitable for different data sets.

Although our algorithms can find different DFSs, we do not know how these DFSs interact to each other. Let us assume we have two DFSs, i.e. DFS1 and DFS2, both of which are able to achieve 75% accuracy. If we put these two DFSs

6.2. Limitations and Future Directions

---

together and input them into the classifier, what will happen? In fact, we not only need to study the cooperation among genes, but also need to study the interaction among gene sets, which can be found among different genomic data sets [112].

In this study, the major index we used to evaluate the “quality” of DFSs is the testing accuracy. However, only using this index may be not enough to fully evaluate DFSs (See e.g., [113]). In the future, we plan to include other criteria for this evaluation task. In addition, now we use a sample’s distance to the decision boundary to determine whether it is “reliable” in the A_2 of the AM-DFS. We also plan to try other criteria to do this judgement.

Author's Publications

Patent Application

1. F. Chu and L. Wang. Discriminative Feature Selection System Using Active Mining Technique. *U.S. Patent Application No. 11/767,514*, filed on 06/24/2007.

Journal Papers

1. L. Wang, F. Chu, and W. Xie. Accurate Cancer Classification Using Expressions of Very Few Genes. *IEEE/ACM Transactions On Computational Biology and Bioinformatics*, 4:40-53, 2007.
2. F. Chu and L. Wang. Applications of Support Vector Machines to Cancer Classification With Microarray Data. *International Journal of Neural Systems*, 15:475-484, 2005.
3. L. Wang, N. Zhou, and F. Chu. A General Wrapper Approach to Selection of Class-dependent Features. *IEEE Transactions On Neural Networks*, accepted.
4. F. Chu and L. Wang. Active Mining Discriminative Gene Sets. *IEEE Transaction on Nanobioscience*, revised.
5. F. Chu and L. Wang. Active Mining Discriminative Feature Sets. *Journal of Machine Learning Research*, submitted.

6.2. Limitations and Future Directions

Book Chapters

1. F. Chu and L. Wang. Cancer Classification with Microarray Data using SVMs, in *Bioinformatics using Computational Intelligence Paradigms*, Udo Seiffert and Lakhmi C. Jain (Editors), 167-190, Springer-Verlag, 2005.
2. F. Chu, G. Jin, and L. Wang. Cancer Diagnosis and Protein Secondary Structure Prediction Using Support Vector Machines, in *Support Vector Machines: Theory and Applications*, Lipo Wang (Editor), 343-363, Springer-Verlag, 2005.
3. F. Chu and L. Wang. Biomedical Data Mining Using RBF Neural Networks, in *Encyclopedia of Data Warehousing and Mining*, John Wang (Editor), Idea Group Publishing, Hershey, PA, 2005.

Conference Papers

1. F. Chu and L. Wang. A Novel Approach Searching for Discriminative Gene Sets. *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC2006)*, 1660-1665, Oct., 2006.
2. F. Chu and L. Wang. Active Mining Discriminative Gene Sets, *Proceedings of 8th International Conference on Artificial Intelligence and Soft Computing*, Lecture Notes in Computer Science, 4029, 880-889, June, 2006.
3. F. Chu and L. Wang. Applying RBF Neural Networks to Cancer Classification Based on Gene Expressions, *Proceedings of World Congress of Computational Intelligence*, July, 2006.
4. F. Chu and L. Wang. Estimating Error-Dimensionality Relationship for Gene Expression Based Cancer Classification, *Proceedings of 1st International Conference on Biomedical Engineering*, Singapore, Sep. 2004.

6.2. Limitations and Future Directions

---

5. F. Chu, W. Xie, and L. Wang. Gene Selection and Cancer Classification Using a Fuzzy Neural Network, *Proceedings of 23rd International Conference of the North American Fuzzy Information Processing Society (NAFIPS2004)*, 2:555-559, June, 2004.
6. F. Chu and L. Wang. Gene Expression Data Analysis Using Support Vector Machines, *Proceedings of the International Joint Conference on Neural Networks (IJCNN'03)*, 3:2268-2271, July 2003.
7. X. Fu, L. Wang, K. S. Chua, and F. Chu. Training RBF neural networks on unbalanced data, *Proceedings of the 9th International Conference on Neural Information Processing (ICONIP '02)*, 2:1016-1020, Nov. 2002.

