GENE ONTOLOGY DRIVEN FEATURE SELECTION FROM MICROARRAY GENE EXPRESSION DATA

JIANLONG QI

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-33447-8
Our file Notre référence
ISBN: 978-0-494-33447-8

Gene Ontology Driven Feature Selection from Microarray Gene Expression Data

by

© Jianlong Qi

A thesis submitted to the

School of Graduate Studies

in partial fulfilment of the

requirements for the degree of

Master of Science

Department of Computer Science

Memorial University of Newfoundland

June 2007

St. John's Newfoundland

Abstract

Structural and functional data from analysis of the human genome has increased many fold in recent years, presenting enormous opportunities and challenges for machine learning. In particular, gene expression microarrays are a rapidly maturing technology that provides the opportunity to assay the expression levels of thousands or tens of thousands of genes in a single experiment.

In the analysis of microarray gene expression data, one of the main challenges is the small sample size compared with the large number of genes. Among these thousands of genes, only a small number of genes are relevant. To cope with this issue, feature selection, which is the process of removing features not relevant to the labeling, is an essential step in the analysis of microarray data. In this thesis, we present work in this area.

In literature, most of the feature selection methods are solely based on gene expression values. However, due to the intrinsic limitations of microarray technology and a small number of samples, some expression levels may not be accurately measured or they are not a good estimation of the underlying distribution. This can reduce the effectiveness of feature selection. To resolve this deficiency, we explore the possibility of integrating Gene Ontology (GO) into feature selection in this work. GO represents a controlled biological vocabulary and a repository of computable biological knowledge. (Details will be introduced in the subsequent sections.)

The main contributions of this thesis are the following: (1) a statistical assessment of the capability of GO based similarity (semantic similarity) in catching redundancy, and a new similarity measure that takes into account both expression similarity and

semantic similarity, and (2) a method to incorporate GO annotation in the discriminative power of genes, which evaluates genes based on not only their individual discriminative powers but also the powers of GO terms annotating them[1].

[1] These two methods were presented respectively at the 2006 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB 2006) [38] and 2007 ACM Symposium on Applied Computing Special Track on Data Mining (SAC 2007) [39].

Acknowledgments

I would like to thank most of all my supervisor, Dr. Jian Tang, for his continued academic and financial support without which this work would not have been completed. His enlightening discussions and suggestions helped greatly to improve the thesis quality. I would also like to thank Dr. Zhaozhi Fan for his valuable suggestion on statistic and Dr. Yuanzhu Chen for his helpful suggestion on writing thesis. Thanks also go to Miss Elaine Boone for her help on the submission of my thesis. I need to thank the computer support staff of the department for their technical expertise in software and hardware. Finally, my sincere thanks go to my family and friends for their enthusiastic support, kindness and understanding during these busy years.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Introduction to Microarray Technology

Lives are built by cells that contain structural features. The functions of these cells are performed by several types of molecules, such as protein, deoxyribonucleic acid (DNA) and ribonucleic acid (RNA). DNA sequence encodes the complete genetic information for protein synthesis, which consists of three stages: transcription, splicing and translation. In protein synthesis, a gene, which is a strand of DNA molecule in the nucleus, is transcribed to a messenger RNA (mRNA), and then this mRNA is translated to a protein. This entire process, taking the information contained in genes and turning that information into proteins, is called gene expression.

Gene expression levels in a cell are very important for biologists, because they evaluate the state of a cell based on what genes are expressed within it. Microarray technology allows researchers to determine the expression level of a gene by measuring the corresponding mRNA abundance. The mRNA is not the ultimate product of a gene and the correlation between the mRNA and protein abundance in the cell may not be straightforward. The mRNA level, instead of the protein level, is measured due to the following reasons. The absence of mRNA in a cell is very likely to imply a low level of the ended protein. In addition, the measurement of mRNA levels can be done in a high-throughput way and is cheaper than the direct measurement of protein levels [4].

Microarrays exploit the preferential binding of complementary single-stranded nucleic-acid sequence. The basic principle is that the unknown samples are hybridized to an order array of immobilized DNA molecules whose sequences are known [15]. This idea of using a piece of DNA as a probe to determine the presence of the complementary DNA (cDNA) in a solution is evolved from Southern blotting technology, whereby fragmented DNA is attached to a substrate and then probed with a known gene or fragment. Compared with the traditional approach to genomic research, the most attractive advantage of microarray technology is its capability of monitoring the expression levels of tens of thousands of genes in parallel.

A microarray is a small chip (made of chemically coated glass, nylon membrane or silicon), onto which tens of thousands of DNA probes are attached in fixed grids by a robot arrayer using contact or non-contact printing methods. Generally, microarrays are categorized into two groups: cDNA microarrays and oligonucleotide arrays (abbreviated oligo chip). A cDNA microarray simultaneously analyzes two samples, the test sample and the reference sample. In contrast, in oligo chips, the test sample and the reference are separated, and they are analyzed on different chips. In other words, the two samples on a cDNA chip can be viewed as comparable to two samples on 2 oligo chips [48]. Despite differences in the details of their experiment protocols, both types of experiments consist of the following procedures: target preparation, hybridization, scanning process, and normalization [23].

1.1.1 Target Preparation

In cDNA microarray experiments, mRNA samples are extracted from both the test and the reference samples and then synthesized to cDNA by the reverse transcription. Then, the test and reference cDNA samples are differentially labeled with fluorescent dyes or radioactive isotopes. Differential labeling is not necessary for oligo chips. In oligo chips, sample mRNA is first reverse transcribed into single-stranded cDNA. The single-stranded cDNA is then converted to a double-stranded cDNA. Finally, the double-stranded cDNA is transcribed to complementary RNA(cRNA) [48].

1.1.2 Hybridization

In cDNA microarray, the differentially labeled test and reference cDNAs are mixed in equal amounts and hybridized with the arrayed DNA sequences. Each spot on the microarray chip contains enough DNA copies to allow probe hybridization from both samples without interference [15]. In oligo chip, cRNAs are hybridized with the oligodeoxynucleotide probes on the glass slide and then bound to an avidin-conjugated fluorophore.

1.1.3 Scanning Process and Normalization

After hybridization is completed, the intensity of the fluorescence emitted from the labeled and hybridized targets is scanned and digitally imaged. Then, raw signal intensity, either from cDNA or oligo chip, must be adjusted to a common standard (normalized) to correct for differences in overall array intensity that include background noise as well as differences in efficiency in detection and data acquisition. In other words, normalization processes make the results from different experiments comparable [40]. After normalization, the raw gene expression levels are presented as an expression ratio of test vs control sample, or the gene expression profiles from several samples may be compared with a clustering algorithm [48]. This leads to gene expression values that are suitable for statistical analysis.

1.2 Limitation of Microarray Technology

Microarray technology has the potential to greatly enhance our knowledge about gene expression, but there remain challenging problems associated with the acquisition and analysis of microarray data. A main problem is that running the microarray experiment can be technically error prone. As a result, microarray data may contain inaccurate expression levels or missing values.

In [25, 28], the authors show that incorrect cDNA sequences may be attached during the manufacturing of the chips and this could compromise the fidelity of the DNA fragments immobilized to the microarray surface. Moreover, the hybridization process can fail and this results in incomplete information for some spots on the slides.

In addition, microarrays measure the mRNA abundance indirectly by measuring the fluorescence of the spots on the array for each fluorescent, so the raw data produced by microarrays are monochrome images [4]. Transforming these raw images into the gene expression levels is a very complicated process. This process may depend on properties of the hardware such as the scanner, and manual adjustment might be involved.

Another difficulty in acquiring gene expression levels is the identification of spots with the respective genes. In the microarray analysis, it is very difficult to distinguish between genes with a high degree of sequence similarity, because arrays for higher eukaryotes, such as human, are typically based on expressed sequence tag (EST) and linking the EST to the respective gene is complicated [28, 4]. Moreover, it is possible that the same genes are represented by several spots on the array, but measurements from these different spots may differ.

The high experimental cost is another weakness for microarray technology. Although this technology enables us to measure gene expression levels of thousands of genes in a single chip, the cost of a chip is high and consequently the number of samples is very limited compared with the number of genes in a microarray dataset. The asymmetry between the number of samples and that of genes makes the statistical analysis of microarray data a challenging task. How to cope with the small number of samples and inaccurate information in microarray has been the topic of many researches.

1.3 The Analysis of Microarray Data

The result of a microarray experiment is normally organized into a data matrix, where its rows represent the expression levels of the genes among all the samples, while the columns represent sample observations to be analyzed. Hence, there are two straightforward ways to study gene expression matrices [4]. By comparing expression levels of genes in different rows, co-regulated genes can be identified. On the other hand, by comparing samples, genes that are differentially expressed can be found.

Generally, the analysis of microarray data can be divided into two categories: unsupervised and supervised. Unsupervised approaches group together objects (genes or samples) with similar properties. Some main goals of these clusterings include identifying candidate genes and discovering new classes of diseases that may be critical for correct diagnoses and treatment selections.

One of the main goals of supervised expression data analysis is to construct classifiers, which assign predefined classes to a sample. This process is also called sample classification. In a sample classification, each observation is labeled in advance. The labeling contains knowledge of disease subtypes or the tissue origin of a cell type. Classifiers are built from the microarray gene expression data with the purpose of predicting the label of any unknown observation. Because the number of genes is much great than the number of samples, and many genes are not relevant to sample labels in microarray data, feature selection is a necessary pre-step for sample classification. This process removes genes irrelevant to class labels so that classification is more accurate and efficient. In Chapter 3, we will discuss in more detail feature selection in microarrays.

1.4 Benchmark Microarray Datasets

In this section, we will review several benchmark microarray datasets that will be used in this work. These datasets have been widely used in existing work because most expression values in these sets reflect the intrinsic biological characteristics of observed genes.

1.4.1 Leukemia Dataset

Golub [16] introduced the leukemia dataset in 1999. It consists of 62 bone marrow and 10 peripheral blood samples obtained from acute leukemia patients. These samples were extracted from two subtypes of acute leukemia cells: acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML). RNAs prepared from these samples were hybridized to high-density oligonucleotide microarrays and a quantitative expression level was measured for each gene. This dataset is considered to have a good separability between samples from different subtypes of leukemia, since in several works [56, 11], 100% classification accuracy is attained.

1.4.2 Lung Cancer Dataset

The lung cancer dataset is presented in [17]. It contains 181 tissue samples from two types of cancers: malignant pleural mesothelioma (MPM) and adenocarcinoma (ADCA) of the lung. Lung ADCA tumors consist of both primary malignancies and metastatic ADCAs of breast and colon origin. MPM samples contain relatively pure tumors. Distinguishing MPM from ADCA of the lung is challenging from both the clinical and pathological perspective, because patients of these two types of tumors often present a very similar symptom, but the ultimate treatments are very different. In this dataset, because both tumor types have different cell types of origin, the gene expression levels in MPM and ADCA samples vary significantly. This leads to a very high predicative accuracy when the samples are classified by their gene expression levels [17].

1.4.3 Colon Cancer Dataset

Colon cancer dataset [2] consists of colon adenocarcinoma specimens and normal colon tissue obtained from colon cancer patients, and these samples are analyzed with the oligonucleotide arrays. There are 62 samples in the dataset, 40 tumor samples and 20 normal samples. The normal samples include a mixture of muscle and epithelial tissue, while the tumor samples are biased to epithelial tissue of the carcinoma. This difference of tissue composition between the two classes of samples makes the classification by gene expression levels difficult, because tumor samples with high muscle content might be misclassified as normal samples [2]. This indicates the importance of improving tissue purity in the collection of in living organism.

1.4.4 Breast Cancer Dataset

Breast cancer dataset [50] consists of tumor samples from 97 breast cancer patients. For each patient, RNAs are isolated from a snap-frozen tumor within one hour after surgery. These samples are categorized into two groups: relapse and non-relapse. Relapse samples are extracted from patients who developed distant metastases within 5 years after the surgery, while non-relapse samples are extracted from patients who continued to be disease-free after a period of at least 5 years. Chemotherapy reduces the risk of distant metastases, but patients may suffer from the side effects caused by this treatment, so the accurate prediction of metastases is critical for the clinic treatment of breast cancers.

1.4.5 Prostate Cancer Dataset

Prostate cancer dataset is presented by Singh in [47]. This dataset consists of gene expression levels extracted from 52 tumor and 50 non-tumor prostate samples using oligonucleotide microarrays containing 12,600 probes. This dataset has a rather good behavior in term of the separation of different classes, because the gene expression level-based classification attains a predicting accuracy from 86% to 92%. However, this level of accuracy is not sufficient to replace histological examination [47].

1.5 The Research Objective of This Work

One of the main challenges in sample classification of microarray data is the small sample size compared with the large number of features (genes). For a typical dataset, there are 2,000-30,000 genes while the number of samples is in the range of 40-200. When the number of features is large with respect to the sample size, the classifier may only perform well on training samples but not on unseen data [46]. This is referred to as overfitting. To cope with this issue, feature selection becomes an essential step in the analysis of microarray data, because this process removes features that are irrelevant to labeling.

There are a number of advantages of feature selection. Firstly, microarray data sets often contain a significant number of genes whose expression levels are not relevant to class labels. In other words, these genes, i.e. irrelevant genes, are not biologically informative for the classification. It has been proven that including irrelevant genes into the selected feature subset can decrease the effectiveness of classification algorithms, so feature selection improves the accuracy of classifiers [58]. Secondly, feature selection reduces the computational cost of the classification process. Particularly, in high-dimensional problems, such as the analysis of microarray data, it is compulsory to drastically reduce the number of genes to be measured in order to make classification efficient. Finally, feature selection leads to more interpretable results. The fundamental goal of the analysis of microarray data is to identify genes whose expression patterns have meaningful biological relationships with the classification and this can assist in some biological and/or bio-medical processes, such as drug discovery and early diagnosis of diseases. The selected subset of genes produced by feature selection makes the identification process feasible due to the relevance of the genes as well as, presumably, the small size of the subset.

In the literature, most feature selection methods for microarray data are driven by gene expression levels. However, due to the intrinsic limitations of microarray technology, as mentioned in Section 1.2, and the small number of samples, some expression levels cannot be accurately measured or are not sufficient to estimate the underlying distribution. This may reduce the effectiveness of feature selection methods based solely on expression levels. One feasible approach to overcome this deficiency is to apply Gene Ontology (GO). GO is one of the most important ontologies within the bioinformatics community. This ontology defines a shared, structured and controlled vocabulary to annotate molecular attributes across model organisms [13]. In this thesis, we will study methodologies and some underlying theoretic principles for GO-based feature selections.

The main contributions of this thesis are as follows: (1) a statistical assessment of the correlation between GO-based similarity (semantic similarity) and expression similarity, and a new similarity measure that takes into account both expression similarity and semantic similarity, and (2) a method to incorporate GO annotation into the discriminative power of genes, which evaluates genes based on, not only their individual discriminative power, but also the discriminative power of the GO terms annotating them. The novelty of the GO-based method is that it incorporates biological knowledge into the traditional co-relation measurement, and therefore, reduces the likelihood that a gene is related to the labeling incidentally. The effectiveness of our method is demonstrated by application to several widely used datasets.

This thesis includes the following subjects: Chapter 2 reviews the basic concepts in Gene Ontology (GO). It includes its basic structure, the biological nature of annotation, the concept of similarity between GO terms, as well as that between gene products.

At the beginning of Chapter 3, we introduce several classification algorithms that will be used in this work. Then, we review related works of feature selection in microarray. This includes a description of wrapper and filter models, a comparison between these two models, the concept of feature redundancy and two methods to detect redundancy: Markov Blanket and Pearson Correlation Coefficient. In addition, several GO-based feature selection methods are discussed.

Chapter 4 explores the possibility of incorporating GO to remove feature redundancy for microarray data. In this chapter, we demonstrate the intrinsic ability of the GO-based similarity (semantic similarity) in detecting redundancy by using statistical experiments. Furthermore, we propose a strategy to integrate the semantic similarity into the traditional expression similarity. This chapter concludes with experimental results of the proposed method on four public datasets. These results show that feature selection using the new similarity measure leads to higher accuracy.

Chapter 5 presents a method to integrate GO into the discriminative power of genes for feature selection of microarray data. In this chapter, we propose a method that ranks genes by considering not only the individual discriminative power, but also the biological information contained in their GO annotations. We conduct experiments to demonstrate the effectiveness of this method.

In Chapter 6, we conclude the thesis by summarizing the main results.

Chapter 2

Gene Ontology

In this chapter, we will introduce background knowledge about Gene Ontology (GO), including GO annotation and GO-based similarity.

2.1 Gene Ontology Annotation

An ontology is a set of vocabulary terms that label concepts in a domain. These terms should have definitions and be placed within a structure of relationships. Typical examples of relationships are the "is a" relationship between parent and children and the "part of" relationship between part and whole.

The Gene Ontology (GO) is one of the most important ontologies within the bioinformatics community and is being developed by the Gene Ontology Consortium [14]. The primary goal of GO is to define a shared, structured and controlled vocabulary to annotate molecular attributes across model organisms [13]. GO represents a repository of computable biological knowledge and comprises three ontologies: molecular function (MF), biological process (BP), and cellular component (CC). MF represents information on the role played by a gene product. BP refers to a biological objective to which a gene product contributes. CC represents the cellular localization of the gene product, including cellular structures and complexes [52].

GO terms and their relationships are represented by Directed Acyclic Graphs (DAG) where each node except for the root has one or more parent nodes. Any parent surmises the meaning of all its children. No cyclic relationships between terms are allowed. Figure 2.1 presents a part of DAG in GO. Terms near leaves in DAG contain more biological knowledge than those near the root.

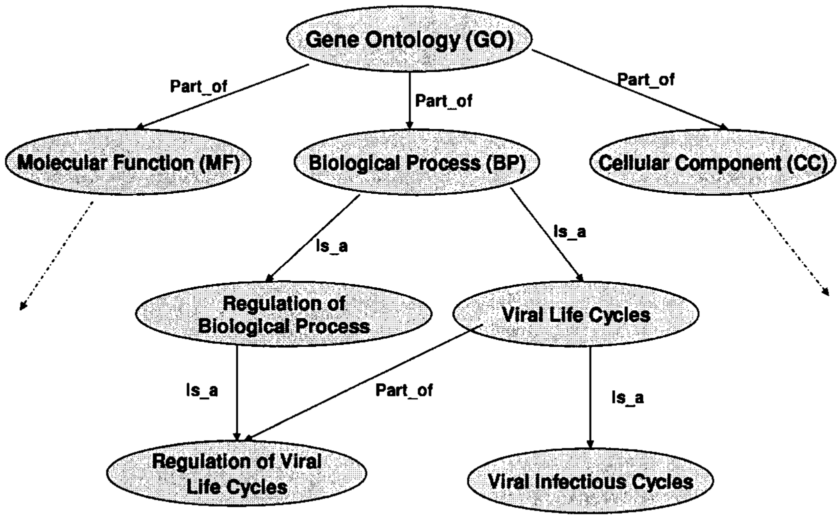


Figure 2.1: A Part of Gene Ontology DAG. Usage explained in text

There are two kinds of relationships between children nodes and parent nodes: "is a" and "part of."The first relationship is used when a child class is a subclass of a parent class. For example, as shown in Figure 2.1, 'Regulation of Biological Process' is a child of 'Biological Process (BP).' The second relationship is used when a child is a component of a parent. For example, 'Regulation of Viral Life Cycle' is part of 'Viral Life Cycle.'