Appendix A

Estimating Error-Dimensionality Relationship

In this appendix, we introduce a simple method for estimating error-dimensionality relationship for Fisher linear discriminant (see, e.g., [104]). The reason why we choose the FLD as our classifier is because it is a simple yet very powerful classifier. And complicated classifiers will make the estimation of classification errors also very complicated or even impossible.

A.0.1 Fisher Linear Discriminant

Mathematically, an FLD can be described as follows. There are two groups of samples, $\{\mathbf{Y}_{1i}, i = 1, 2, \dots, n_1\}$ and $\{\mathbf{Y}_{2i}, i = 1, 2, \dots, n_2\}$, which belong to class C_1 and class C_2 , respectively. Their mean vectors $\mathbf{M}_1$ and $\mathbf{M}_2$ are given by

$$\mathbf{M}_1 = \frac{1}{n_1} \sum_{i=1}^{n_1} \mathbf{Y}_{1i} \quad (\text{A.1})$$

$$\mathbf{M}_2 = \frac{1}{n_2} \sum_{i=1}^{n_2} \mathbf{Y}_{2i}. \quad (\text{A.2})$$

And their within-class covariances, S_1 and S_2 , are given by

$$S_1 = \frac{1}{n_1 - 1} \sum_{i=1}^{n_1} (\mathbf{Y}_{1i} - \mathbf{M}_1)(\mathbf{Y}_{1i} - \mathbf{M}_1)' \quad (\text{A.3})$$

$$\mathbf{S}_2 = \frac{1}{n_2 - 1} \sum_{i=1}^{n_2} (\mathbf{Y}_{2i} - \mathbf{M}_2)(\mathbf{Y}_{2i} - \mathbf{M}_2)'. \quad (\text{A.4})$$

Therefore, the overall within-class covariance, $\mathbf{S}$, can be obtained by:

$$\mathbf{S} = \frac{(n_1 - 1)\mathbf{S}_1 + (n_2 - 1)\mathbf{S}_2}{n_1 + n_2 - 2}. \quad (\text{A.5})$$

The basic idea of FLDs is to project multivariate samples into one dimension in which the samples belonging to different classes should be separated as much as possible. According to [104], such a projection can be achieved by the transformation below:

$$y = \mathbf{W}'\mathbf{Y} = (\mathbf{M}_1 - \mathbf{M}_2)'\mathbf{S}^{-1}\mathbf{Y}. \quad (\text{A.6})$$

where $\mathbf{W}'$ is the transpose of $\mathbf{W}$, and $\mathbf{S}^{-1}$ is the inverse of $\mathbf{S}$. Noting that y is a sum of a number of random variables, the discriminant threshold that is obtained based on the *central limit theorem* (see e.g. [104]) should be

$$t = \frac{1}{2}(\mathbf{M}_1 - \mathbf{M}_2)'\mathbf{S}^{-1}(\mathbf{M}_1 + \mathbf{M}_2). \quad (\text{A.7})$$

Hence, the FLD classifies a sample, e.g. $\mathbf{Y}_0$, with the following rule:

If $(\mathbf{M}_1 - \mathbf{M}_2)'\mathbf{S}^{-1}\mathbf{Y}_0 \geq t$, $\mathbf{Y}_0$ is categorized to C_1 ;
otherwise, if $(\mathbf{M}_1 - \mathbf{M}_2)'\mathbf{S}^{-1}\mathbf{Y}_0 < t$, $\mathbf{Y}_0$ is categorized to C_2 .

A.0.2 Relationship between Dimensionality and Classification Errors

Since all the samples can be regarded as observations of random variables, a classifier designed with these samples can also be regarded as a function of random variables, i.e., the classifier's parameters are random variables. Furthermore, its

---

probability of misclassification (PM) is also a random variable. Therefore, we can use the *expectation* of PM ($\hat{PM}$) to *estimate* the errors the classifier will make.