Each gene product can be annotated with a set of GO-terms. For example, if a gene, g_i, is annotated with the GO-term 'Regulation of Viral Life Cycle' in BP, this indicates that g_i participates in the biological process of regulation of viral life cycle. The quality of an association between a gene product and a GO-term is represented by an evidence code. There are 14 types of evidence codes. Typical examples include:

- Traceable Author Statement (TAS): This evidence code is used when annotations are supported by articles or books.
- Inferred from Expression Pattern (IEP): This evidence code is used when annotations are inferred from the timing or location of expressions of genes.
- Inferred from Electronic Annotation (IEA): This evidence code is used for annotations that directly depend on computation without review by curators.

Compared with other types of evidence codes, IEA annotations lack reliability, because IEA is used when no curator has checked the annotation to verify its accuracy. Hence, only associations supported by non-IEA codes are considered in our research. GO annotations are the basis for GO-based similarity, which we discuss next.

2.2 Gene Ontology Based Similarity

An important concept relating to GO is similarities between terms. There are two methods to calculate the similarity, the edge counting model and the information-theoretic model [6]. In the edge counting model, the distances are measured by the number of edges between terms. If there are multiple paths, the shortest or average distance may be used. This model has a weakness in that it assumes that nodes and links are uniformly distributed in an ontology [52]. In the information-theoretic model, the similarity between terms c_i and c_j is calculated by the information carried by their smallest common parent c:

$$sim(c_i, c_j) = -\log(p(c)), \tag{2.1}$$

where $p(c)$ is the probability of finding a child of c in a DAG. This is calculated as:

$$p(c) = \frac{the\ \ number\ \ of\ \ children\ \ of\ \ c}{the\ \ total\ \ number\ \ of\ \ terms\ \ in\ \ the\ \ DAG}. \tag{2.2}$$

The smallest common parent of c_i and c_j is their lowest common ancestor. For instance, in Figure 2.1, the smallest common parent of 'Regulation of Viral Life Cycle' and 'Viral Infectious Cycle' is 'Viral Life Cycle.' As c goes up to the root of the ontology, the value of $p(c)$ and $-\log(p(c))$ monotonically approach 1 and 0 respectively. This indicates that the similarity between c_i and c_j decreases. Hence, the more specific term c is, the more similar these two terms are. A limitation of this model is that it does not take into account the information carried by c_i and c_j. For instance, in Figure 2.2, according to Equation 2.1 we have $sim(B, C) = sim(B, A)$, but intuitively B is more similar to A than it is to C.

Lin [31] proposes a more sophisticated approach, which considers not only the information shared by two terms, but also that owned by themselves. Given two terms c_i and c_j, their similarity is defined as:

$$sim(c_i, c_j) = \frac{|2 \times \log(p(c))|}{|\log(p(c_i)) + \log(p(c_j))|}, \tag{2.3}$$

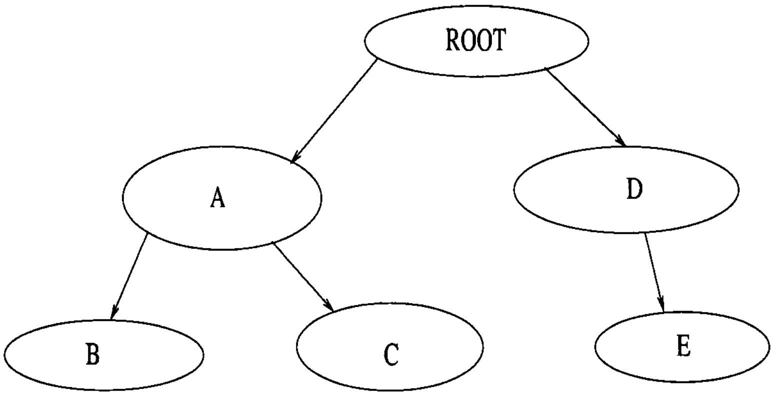


Figure 2.2: GO Similarity

where c is again the smallest common parent of c_i and c_j. This value varies between one and zero since it represents the proportion of the information shared by two terms to the total they have. For a given smallest common parent, the more specific these two terms are, the less similar they are. Jiang [24] reports the semantic distance function, which measures the difference between the shared information and the total information:

$$dis(c_i, c_j) = 2 \times \log(p(c)) - [\log(p(c_i)) + \log(p(c_j))]. \tag{2.4}$$

Given a pair of gene products, g_i and g_j, which are annotated by a set of terms A_i and A_j respectively, where A_i and A_j comprise m and n GO-terms, the GO-based similarity, referred to as *semantic similarity*, is defined as the average inter-set similarity between terms in A_i and those in A_j [52]:

$$Semantic(g_i, g_j) = \frac{1}{m \times n} \times \sum_{c_k \in A_i, c_p \in A_j} sim(c_k, c_p). \tag{2.5}$$

The *semantic distance* between these two gene products, g_i and g_j, is defined as:

$$Distance(g_i, g_j) = \frac{1}{m \times n} \times \sum_{c_k \in A_i, c_p \in A_j} dis(c_k, c_p). \tag{2.6}$$

In this work, we downloaded GO annotations from the GO website (http://www.geneontology.org/) [1]. We collected the GO annotations for gene products from SOURCE [10]. The GO similarity for a pair of genes is calculated from their annotated terms in BP ontology.

[1] The GO annotation database we use in our experiments was published on March 2007.

Chapter 3

Related Work

In this chapter, we first introduce the classification algorithms that will be used in our experiments. Then, we review two typical feature selection models for microarray data, i.e. wrapper and filter. We further introduce the concept of feature redundancy and some typical approaches to detecting it. Last, we present several analytical pieces of work on microarrays using Gene Ontology.

3.1 Classification Algorithm

3.1.1 Naive Bayes

We first introduce Bayes Theorem, which is the basis for Naive Bayes classification algorithm. Let X be a sample observation, represented by a vector of attributes and C be a class. For a classification problem, we want to determine $P(C|X)$, i.e. the probability that the observed data sample X belongs to C. Bayes Theorem [19] states

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)},$$

where $P(C|X)$ is the posterior probability. Bayes Theorem is useful in that it provides a way of calculating the posterior probability using $P(C)$, $P(X)$ and $P(X|C)$, which could be estimated from the training samples.

A Naive Bayes (NB) classifier is based on the Bayes Theorem. Because datasets often contain many attributes, this may lead to an extremely high computational cost in calculating $P(X|C)$. A NB classifier assumes that features are independent of each other for a given class. This assumption is called class conditional independence. Despite this seemingly arguable assumption, it has been shown that NB is comparable in performance with many more complex algorithms, such as decision trees and neural network classifiers.

3.1.2 Decision Trees

A decision tree learning algorithm is a greedy algorithm that constructs decision trees in a top-down recursive manner. A decision tree is a tree structure where non-leaf nodes represent tests on one or more features and leaf nodes reflect classification outcomes. An unknown sample is classified by starting at the root node, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute. A critical task in the construction of a decision tree is how to select the test attribute at each node.

ID3 [41], a well known decision tree induction algorithm, uses an entropy-based measure, known as information gain, as a heuristic for selecting an attribute. An attribute is selected if it best separates the training samples into individual classes. A main limitation of this algorithm is that it requires all attributes to be categorical.

C4.5 [42] enhances ID3 by eliminating this requirement. Let a continuous attribute with values $A_1, A_2, .., A_m$ in increasing order in the training samples. Then, for each value $A_j (j = 1, 2, ..m)$, C4.5 partitions the samples into two subsets, one with values less than or equal to A_j, and the other with values greater than A_j. For each partitioning, it calculates the information gain. It chooses the partition that maximizes the information gain.

3.1.3 Support Vector Machines

Support Vector Machines (SVMs) [7] find hyperplanes in the feature vector space to separate samples in two classes with the maximum boundary. We illustrate the mechanism of SVM with the simplest case: linear machines trained on separable data. Let $\{x_i, y_i\}_{i=1}^{l}$ be the set of l training samples where x_i is a d-dimension vector($x_i \in R^d$) and y_i is the class label ($y_i \in \{-1, 1\}$). SVM solves the following optimization problem: minimizing $\|w\|^2$ subject to

$$x_i \bullet \mathbf{w} + b \geq +1 \text{ for } y_i = +1$$

and

$$x_i \bullet \mathbf{w} + b \leq -1 \text{ for } y_i = -1,$$

where $\mathbf{w} \in R^d$, $\|w\|$ is the Euclidean norm and $b \in R$. The idea of an SVM is illustrated in Figure 3.1. Samples lying on the hyperplane $H_1 : x_i \bullet \mathbf{w} + b = 1$ ($H_2 : x_i \bullet \mathbf{w} + b = -1$) are the closest positive (negative) samples to the separating hyperplane. These samples are called as support vectors. Since H_1 and H_2 are parallel (they have the same normal) and no training samples fall between them, we can find the pair of hyperplanes which gives the maximum margin between H_1 and H_2 by minimizing $\|w\|^2$.

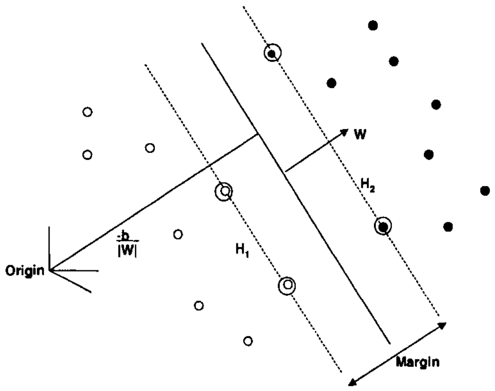


Figure 3.1: Linear Separating Hyperplanes for the Separable Case. The support vectors are circled [7].

Training a support vector machine requires the solution of a very large quadratic programming problem. In [36], the authors propose the sequential minimal optimization (SMO) to solve this problem. This method breaks the large quadratic programming problem into a series of smallest problems and avoids the time-consuming computation. In the case that the samples are not separable in the input space, a kernel function can be used to map the input feature vectors into a higher dimensional space so that they become separable in the used space. Examples of typical kernel functions include linear, Gaussians, polynomials and neural network.

In an SVM, the location of the separating hyperplane is only affected by support vectors. This property makes SVMs robust to noise, since the classification algorithm only focus on the subset of samples that are critical for distinguishing between samples from different classes, ignoring the remaining samples. In [5], the authors show the effectiveness of SVMs in the analysis of microarray data.

3.1.4 Logistic Regression

Logistic Regression (LR) is a generalized linear model, which forms a predictor variable from the linear combination of the feature variables. In a two-class classification, let X be a sample with n attributes $x_1, x_2, ..., x_n$. In addition, p and $1-p$ represent the probability of X representing class 0 and 1, respectively. The normal logistical regression model is

$$\eta = \log \frac{p}{1-p} = \alpha + \sum_{j=1}^{n} \beta_j x_j,$$

where α and $\beta_1, \beta_2, ...\beta_n$ could be estimated by maximum likelihood criterion. η is called the linear predictor and the logarithm is called link function. When the number of attributes is much more than the number of samples, such as in microarray data, a penalty on the sum of the squares of the regression coefficients is introduced. This penalty is called ridge regression [8].

In [45, 61], the authors demonstrate that LR and SVM perform similarly in the classification of microarray data. Compared to SVM, the main advantage of LR is that it provides an estimate of the class probability. This allows LR to explicitly show the prediction strength and reduce the possibility of incorrect classification.

3.2 Feature Selection for Microarray Data

Feature selection methods can be broadly categorized into two groups: filters and wrappers [26]. A wrapper model is coupled with a learning algorithm. It evaluates the goodness of feature subsets by the accuracy of the classifier generated by the learning algorithm. A filter model is independent of any learning algorithm. It selects the optimal feature subset according to the intrinsic characteristics of the features [60]. The wrapper and filter models have been extensively adopted in analysis of microarray data [22, 58].

Since this thesis focuses on the analysis of microarray data, we will discuss the details of these two models for feature selection in the context of microarray data.

3.2.1 Wrapper Model

The general structure of a wrapper model is depicted in Figure 3.2. The feature subset selection algorithm conducts a search for a good subset using a learning algorithm as part of an evaluation function. For each candidate feature set, this learning algorithm trains a classifier, i.e. a hypothesis, used for feature evaluation [26]. The performance of the evaluation is often estimated by the cross-validation accuracy of the classifier when it is trained by this subset. The motivation of this design is that the selected features should depend not only on the features and concept to be learned, but also on the learning algorithm itself [22].

The wrapper model conducts a search in a space where each state represents a feature subset. A complete search in the feature space is not feasible since the number of features (genes) is huge. Hence, a wrapper model must determine the nature of the search process: an initial state, a termination condition, and a search engine.

According to the starting point in the space, searches can be grouped into two categories: forward selection and backward elimination. Forward selection refers to a search that starts from an empty set of features and successively adds features.

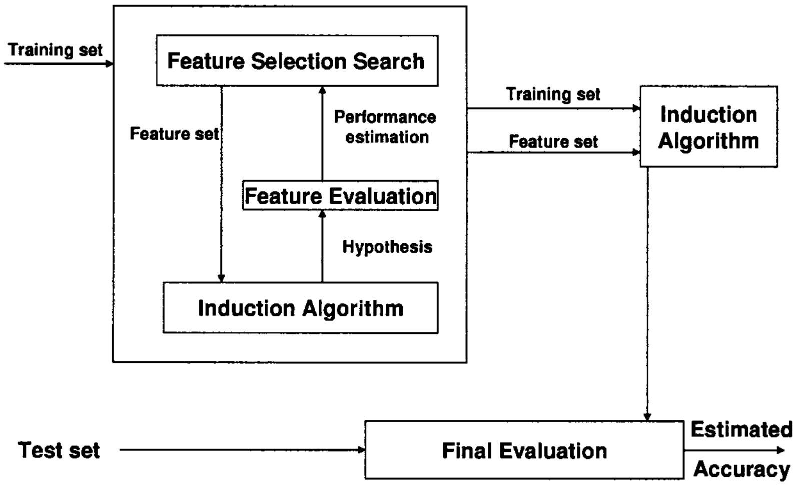


Figure 3.2: The Wrapper Approach to Feature Subset Selection

Backward elimination refers to a search that starts from the full set of features and successively removes features [26]. Since the number of genes can be very large but only a small number of genes are usually needed to discriminate between different classes, forward selection is applied more often.

For the termination condition, a typical criterion is the "non-improvement" of the classification accuracy of any alternative feature subset suggested by the search engine. Another common criterion is to fix a number of possible solutions to be visited along the search.

The search engine decides the strategy of the search. Typical search engines include sequential forward selection (SFS), sequential floating forward selection (SFFS) [37], and best-first search [26]:

- SFS is an iterative greedy process. It starts from the empty set. Then, in each iteration, it adds the feature from the remaining feature set, which maximizes the accuracy of the current set. This process stops when no additional feature improves the accuracy of the current set. The limitation of this algorithm is that it may only attain local maxima.

- SFFS, in contrast to SFS, is not monotonically incremental. After each forward step, SFFS may take several backward steps if these backward steps lead to a better classification accuracy.

- Best-first, similar to SFS, is a forward selection strategy. However, it is more robust and thorough than SFS. The basic idea is to allow the search engine to back-track within limited space. That is, if adding features to the current set, s, does not increase the classification accuracy, the search engine explores a collection C of feature sets as follows. C is defined as the collection of all the children of the feature sets on the path between the root of the search and s, excluding those on the path. The search engine picks the set with the highest classification accuracy in C, denoted by $s^{'}$. If $s^{'}$ has a higher accuracy than s, the search engine resumes its search from $s^{'}$. Otherwise, C is updated by replacing $s^{'}$ with its children. In this case, the search engine again picks the best from C for further processing. The number of repetitions of the above is determined by a parameter to control the extensiveness of the search. Hence, best-first algorithm is more resistant to local maxima than SFS.

Although wrapper models can achieve rather promising results in the analysis of microarray data, a major limitation is that they normally require prohibitive computation, because a classification algorithm is repetitively executed to calculate the accuracy. In addition, since a wrapper model usually uses cross validation repetitively on a single data set, the probability that it finds a feature subset that performs well on the validation data incidentally cannot be ignored [58]. The situation could deteriorate for microarray data, because the number of samples is typically small.

In the study of microarray data, considerable work has been proposed to select informative genes using wrapper models. Here, feature selection is wrapped around with different classification algorithms. In the sequel, we provide two representatives based on forward selection and backward elimination, respectively.

3.2.1.1 An Example of Sequential Forward Selection

In [22], the authors apply SFS to build feature wrappers. This feature selection method is applied on four classification algorithms: NB, C4.5, IB1 and CN2, respectively. NB and C4.5 have been introduced in the previous section. IB1 is a case-based, Nearest-Neighbor classifier [1]. This algorithm predicts the class label of a test sample with the label of the nearest training sample regarding to this test sample. CN2 represents a classification model by a set of **IF-THEN** rules, where the **THEN** part represents the class predicted for samples that match the conditions of the **IF** part [9]. These algorithms are selected because they have completely different approaches to learning and at the same time they all have long standing tradition in the classification history.

To evaluate the performance of SFS for sample classification in microarray data, experiments are performed on several benchmark datasets, including the leukemia and colon cancer data set. The results show that compared to the no gene selection method, SFS notably improves classification accuracies and reduces the number of genes selected.

3.2.1.2 An Example of Backward Elimination

Guyon [18] proposes a wrapper method of gene selection utilizing SVM based on Recursive Feature Elimination (RFE). RFE is an instance of backward feature elimination, which includes an iterative procedure that starts with the full set of features and then removes the feature with the smallest ranking criterion in each iteration. In this work, the ranking value of a gene is determined by the respective weight magnitude in the weight vector of SVM. To reduce the computational time, the authors suggest deleting chunks of features in the first few iterations and then removing one feature at a time when the number of features is less than one hundred.

This RFE-based SVM is applied to classify two benchmark microarray datasets: the leukemia and colon cancer. Compared to the baseline method [16, 2], genes selected by this approach yield better classification performance and are biologically relevant to class label. For example, the RFE-based SVM attains 100% leave-one-out cross-validation accuracy with 2 genes in the leukemia dataset, while the baseline method uses 60 genes to achieve the same result. In addition, the proposed method gets 98% accuracy with 4 genes in the colon cancer dataset, but the highest accuracy for the baseline method is only 86%.

3.2.2 Filter Model

As depicted in Figure 3.3, filter methods are essentially data pre-processing or data filtering models. Features are selected based on the intrinsic characteristics which determine their relevance or discriminative power with regard to the class label [11].