In fact, a lot of efforts have been made to determine the relationship between dimensionality and $\hat{PM}$ [114–116]. For an FLD, its $\hat{PM}^{(F)}$ can be calculated with the following equation [117]:

$$\hat{PM}^{(F)} = \Phi \left\{ -\frac{\hat{\delta}}{2} \left[\left(1 + \frac{p}{n_1 + n_2 - p} \right) \left(1 + \frac{(p-1)(n_1 + n_2)}{n_1 n_2 \hat{\delta}^2} \right) \right]^{-\frac{1}{2}} \right\} \quad (\text{A.8})$$

where $\Phi(a)$ is the standard Gaussian cumulative distribution function

$$\Phi(a) = \int_{-\infty}^a \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt. \quad (\text{A.9})$$

$\hat{\delta}^2$ is an unbiased estimate of the Mahalanobis distance between the two classes.

$$\hat{\delta}^2 = \left[(\mathbf{M}_1 - \mathbf{M}_2)' \mathbf{S}^{-1} (\mathbf{M}_1 - \mathbf{M}_2) \frac{n_1 + n_2 - p - 3}{n_1 + n_2 - 2} \right] - \frac{n_1 + n_2}{n_1 n_2} p. \quad (\text{A.10})$$

Bibliography

- [1] A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu, et al. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, 2000.
- [2] J. M. Khan, J. S. Wei, M. Ringner, L. H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. R. Antonescu, C. Peterson, et al. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine*, 7:673–679, 2001.
- [3] R. Tibshirani, T. Hastie, B. Narashiman, and G. Chu. Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proc. Natl. Acad. Sci. USA*, 99:6567–6572, 2002.
- [4] J. M. Deutsch. Evolutionary algorithms for finding optimal gene sets in microarray prediction. *Bioinformatics*, 19:45–52, 2003.
- [5] Y. Lee and C. K. Lee. Classification of multiple cancer types by multiclass support vector machines using gene expression data. *Bioinformatics*, 19:1132–1139, 2003.
- [6] G. Piatetsky-Shapiro and W. J. Frawley, editors. *Knowledge Discovery In Databases*. AAAI Press/MIT Press, Cambridge, MA., 1991.

Bibliography

- [7] N. Kambhatla and T. K. Leen. Dimension reduction by local principal component analysis. *Neural Comp.*, 9:1493–1516, 1997.
- [8] J. Porrill and J. Stone. Independent components analysis for signal separation and dimension reduction. *Technical Report 124*, Psychology Department, Sheffield University, Sheffield, England.
- [9] A. L. Boulesteix. Pls dimension reduction for classification with microarray data. *Statistical Applications in Genetics and Molecular Biology.*, 3:33, 2004.
- [10] W. Pan. A comparative review of statistical methods for discovering differentially expressed genes in replicated microarray experiments. *Bioinformatics*, 18:546–554, 2002.
- [11] B. L. Welch. The generalization of student’s problem when several different population are involved. *Biometrika*, 34:28–35, 1947.
- [12] R. A. Fisher. The goodness of fit of regression formulae and distribution of regression coefficients. *Journal of the Royal Statistical Society*, 85:597–612, 1922.
- [13] Y. Lai, B. Wu, L. Chen, and H. Zhao. Statistical method for identifying differential gene-gene coexpression patterns. *Bioinformatics*, 20:3146–3155, 2004.
- [14] L. Yu and H. Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In T. Fawcett and N. Mishra, editors, *ICML*, pages 856–863. AAAI Press, 2003.
- [15] M. Hall. Correlation-based feature selection for machine learning. Hall, M. A. 1998. Correlation-based Feature Selection for Machine Learning. Ph.D

Bibliography

- diss. Hamilton, NZ: Waikato University, Department of Computer Science., 1998.
- [16] X. Fu and L. Wang. Data dimensionality reduction with application to simplifying rbf network structure and improving classification performance. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 33:399–409, 2003.
- [17] E. P. Xing, M. I. Jordan, and R. M. Karp. Feature selection for high-dimensional genomic microarray data. In *Proc. 18th International Conf. on Machine Learning*, pages 601–608. Morgan Kaufmann, San Francisco, CA, 2001.
- [18] Y. Wu and A. Zhang. Feature selection for classifying high-dimensional numerical data. *CVPR*, 02:251–258, 2004.
- [19] P. Hancock, A. Burton, and V. Bruce. Face processing : Human perception and principal components analysis. *Memory and Cognition*, 24:26–40, 1996.
- [20] K. Torkkola. Feature extraction by non-parametric mutual information maximization. *Journal of Machine Learning Research*, 3:1415–1438, 2003.
- [21] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97:273–324, 1997.
- [22] M. A. Hall and G. Holmes. Benchmarking attribute selection techniques for discrete class data mining. *IEEE Transactions on Knowledge and Data Engineering*, 15(6):1437–1447, 2003.
- [23] C. N. Hsu, H. J. Huang, and S. Dietrich. The annigma-wrapper approach to fast feature selection for neural nets. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 32:207–212, 2002.