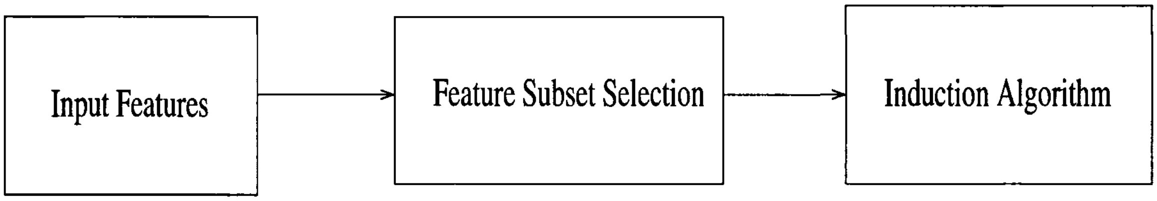


Figure 3.3: The Filter Approach to Feature Subset Selection

The main advantage of filter approaches compared with wrapper.models is that they can be computed easily and efficiently. In addition, filter models do not depend on the classification algorithms, and therefore have better generalization capabilities. However, filter model evaluates gene in isolation without considering correlations between genes. As a result, it is possible that genes in the selected feature subset are highly correlated with each other. This high correlation leads to feature redundancy that could deteriorate the effectiveness of classification. We will discuss feature redundancy in more detail in Section 3.2.3.

In filter model, genes are selected based on the individual discriminative power of genes, so how to accurately determine discriminative power is important. In the following, we will review several criteria to measure this power.

3.2.2.1 Signal-to-Noise Statistic and *t*-statistics

Golub [16] suggested an evaluation of genes by their signal-to-noise statistic values. Given two labels to the sample observations, the signal-to-noise statistic value of a gene is defined as:

$$P(g) = \frac{[\mu_1(g) - \mu_2(g)]}{[\sigma_1(g) + \sigma_2(g)]}, \tag{3.1}$$

where $[\mu_1(g), \sigma_1(g)]$ and $[\mu_2(g), \sigma_2(g)]$ denote the means and standard deviations of the logarithms of the expression levels of gene g for the samples in class 1 and class 2 respectively. This value measures the between class distance in standard deviation.

Another similar measurement is t-statistics [32]. Given two labels to the sample observations, the t-statistics value of a gene is defined as:

$$T(g) = \frac{|\mu_1(g) - \mu_2(g)|}{\sqrt{\frac{\sigma_1(g)^2}{n_1} + \frac{\sigma_2(g)^2}{n_2}}}, \tag{3.2}$$

where n_1 and n_2 denote the numbers of samples in class 1 and 2, respectively. The other variables are defined as Equation 3.1. In [11, 32], t-statistics is applied to analyze microarray data.

3.2.2.2 Information Gain

One of the most widely used feature ranking models is information gain which measures the number of bits of information obtained for class prediction by knowing the value of a feature [34]. Let $\{c_i\}_{i=1}^m$ denote the set of classes and U represent the set of possible values for feature f; the information gain of f is calculated as:

$$G(f) = -\sum_{i=1}^{m} P(c_i) \log P(c_i) + \sum_{u \in U} \sum_{i=1}^{m} P(f = u) P(c_i | f = u) \log P(c_i | f = u), \tag{3.3}$$

where $P(c_i)$ is the probability that an arbitrary sample belongs to class c_i and $P(c_i | f = u)$ is the corresponding conditional possibility when f has the value of u. The value of $G(f)$ represents the expected reduction in the entropy for the labeling caused by knowing the value of the attribute of f. To calculate the information gain, the numeric values of gene expression levels are required to be discretized. This is typically achieved via the entropy-based discretization method [12]. In [30], the authors demonstrated the effectiveness of this discretization model in microarray data.

3.2.2.3 χ^2-statistic

Another common filter model is the $Chi - Squared(\chi^2)$ method which measures the lack of independence between features and class labels [33]. This method also requires numeric features to be discretized first. The χ^2 value of a feature is calculated as:

$$\chi^2 = \sum_{i=1}^{m} \sum_{j=1}^{k} \frac{(A_{ij} - E_{ij})^2}{E_{ij}},$$

where m is the number of intervals, k is the number of classes, A_{ij} is the number of samples in the i^{th} interval, j^{th} class, R_i is the number of samples in the i^{th} interval, C_j is the number of samples in the j^{th} class, N is the total number of samples, and E_{ij} is the expected frequency of A_{ij}. This is calculated as :

$$E_{ij} = \frac{R_i \times C_j}{N}.$$

In [32], the authors applied this model for feature selection in microarray data.

3.2.2.4 Relief-F

The Relief-F algorithm is another popular approach to calculate the discriminative power of a gene. A key idea of Relief-F is to estimate the quality of features according to how well their values distinguish between instances that are near to each other [43]. For this purpose, Relief-F draws instances at random. Furthermore, given a randomly selected instance R, this algorithm searches for its two nearest neighbors: one from the same class, called the nearest hit H, and the other from the different class, called the nearest miss M. Then, Relief increases the quality estimation of a feature if instance R and M have different values of this feature, since it separates two instances with different class values. On the other hand, Relief-F decreases the quality estimation of this feature if the values of this feature of instance R and H are different, since this feature separates two instances at the same class.

In [55], the authors apply Relief-F to select informative genes for microarray data. Experimental results suggest that the performance of Relief-F is comparable with Information gain and $Chi - Squared(\chi^2)$.

3.2.3 Feature Redundancy

One common approach to filter models for microarray data is to select the top-ranked genes, where this ranking is normally based on the individual discriminative power of genes. The problem with this approach is that it evaluates genes in isolation without considering correlations between genes. However, in feature selection, it has been recognized that the combination of two highly ranked features does not necessarily lead to a better feature subset because it is possible that these two features are redundant. The redundancy between two features is signified by the fact that the class-discriminative power of either one will not change much if the other is removed.

Redundancy among selected feature subsets can lead to two problems. Redundancy may affect the efficiency of the classification algorithm, since the dimensionality of the selected gene set increases. In addition, a gene subset with redundancy has a less comprehensive representation of the targeted classes than one of the same size without redundancy [60].

In the following, we will review several works, which apply Markov Blanket and Pearson correlation coefficient to detect feature redundancy in microarray data.

3.2.3.1 Markov Blanket

Koller and Sahami [27] introduced Markov Blankets to detect redundancy. This method essentially is a stronger version of conditional independence from classical statistics. Given a feature F_i and class label C and $M_i \subset F$ $(F_i \notin M_i)$, M_i is said to be a Markov Blanket for F_i iff

$$P(F - M_i - \{F_i\}, C | F_i, M_i) = P(F - M_i - \{F_i\}, C | M_i).$$

The optimal feature subset can be constructed by a backward elimination process in which unnecessary features are removed one by one, because this process guarantees that a feature removed in an earlier phase will still find a Markov Blanket in any later phase.

In [60, 58], the authors use Markov Blankets to remove redundancy in gene selection. One common characteristic of these two works is that they both adopt heuristic approximation in calculating Markov Blankets. This is caused by the following two reasons. First, a complete search is not computationally affordable in microarray data since the number of features is enormous. Second, the full population is required for the optimal subset. However, microarray data just provides a limited number of training samples which is only a portion of the full population.

3.2.3.2 Minimum Redundancy Maximum Relevance

Another widely used measure for detecting redundancy is Pearson correlation coefficient, referred to as *expression similarity*. Given two genes g_i and g_j, the Pearson correlation coefficient is defined as:

$$Pearson(g_i, g_j) = \frac{\sum_{d=1}^{p} (g_{id} - \mu_{g_i})(g_{jd} - \mu_{g_j})}{\sqrt{\sum_{d=1}^{p} (g_{id} - \mu_{g_i})^2} \sqrt{\sum_{d=1}^{p} (g_{jd} - \mu_{g_j})^2}},$$

where μ_{g_i} and μ_{g_j} are the means for g_i and g_j respectively and p is the dimension.

In [11], the authors use expression similarity to measure the redundancy among genes and proposed the minimum redundancy - maximum relevance (MRMR) approach to selecting the optimum feature set. Let S be the feature subset we are looking for. The condition of maximum relevance to classification in S can be defined as

$$maxV, \quad V = \frac{1}{|S|} \sum_{g_i \in S} D_{g_i},$$

where D_{g_i} is the discriminative power of gene g_i. Many models can be used to determine this power, such as methods presented in Section 3.2.2. Furthermore, the minimum redundancy condition is defined as

$$minW, \quad W = \frac{1}{|S|^2} \sum_{g_i, g_j \in S} Pearson(g_i, g_j).$$

In order to optimize these two objective functions simultaneously, the authors use two combined criteria as either:

$$max(V - W)$$

or

$$max(V \div W).$$

Because the exact maximization process requires extremely high computational cost, the linear incremental search algorithm, i.e. a forward, hill climbing search strategy, is used to generate a sub-optimal solution in the experiment. Experiments indicate that the features selected by MRMR lead to higher accuracy than those features selected by evaluating relevance only.

3.2.3.3 HykGene

In [56], the authors propose a clustering based method to remove redundancy (HykGene), where redundancy among genes are measured by expression similarity. Figure 3.4 depicts the workflow of this algorithm. This method first selects the top-ranked genes and then applies the hierarchical clustering (HC) algorithm on these pre-selected genes. Measuring the homogeneity between genes by their expression similarities, HC builds a dendrogram. Furthermore, clusters are extracted by analyzing the dendrogram and then this method collapses each cluster into one representative gene. A representative from a cluster is the gene with the minimum sum of squares of dis-

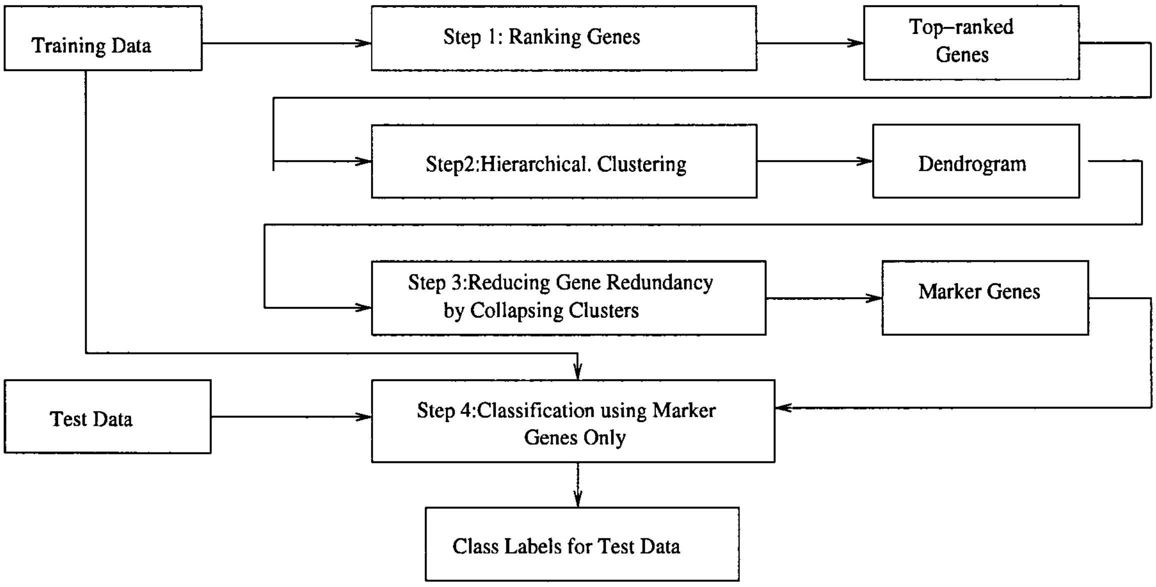

Training Data
Step 1: Ranking Genes
Top–ranked Genes
Step2:Hierarchical. Clustering
Dendrogram
Step 3:Reducing Gene Redundancy by Collapsing Clusters
Marker Genes
Test Data
Step 4:Classification using Marker Genes Only
Class Labels for Test Data

Figure 3.4: The Workflow Diagram of HykGene

tances to all other genes in this cluster. Last, these representative genes form the selected gene subset. One novel point of this method is that the best number of clusters is determined by the accuracy of the classification algorithm on training data. A similar approach is introduced by Hanczar [20].

3.2.4 Integrating Gene Ontology into the Analysis of Microarray Data

In spite of the enormous potential of microarray technology, there remain challenging problems associated with the analysis of microarray data. We have identified these problems in Section 1.2. These problems could lead to inaccurate information in microarray data. As a consequence of imprecise gene expression data and missing values, the effectiveness of gene selection methods, which determine discriminative power and catch redundancy using only expression values, can diminish.

Hence, it is reasonable to conjecture that, should a more precise measurement be used, these methods would be more efficient. However, due to the high cost of re-experimenting, it may be too costly to solve this problem by improving microarray technology itself. As a cost-effective and practically feasible alternative, some methods have explored the possibility to alleviate the above problems by incorporating biological knowledge, such as Gene Ontology, into the feature selection process. Work in this direction will be the main theme of this thesis.

The Gene Ontology (GO) is an important knowledge resource for biologists and bioinformatics. GO annotations have been incorporated into microarray data analysis for various purposes: in the context of cluster validation, missing value estimation and detecting false informative genes. On the other hand, gene expression values are used to predict the participation of genes in GO biological processes. In the following, some works in these directions are reviewed.

3.2.4.1 Correlation between Semantic and Expression Similarities

One important concept in GO is the semantic similarity. Many researches focus on the correlation between the semantic and expression similarities because such a correlation is the basis for integrating GO to feature selection for microarray data. In [52], the authors study the interplay between the semantic and expression similarities for the yeast dataset. In this work, GO annotations are restricted to non-IEA annotations and the semantic similarity is based on Equation 2.5. A strong correlation between these two similarity measures is observed therein. The experimental results show that, in general, high semantic similarity values are associated with high expression similarity values and that low semantic similarity values are associated with low expression similarity values.

In addition, Wang [53] first clusters genes hierarchically using the expression similarity. Then, the semantic similarity is used to assess the biological soundness of these obtained clusters in the yeast dataset. The clusters exhibiting stronger expression similarity values tend to have higher semantic similarity values. In other words, genes in the same clusters are very likely to participate in the same biological process. This agrees with previous results [52] suggesting that these two expression measures are highly correlated. In contrast, GO-driven hierarchical clustering is also applied, where the similarities between genes are measured by the semantic similarity. In general, the obtained clusters match the clusters using the expression similarity. A major advantage of the GO-based clustering is that it results in more biologically meaningful clusters because it can detect relevant functional relationships that may not be represented by the expression similarity [53].

In [54], the study of the correlation between the semantic and expression similarities is extended to mouse, a multi-cellular organism. Overall, strongly co-expressed genes tend to exhibit higher semantic similarity values than weakly co-expressed genes. However, this correlation is much weaker than that in the yeast data. For example, a large number of strongly co-expressed genes have relatively small semantic similarity values. The authors believe that this discrepancy may be due to the complexity of the functional annotation on multi-cellular organisms. Since the biological processes in multi-cellar organisms are far more complicated than those in single-cellular organisms, functional annotations and our current understanding about multi-cellar organisms are not as precise as that about single-cellular organisms.

3.2.4.2 Predicting GO Biological Process from Gene Expression Patterns

In [29], the authors predict participation of genes in GO biological process from gene expression patterns. The experiment is performed in a dataset that describes the transcript levels of 497 genes during the first 24 hours of the serum response in serum-starved human fibroblasts. 284 of these 497 genes have GO annotations. At first, rules between gene expression patterns and the involvement of genes in GO biological processes are generated from annotated genes. Then, these rules are applied to predict participations of those unannotated genes in GO biological processes. The authors demonstrate that many biological process roles, hypothesized for unannotated genes by these rules, agree with assumptions based on gene sequence homology information.

3.2.4.3 Identifying Functional Classes of Genes with GO

In [35], the authors treats each GO term as a 'gene class.' Then, they use ANOVA (Analysis Of Variance between groups) to measure the statistical significance of the expression pattern of each gene with regard to distinguishing samples from different classes. The significance score of a GO term is calculated as the average of the ANOVA value of genes annotated with this term.

The experiments show that the high scores tend to be given to terms, which are highly relevant to the biological knowledge that class labels represent. For example, in a tumor dataset where tumors fell into two groups depending on whether they were derived from T-cells or B-cells, the GO term with the highest score is "T-cell receptor." This result suggests that the informative nature of genes with regard to sample classification is positively associated with their GO annotations. Consequently, this correlation indicates the feasibility to incorporate GO annotations into determining the discriminative power of genes.

3.2.4.4 Improving Missing Value Estimation with GO

Microarray data often contains missing values. The missing values in the expression levels of a gene are often estimated (or imputed) by the expression values of genes close to it. Imputation algorithms normally use Euclidean distance $d(g_i, g_j)$ to measure the distance between genes g_i and g_j. In [49], the authors apply the GO annotations in the imputation algorithms to guide the gene selection processes, so that the set of genes selected for predicting the missing value of a gene are close, not only in their expression values, but also in their functionalities.

The semantic dissimilarity of two terms c_1 and c_2 is measured by the information content $p(c)$ of their smallest common parent c, which is defined as Equation 2.2. The semantic dissimilarity of two genes g_i and g_j, i.e. $s(g_i, g_j)$, is defined as the average inter-set dissimilarity between terms assigned to them. The authors combine the semantic dissimilarity $s(g_i, g_j)$ and the expression-level-based distance $d(g_i, g_j)$ to generate the conjunctive distance $c(g_i, g_j)$, which is defined as:

$$c(g_i, g_j) = s(g_i, g_j)^{\alpha} \times d(g_i, g_j).$$

The positive weight parameter α determines the contribution of the semantic dissimilarity in the combined distance. In this model, only when the value of $s(g_i, g_j)$ is small, which indicates that g_i and g_j are semantically close to each other, their combined distance $c(g_i, g_j)$ is remarkably reduced from the expression-level-based distance $d(g_i, g_j)$, accordingly. This design ensures that only the most specific GO terms (small semantic dissimilarity values) have a significant effect on the conjunctive distance [49], because general GO terms are not informative enough to reflect the biological roles of genes.

The combined distance is applied in the k-nearest neighbor (kNN) and local least squares (LLS) imputation algorithms. The experimental results demonstrate that this incorporation improves the accuracy of the estimation of missing values.

3.2.4.5 Detecting False Informative Genes by Incorporating GO

Due to the limited number of samples in microarray, the authors experimentally demonstrate that even randomly generated expression levels may result in high discriminative scores in [59]. To alleviate this situation, the authors apply GO annotations to remove noisy data. The following definitions are given for this algorithm.

- ***Informative Genes*** are those genes having discriminative scores greater than θ, i.e. $F(g) > \theta$, assuming F is a single-gene-based discriminative score.

- ***Discriminative Power*** of a GO term is defined as the percentage of informative genes among all genes that are annotated with this GO term, i.e.,

  $$DP(go) = \frac{|\{g|g \in go \wedge F(g) > \theta\}|}{|go|}.$$

  Here $g \in go$ denotes that a gene g is annotated by the GO term go and $|go|$ denotes the number of genes that are annotated by the GO term go.