Bibliography

- [24] R. Setiono and H. Liu. Neural-network feature selector. *IEEE Transactions on Neural Networks*, 8:654–662, May 1997.
- [25] K. Toutanova and C. Manning. Feature selection for a rich hpsg grammar using decision trees. In *Proceedings of CoNLL-2002*, pages 77–83. Taipei, Taiwan, 2002.
- [26] A. R. Dabney and J. D. Storey. Optimal feature selection for nearest centroid classifiers, with applications to gene expression microarrays, 2005.
- [27] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422, 2002.
- [28] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature selection for SVMs. In *NIPS*, pages 668–674, 2000.
- [29] C. Ambroise and G. J. McLachlan. Selection bias in gene extraction on the basis of microarray gene-expression data. *PNAS*, 99:6562–6566, 2002.
- [30] Y. Wang, F. Makedon, J. Ford, and J. Pearlman. Hykgene: a hybrid approach for selecting marker genes for phenotype classification using microarray gene expression data. *Bioinformatics*, 21:1530–1537, 2005.
- [31] M. Sebban and R. Nock. A hybrid filter/wrapper approach of feature selection using information theory. *Journal of Pattern Recognition*, 35:835–846, 2002.
- [32] C. Bhattacharyya, L. R. Grate, A. Rizki, D. Radisky, F. J. Molina, M. I. Jordan, M. J. Bissell, and I. S. Mian. Simultaneous classification and relevant feature identification in high-dimensional spaces: application to molecular profiling data. *Signal Process.*, 83:729–743, 2003.

Bibliography

- [33] H. Wang, D. Bell, and F. Murtagh. Axiomatic approach to feature subset selection based on relevance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:271–277, 1999.
- [34] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.
- [35] A. Jemal, R. Siegel, E. Ward, T. Murray, J. Xu, and M. J. Thun. Cancer Statistics, 2007. *CA Cancer J Clin*, 57:43–66, 2007.
- [36] M. Silverstein, M. Lagios, A. Recht, D. Allred, S. Harms, R. Holland, D. Holmes, L. Hughes, R. Jackman, and T. Julian. Image-detected breast cancer: State of the art diagnosis and treatment. *Journal of the American College of Surgeons*, 201:586–597, 2005.
- [37] M. K. Kalra, M. M. Maher, P. R. Mueller, and S. Saini. State-of-the-art imaging of pancreatic neoplasms. *Br J Radiol*, 76:857–865, 2003.
- [38] E. Even-Sapir, Y. Parag, H. Lerman, M. Gutman, C. Levine, M. Rabau, A. Figer, and U. Metser. Detection of Recurrence in Patients with Rectal Cancer: PET/CT after Abdominoperineal or Anterior Resection. *Radiology*, 232:815–822, 2004.
- [39] M. Chatterjee, A. Ionan, S. Draghici, and M. A. Tainsky. Epitomics: global profiling of immune response to disease using protein microarrays. *OMICS: A Journal of Integrative Biology*, 10:499–506, 2006.
- [40] M. A. Tainsky, M. Chatterjee, N. K. Levin, S. Draghici, and J. Abrams. Multianalyte tests for the early detection of cancer: Speedbumps and barriers. *Biomarker Insights*, 2007.

Bibliography

- [41] H. Lodish, A. Berk, P. Matsudaira, C. A. Kaiser, M. Krieger, M. P. Scott, L. Zipursky, and J. Darnell. *Molecular Cell Biology*. W H Freeman and Co., New York, U.S.A, 5th edition, 2003.
- [42] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, 270:467–470, 1995.
- [43] A. Abdullah-Sayani, J. M. Bueno de Mesquita, and M. J. van de Vijver. Technology insight: tuning into the genetic orchestra using microarrays—limitations of dna microarrays in clinical practice. *Nat Clin Pract Oncol*, 3:501–516, September 2006.
- [44] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2:45–66, 2001.
- [45] Z. R. Yang and K. C. Chou. Bio-support vector machines for computational proteomics. *Bioinformatics*, 20:735–741, 2004.
- [46] Z. R. Yang. Biological applications of support vector machines. *Briefings in Bioinformatics*, 5:328–338, 2004.
- [47] Y. Lee, Y. Lin, and G. Wahba. Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radiance data. *Journal of American Statistical Association*, 99:67–81, 2004.
- [48] Z. Hou and J. M. Parker. Texture defect detection using support vector machines with adaptive gabor wavelet features. *wacv-motion*, 1:275–280, 2005.

Bibliography

- [49] F. Rossi and N. Villa. Support vector machine for functional data classification. *Neurocomputing*, 69:730–742, March 2006.
- [50] B. E. Wohlberg, D. M. Tartakovsky, and A. Guadagnini. Subsurface characterization with support vector machines. *IEEE Trans. Geosci. Remote Sens.*, 44:47–57, 2006.
- [51] B. E. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152, 1992.
- [52] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
- [53] Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.
- [54] V. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, September 1998.
- [55] P. Wolfe. A duality theorem for nonlinear programming. *Quarterly of Applied Mathematics*, 20:173–195, 1961.
- [56] J. C. Platt. Fast training of support vector machines using sequential minimal optimization. pages 185–208, 1999.
- [57] T. M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, 14:326–334, 1965.
- [58] S. Knerr, L. Personnaz, and G. Dreyfus. Single-layer learning revisited: A stepwise procedure for building and training a neural network. In *F. Fogel-*

Bibliography

- man Soulie and J. Herault (eds.) *Neurocomputing: Algorithms, Architecture and Applications*,, pages 41–50. Springer-Verlag, 1990.
- [59] T. Hastie and R. Tibshirani. Classification by pairwise coupling. In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. The MIT Press, 1998.
- [60] Y. Frayman and L. Wang. Data mining using dynamically constructed recurrent fuzzy neural networks. In X. Wu, K. Ramamohanarao, and K. B. Korb, editors, *PAKDD*, volume 1394 of *Lecture Notes in Computer Science*, pages 122–131. Springer, 1998.
- [61] Y. Frayman and L. Wang. A dynamically-constructed fuzzy neural controller for direct model reference adaptive control of multi-input-multi-output nonlinear processes. *Soft Computing*, 6:244–253, 2002.
- [62] C. M. Higgins and R. M. Goodman. Fuzzy rule-based networks for control. *IEEE Transactions on Fuzzy Systems*, 2:82–88, 1994.
- [63] L. X. Wang and J. M. Mendel. Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man, and Cybernetics*, 22:1414–1427, 1992.
- [64] M. Sugeno and G. T. Kang. Structure identification of fuzzy model. *Fuzzy Sets Syst*, 28:15–33, 1988.
- [65] P. Werbos. Beyond regression: new tools for prediction and analysis in the behavioral sciences.
- [66] L. Rutkowski and K. Cpalka. Flexible neuro-fuzzy systems. *IEEE Transactions on Neural Networks*, 14:554–574, 2003.

Bibliography

- [67] A. R. Webb. *Statistical Pattern Recognition*. John Wiley and Sons Ltd., West Sussex, England, 2nd edition, 2002.
- [68] A. W. Whitney. A direct method of nonparametric measurement selection. *IEEE Transactions on Computers*, 20:1100–1103, 1971.
- [69] C. Ambroise and G. J. McLachlan. Selection bias in gene extraction on the basis of microarray gene-expression data. *Proc. Natl. Acad. Sci. USA*, 99:6562–6566, 2002.
- [70] P. Pudil, J. Novovičová, and J. Kittler. Floating search methods in feature selection. *Pattern Recogn. Lett.*, 15(11):1119–1125, 1994.
- [71] D. J. Newman, S. Hettich, C. L. Blake, and C. J. Merz. UCI repository of machine learning databases, 1998.
- [72] A. A. Alizadeh, M. B. Eisen, E. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu, et al. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, February 2000.
- [73] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, 17:520–525, 2001.
- [74] O. L. Mangasarian, W. N. Street, and W. H. Wolberg. Breast cancer diagnosis and prognosis via linear programming. *Operations Research*, 43:570–577, 1995.
- [75] S. Haykin. *Neural Networks: A Comprehensive Foundation, (2nd Ed.)*. Prentice-Hall, Inc, 1999.