- ***Informative GO Term*** is defined as a GO term go whose discriminative power is larger than γ and the number of informative genes annotated with go is larger than β, i.e.,

  $$DP(go) > \gamma \quad \text{and} \quad |\{g|g \in go \wedge F(g) > \theta\}| > \beta.$$

The authors propose that genes relevant to class labels should not only have high discriminative values, but also be annotated by informative GO terms. Hence, if an informative gene is not annotated by any informative GO term, this indicates that the high discriminative value of this gene may be generated by noise.

Chapter 4

Integrating GO Annotation into Similarity Measures

In this chapter, we introduce a GO-based model for similarity measure. In the following sections, we first statistically assess the correlation between the semantic and expression similarities where the expression similarity is widely used for detecting feature redundancy in microarray data. This correlation enables us to apply the semantic similarity to detect redundancy. Then, we describe a method to combine the semantic and expression similarities to form a conjunctive similarity. Finally, we evaluate the effectiveness of this measure by applying it to two well-known feature selection methods.

4.1 Correlation between Semantic and Expression Similarities

In [52], the authors show that, for the yeast data, the semantic similarity is positively related to the expression similarity. Is this also the case in a general expression data? While it may be hard to obtain a definite answer to this question, we use a statistical framework to give a plausible argument on another widely used data set, the leukemia data set [16]. In addition, rather than examining the question separately, we put it in a context of relevance to class labeling. This may give us some insights into the biological nature of the interplay between the semantic similarity and relevance. In this section, we first introduce a measure for the goodness of a feature subset. Then, we use this measure to assess the correlation between the expression and semantic similarities statistically.

4.1.1 Goodness Measure of Feature Subset

Given a feature subset S and a class labeling C, the relevance of S with respect to C can be defined as

$$\frac{1}{|S|}\sum_{g_i \in S} D_{g_i},$$

where D_{g_i} is the discriminative power of gene g_i with regard to C. In this thesis, we use the information gain defined in Equation 3.3. The numeric values of gene expression levels are quantized by the entropy-based discretization method [12] before they are used to calculate the information gain. The redundancy among the features in S is defined as

$$R(S) = \frac{1}{|S|^2} \sum_{g_i, g_j \in S} Pearson(g_i, g_j).$$

In an optimal feature subset, features should be "minimally similar" to each other and "maximally relevant" to the class labeling. Hence, the goodness of S is defined as the quotient between the relevance and redundancy:

$$Quo(S) = \frac{\frac{1}{|S|} \sum_{g_i \in S} D_{g_i}}{\frac{1}{|S|^2} \sum_{g_i, g_j \in S} Pearson(g_i, g_j)}. \tag{4.1}$$

4.1.2 Statistical Assessment of Correlation between Expression and Semantic Similarities.

In this subsection, we assess the correlation between the expression and semantic similarities statistically. The experiment is performed on the Leukemia data set (Section 1.4).

Because we intend to find out how the semantic similarity plays a role in identifying redundancy, we expect to see a reasonable intensity of redundancy of the genes in the data that we will be working on. In order to do that, we select the top t genes according to their information gain. Because of their high relevance to the class labels, we expect many of them are redundant. We then use the hierarchical clustering algorithm[1] to partition these genes into k clusters using the semantic similarity. Then we select a representative from each cluster. A representative is the gene with the minimum sum of squares of distances to all other genes in the cluster. The goodness of the selected feature subset is assessed by the quotient between the relevance and redundancy defined in Equation 4.1. In order to evaluate this feature subset, we

[1] Similar to the one used in HykGene [56], where the traditional expression similarity is used.

compare it with feature subsets selected randomly. Assuming that the goodness values for randomly selected genes follow the normal distribution, we formulate a null hypothesis and an alternative hypothesis as

$$H_0 : \mu_g \geq GO_g \tag{4.2}$$

and

$$H_1 : \mu_g < GO_g, \tag{4.3}$$

where μ_g represents the mean of the population for the goodness value of the gene sets selected randomly, and GO_g represents the goodness value of the gene set selected by the method mentioned above. The null hypothesis claims that the mean of the goodness value of a randomly selected set of genes is at least as good as that of the genes selected by GO-based clustering. This would suggest that the semantic similarity has no correlation with the expression similarity. If the test strongly suggests otherwise, then the two types of similarity measures have a non-negligible correlation. Consequently, the ability of the semantic similarity in detecting redundancy would be established.

The choice of the number of the top genes t is somewhat arbitrary. In our case, we choose 100, since this value is well above the number of the most informative genes reported in the current literature, yet small enough to speed up the process. Among these 100 genes, 35 of them do not have any GO annotation. To cope with this problem, we first cluster the remaining 65 genes into 8 groups by the semantic similarity-based clustering algorithm. (The reason to have 8 clusters is that this is roughly the number of genes that are most informative for the leukemia data set.) Then, we add each gene without a GO annotation to a cluster with which it has the smallest average expression similarity.

We assume that the goodness values for randomly selected genes follow a normal distribution. To verify this, we randomly select 8 genes from the top 100, and then calculate their goodness value. This experiment is repeated 50 times and consequently results in 50 goodness values. Figure 4.1 shows the probability plot of the goodness values[2]. In this figure, the values of the horizontal and vertical axis for each point are the sample goodness value and the corresponding percentile in the standard normal distribution, respectively. Since using $\Phi^{-1}\left(\frac{i}{n}\right)$ leads to $\Phi^{-1}(1) = \infty$ for the sample with the largest goodness value, the percentiles are approximated by $\Phi^{-1}\left(\frac{i-0.3}{n+0.4}\right)$, where $n = 50$ is the number of samples and i is the rank of each sample, i.e, $i = 1$ for the smallest and $i = n$ for the largest goodness value [3]. We observe that these plotted points roughly fall into the vicinity of the fitted line. This suggests that the probability distribution under test is fairly close to a normal distribution.

Furthermore, to assess the normality more objectively, we perform the Wilks-Shapiro test (W test) [44], based on measures of the linear correlation in the probability plot. In contrast to the probability plot, the W test is a formal procedure, which formulates the null and alternative hypotheses as: the samples are drawn from a normal distribution and the samples are not drawn from a normal distribution.

Let n denote the number of the observations and $Y' = \langle y_1, y_2, ..., y_n \rangle$ denote a vector of the ordered random observations, where y_i is the i^{th} smallest value. We use $\bar{y}$ to denote the sample mean. In addition, let $M' = \langle m_1, m_2, ..., m_n \rangle$ denote the expected values of the standard normal order statistics for a sample of size n, and $V = (v_{ij})$ be the corresponding covariance matrix of M. In other words, we let

[2] The probability plot is generated with Minitab.

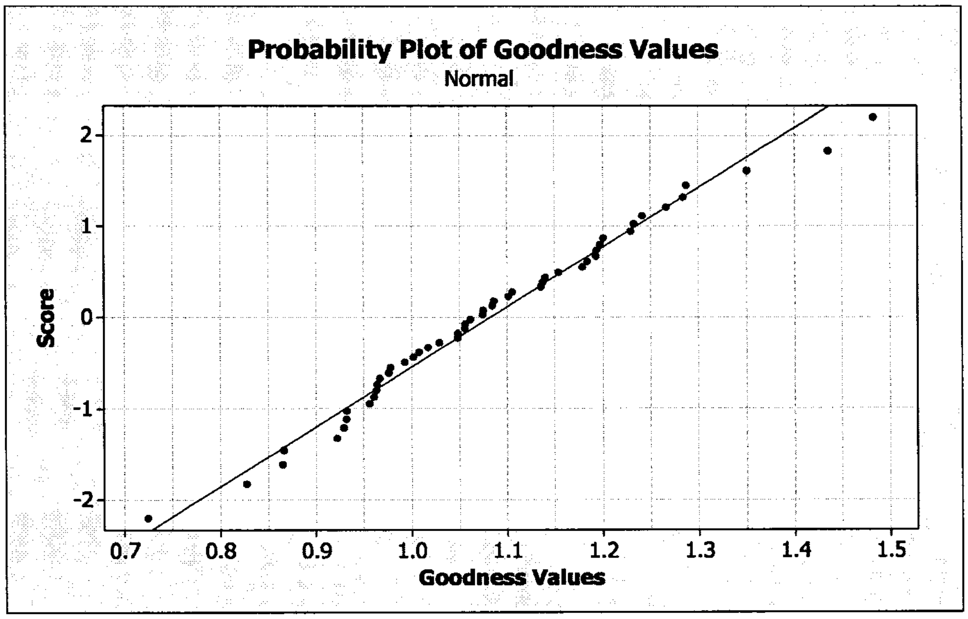


Figure 4.1: Probability Plot of Goodness Values

$x_1 \leq x_2 \leq ... \leq x_n$ be the n ordered observations from a standard normal distribution. Then we have $E(x_i) = m_i$ and $cov(x_i, x_j) = v_{ij}$ $(1 \leq i \leq n)$ [21].

The W test statistic is defined as:

$$W = \frac{(\sum_{i=1}^{n} a_i y_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2},$$

where

$$A' = \langle a_1, a_2, ..., a_n \rangle = M'V^{-1}\Big[M'V^{-1}V^{-1}M\Big]^{-1/2}.$$

W may be thought of as the squared correlation coefficient between the ordered sample values and A'. The value of W indicates the departure from normality, so the smaller the W value, the larger the departure.

Generally, if the p-value of the W test is more than 0.05, the null hypothesis is accepted, indicating that the underlying distribution is a normal distribution. In this experiment, we obtain $W = 0.985$ and the p-value is 0.78. This result strongly demonstrates that the probability distribution under test is normal[3]. Thus, the hypothesis test with regard to Equations 4.2 and 4.3 is a lower tail one sided t-test.