Bibliography

- [76] S. Dudoit, J. Fridlyand, and T. Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *J Am Stat Assoc*, 97:77–87, 2002.
- [77] K. Z. Mao. Rbf neural network center selection based on fisher ratio class separability measure. *IEEE Transactions on Neural Networks*, 13:1211–1217, 2002.
- [78] V. G. Tusher, R. Tibshirani, and G. Chu. Significance analysis of microarrays applied to the ionizing radiation response. *Proc Natl Acad Sci USA*, 98:5116–5121, 2001.
- [79] J. Devore and R. Peck. *Statistics: the Exploration and Analysis of Data*, 3rd Ed. Duxbury Press, Pacific Grove, CA, 1997.
- [80] R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu. Class prediction by nearest shrunken centroids with applications to DNA microarrays. *Statistical Science*, 18:104–117, 2003.
- [81] S. Dudoit, J. Fridlyand, and T. P. Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, 97:77–87, 2002.
- [82] X. Chen, S. T. Cheung, S. So, S. T. Fan, C. Barry, J. Higgins, K. M. Lai, J. Ji, S. Dudoit, I. O. Ng, et al. Gene expression patterns in human liver cancers. *Molecular Biology of the Cell*, 13:1929–1939, 2002.
- [83] Y. Frayman, L. Wang, and C. Wan. Cold rolling mill thickness control using the cascade-correlation neural network. *Control and Cybernetics*, 31:327–342, 2002.

Bibliography

- [84] Y. Frayman and L. Wang. Data mining using dynamically constructed fuzzy neural networks. *Lecture Notes in Artificial Intelligence*, 1394:122–131, 1998.
- [85] M. West, C. Blanchette, H. Dressman, E. Huang, S. Ishida, R. Spang, H. Zuzan, J. A. Olson Jr., J. R. Marks, and J. R. Nevins. Predicting the clinical status of human breast cancer by using gene expression profiles. *Proc. Natl. Acad. Sci. USA*, 98:11462–11467, 2001.
- [86] E. Freyhult, P. Prusis, M. Lapinsh, J. E. Wikberg, V. Moulton, and M. G. Gustafsson. Unbiased descriptor and parameter selection confirms the potential of proteochemometric modelling. *BMC Bioinformatics*, 6:50, 2005.
- [87] S. Dudoit, M. J. V. D. Laan, S. Keles, A. M. Molinaro, S. E. Sinisi, and S. L. Teng. Loss-based estimation with cross-validation: application to microarray data analysis and motif finding. *U.C. Berkeley Division of Biostatistics Working Paper Series*, (137), 2003.
- [88] A. Barrier, M. J. V. D. Laan, and S. Dudoit. Prognosis of stage II colon cancer by non-neoplastic mucosa gene expression profiling. *U.C. Berkeley Division of Biostatistics Working Paper Series*, (179), 2003.
- [89] J. Devore and R. Peck. *Statistics: the Exploration and Analysis of Data*. Duxbury Press, Pacific Grove, CA, third edition, 1997.
- [90] C. C. Chang and C. J. Lin. A comparison of methods for multi-class support vector machines. *IEEE Transactions on Neural Networks*, 13:415–425, 2002.
- [91] S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J. Mesirov, T. Poggio, W. Gerald, M. Loda, E. Lander, and T. Golub. Multiclass cancer diagnosis using tumor gene expression signature. *Proc. Natl Acad. Sci. USA*, 98:15149–15154, 2000.

Bibliography

- [92] T. Mitchell. *Machine Learning*. McGraw-Hill, Singapore, 1998.
- [93] S. Draghici, P. Khatri, A. C. C. Eklund, and Z. Szallasi. Reliability and reproducibility issues in dna microarray measurements. *Trends Genet*, December 2005.
- [94] K. H. Pan, C. J. Lih, and S. N. Cohen. Effects of threshold choice on biological conclusions reached during analysis of gene expression by DNA microarrays. *Proceedings of the National Academy of Sciences*, 102:8961–8965, 2005.
- [95] I. M. Ambros, P. F. Ambros, S. Strehl, H. Kovar, H. Gadner, and M. Salzer-Kuntschik. MIC2 is a specific marker for Ewing’s sarcoma and peripheral primitive neuroectodermal tumor. Evidence for a common histogenesis of Ewing’s sarcoma and peripheral primitive neuroectodermal tumors from MIC2 expression and specific chromosome aberration. *Cancer*, 67:1886–1893, 1991.
- [96] H. Kovar et al. Overexpression of the pseudoautosomal gene MIC2 in Ewing’s sarcoma and peripheral primitive neuroectodermal tumor. *Oncogene*, 45:1067–1070, 1990.
- [97] S. Zhan, D. N. Shapiro, and L. J. Helman. Activation of an imprinted allele of the insulin-like growth factor II gene implicated in rhabdomyosarcoma. *The Journal of Clinical Investigation*, 94:445–448, 1994.
- [98] H. Hahn, L. Wojnowski, K. Specht, R. Kappler, J. Calzada-Wack, D. Potter, A. Zimmer, U. Muller, E. Samson, L. Quintanilla-Martinez, et al. Pached target Igf2 is indispensable for the formation of medulloblastoma and rhabdomyosarcoma. *The Journal of Biological Chemistry*, 275:28341–28343, 2000.

Bibliography

- [99] X. Zhou and K. Z. Mao. LS bound based gene selection for DNA microarray data. *Bioinformatics*, 21:1559–1564, 2004.
- [100] Y. Gao, S. M. Strakowski, S. J. Reeves, H. P. Hetherington, W. J. Chu, and J. H. Lee. Fast spectroscopic imaging using online optimal sparse k-space acquisition and projections onto convex sets reconstruction. *Magnetic Resonance in Medicine*, 55:1265–1271, 2006.
- [101] P. Mitra, C. A. Murthy, and S. K. Pal. A probabilistic active support vector learning algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:413–418, 2004.
- [102] C. Zhang and T. Chen. An active learning framework for content-based information retrieval. *IEEE Transactions on Multimedia*, 4(2):260–268, 2002.
- [103] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.
- [104] C. M. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, 1995.
- [105] J. H. Cho, D. Lee, J. H. Park, and I. B. Lee. Gene selection and classification from microarray data using kernel machine. *FEBS letters*, 571:93–98, 2004.
- [106] R. Tibshirani, T. Hastie, B. Narashiman, and G. Chu. Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proc Natl Acad Sci USA*, 99:6567–6572, 2002.

Bibliography

- [107] J. Li and L. Wong. Identifying good diagnostic gene groups from gene expressin profiles using the concept of emerging patterns. *Bioinformatics*, 18:725–734, 2002.
- [108] U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack, and A.J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci. USA*, 96:6745–6750, 1999.
- [109] D. Singh, P. G. Febbo, K. Ross, D. G. Jackson, and J. Manola. Gene expres-
sion correlates of clinical prostate cancer behavior. *Cancer Cell*, 1:203–209, 2002.
- [110] I. Guyon, A. R. S. A. Alamdari, G. Dror, and J. M. Buhmann. Performance prediction challenge. In *Proceedings of the 2006 International Joint Conference on Neural Networks (IJCNN 2006)*, pages 2958–2965, 2006.
- [111] L. Ein-Dor, I. Kela, G. Getz, D. Givol, and E. Domany. Outcome signature genes in breast cancer: is there a unique set? *Bioinformatics*, 21:171–178, 2005.
- [112] S. Kaski, J. Nikkilä, J. Sinkkonen, L. Lahti, J. Knuuttila, and C. Roos. Associative clustering for exploring dependencies between functional genomics data sets. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2:203–216, 2005.
- [113] J. Huang and C. X. Ling. Using auc and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 17(3):299–310, 2005.

Bibliography

- [114] D. H. Foley. Considerations of sample and feature size. *IEEE Transactions on Information Theory*, 18:618–626, Sep 1972.
- [115] S. J. Raudys. Determination of optimal dimensionality in statistical pattern classification. *Pattern Recognition*, 11:263–270, 1979.
- [116] S. Raudys. On dimensionality, sample size, and classification error of non-parametric linear classification algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:667–671, Jun 1997.
- [117] S. J. Raudys and A. K. Jain. Small sample size effects in statistical pattern recognition: recommendations for practitioners. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:252–267, Mar 1991.