In order to detect the significant difference between μ_g and GO_g, we build a region of indifference. If the difference between μ_g and GO_g is not more than a threshold value T_d, we would consider that they are similar. Hence, the null and alternative hypotheses become

$$H_0 : \mu_g \geq GO_g - T_d$$

and

$$H_1 : \mu_g < GO_g - T_d.$$

The mean and standard deviation of these 50 samples are 1.08274 and 0.15225, respectively. The value of T_d is determined as the 5% of the sample mean which is 0.054. Since the goodness value of the GO-based feature subset is 1.188, the P-value with 49 degrees of freedom is 0.011. This result strongly suggests that H_0 is false. Thus, there is a non-trivial correlation between the semantic and expression similarities. Consequently, we demonstrate the ability of the semantic similarity in detecting redundancy.

[3] Wilks-Shapiro test is calculated with Dataplot.

4.2 A Strategy for Feature Selection using GO

4.2.1 A New Similarity Measure

In the previous section, we showed that GO provides an effective mechanism for detecting redundancy from relevant genes. Does this mean that GO is superior to other similarity measures in all cases? Our preliminary experiments, however, indicate that GO is not necessarily better by itself. Our further experiments show that the best result is attained when GO is used conjunctively with some existing similarity measures, such as Pearson co-expression coefficient, Euclidean distance, etc. We now describe a way to incorporate the semantic and expression similarities. We first normalize their values in the following way. Suppose that we select the top m ranked genes. Let $Semantic(g_i, g_j)$ and $Expression(g_i, g_j)$ denote the semantic and expression similarities between genes g_i and g_j, respectively. We calculate the semantic (expression, resp.) similarity for each pair among the top m genes, resulting in $\frac{m\times(m-1)}{2}$ semantic (expression, resp.) similarity values. Then, we take the average of these $\frac{m\times(m-1)}{2}$ values, and denote it by $mean_{(semantic)}$ ($mean_{(expression)}$, resp.). Also, we calculate the variance, and denote it by $variance_{(semantic)}$ ($variance_{(expression)}$, resp.). Then, we define the following normalized similarities:

$$Semantic_{(norm)}(g_i, g_j) = \frac{Semantic(g_i, g_j) - mean_{(semantic)}}{variance_{(semantic)}}$$

and

$$Expression_{(norm)}(g_i, g_j) = \frac{Expression(g_i, g_j) - mean_{(expression)}}{variance_{(expression)}}.$$

The purpose of the above normalization is to convert these two similarities to the same scale. The *conjunctive similarity* between g_i and g_j is defined as:

$$\begin{aligned} ConjSim(g_i, g_j) &= \alpha \times Semantic_{(norm)}(g_i, g_j) \\ &\quad + (1-\alpha) \times Expression_{(norm)}(g_i, g_j). \end{aligned} \tag{4.4}$$

In the above equation, the parameter α, called *GO weight*, is in the range of [0,1]. We discuss how it is determined in Section 4.2.2.

We can also normalize the semantic (expression, resp.) distance. The process is identical to that for the semantic (expression, resp.) similarity, except that the mean and variance are calculated over $\frac{m\times(m-1)}{2}$ semantic (expression, resp.) distance values. (For the definition of semantic distance, refer to Equation 2.6. An expression distance between genes g_i and g_j is understood as one minus the expression similarity between them.) We can define a *conjunctive distance* between g_i and g_j as follows:

$$\begin{aligned} ConjDis(g_i, g_j) &= \alpha \times SemanticDis_{(norm)}(g_i, g_j) \\ &\quad + (1-\alpha) \times ExpressionDis_{(norm)}(g_i, g_j), \end{aligned} \tag{4.5}$$

where $SemanticDis_{(norm)}(g_i, g_j)$ and $ExpressionDis_{(norm)}(g_i, g_j)$ denote the normalized semantic distance and expression distance, respectively.

4.2.2 About the Value of α

A conjunctive similarity represents a convex combination of the normalized semantic and expression similarities. A crucial task in the proposed similarity measure is to determine an appropriate value for α. Let D be a microarray data set. Then, theoretically, selecting the best value for α is equivalent to solving the following optimization problem:

$$\begin{aligned} &\text{Optimize} \quad && O(ConjSim(*, *|\alpha, D) \\ &\text{subject to} && 0 \leq \alpha \leq 1 \end{aligned}$$

where $O(ConjSim(*, *|\alpha, D)$ represents a metric to evaluate the biological soundness of similarity measures. In the general case, the objective function depends on a similarity measure and a data set, i.e., $O(M, D)$.

In practice, the objective function $O(M, D)$ is normally approximated by running a sequence of algorithms that take the similarity measure as the input, and generate an output value as the metric value. That is, we first run a feature selection method on a given data set and output a feature subset based on the similarity measure. Then, we use a learning algorithm to generate a hypothesis, i.e., a classifier, based on the selected feature set. Last, we test the hypothesis using the leave-one-out cross validation. The test result, i.e. the classification accuracy, is then used as the value for the metric. Since it is well known that each classifier can be biased in a different way, for a data set these optimal values of α may vary considerably.

As defined in the above objective function that takes the microarray data set as a parameter, we speculate that the intrinsic characteristics of microarray data sets, i.e. the quality of expression values, may affect the value of α. In other words, the optimal values of α in different data sets may differ. The quality of expression values in a data set is decided by the number of outliers, missing values, background noise, non-specific hybridizations, etc, associated with the microarray manufacturing process. If the expression values in a data set show a high quality, implying that expression values well match with the biological natures of the corresponding genes, then the expression similarity can accurately measure the similarity between genes by itself. Thus, the conjunctive similarity should be very close with the expression similarity, implying the best value of α in this data set is approximately 0.

On the other hand, if the quality is not good, then the semantic similarity is required to contribute more to the conjunctive similarity. Thus, we have $\alpha > 0$. The more precise value depends on how much it should contribute. With the current status of GO annotations, some genes are only annotated with very general terms, which are not adequate and specific enough to identify their precise biological roles. This may be due to the complexity of the functional annotations as we discussed in Section 3.2.4.1. Our perception is that semantic similarity values calculated based only on these general GO terms are not informative in terms of revealing the true biological roles of the genes annotated. Accordingly, the semantic similarity values with such a characteristic are unlikely to improve the objective function value. The point here is: the more specific for the GO term, the more likely the semantic similarity values based on them will improve the objective function value. (This is indeed a general statement. For specific data sets, and specific objective functions, it may be that semantic similarity values that are based on the GO terms up to certain levels can improve the objective function.) On the other hand, conjunctive similarities based on the α values of different magnitudes will be compatible with the semantic similarities on GO terms up to different generality. The larger the α, the more general the GO terms are up to. When α is close to 1, the conjunctive similarity will be compatible with the semantic similarity values on GO terms up to the root level, and when α is near 0, it is compatible only with the GO terms close to the leafs. As mentioned above, the semantic similarity values based on very general GO terms are unlikely to improve the objective function. Thus, the best α value is unlikely to be close 1.

Without any additional knowledge, we speculate that α falls into (0, 0.5]. In addition, as α increases, the objective function should decrease.

4.3 Experiment

The purpose of our experiments is to examine how the optimal values for the parameter α in the conjunctive similarity definition vary across some benchmark data sets, and the learning algorithms.

4.3.1 Data Sets

We use the leukemia [16], colon [2], prostate [47] and breast cancer [50] data sets. Table 4.1 lists a brief description of each of them. Other details have been introduced in Section 1.4. We replace the missing value of a gene by the mean value of that gene. Gene expression levels for each gene are normalized by subtracting their means and dividing by the standard deviations.

4.3.2 Feature Selection Methods

We use HykGene [56] and Minimum Redundancy Maximum Relevance (MRMR) [11] as our feature selection methods. As discussed before, HykGene uses the hierarchical clustering algorithm to put similar (i.e. redundant) features into the same cluster first, and then creates a feature subset by selecting one feature from each cluster. The distance used is the conjunctive distance defined in Equation 4.5.

Title	#Genes	#Samples	#Samples per Class	
Leukemia	7129	72	ALL	AML
			47	25
Colon Cancer	2000	62	Tumor	Normal
			40	22
Prostate Cancer	12600	102	Tumor	Normal
			52	50
Breast Cancer	24481	97	Relapse	Non-Relapse
			46	51

Table 4.1: Summary of microarray data sets used to examine the optimal value of α

For MRMR, it evaluates the quality of the feature set S by the following formula:

$$q(S) = \frac{1}{|S|}\sum_{g_i \in S} D_{g_i} - \frac{1}{|S|^2}\sum_{g_i, g_j \in S} ConjSem(g_i, g_j), \tag{4.6}$$

where D_{g_i} is the t-test value that measures the discriminative power of gene g_i with respect to the class labeling, and $ConjSem$ is defined in Equation 4.4. The feature set with the best quality is the one that maximizes $q(S)$. We use the linear incremental search algorithm to solve this optimization problem. This process assumes that if m features have already been selected, the feature set with $m + 1$ features will include these m features and the additional feature will be selected from the remaining features via a simple linear search based on Equation 4.6.

4.3.3 Classification Algorithms

We use the widely acceptable classification algorithms, LR, NB, SVM and C4.5[4] in our experiment. As introduced in Section 3.1, NB is a simplified case of a class of learning algorithms that are based directly on maximizing the posterior probability of an unknown case belonging to each class. In essence, the decision tree learning (C4.5) is also based, although indirectly, on maximizing posterior probabilities. Unlike the previous two algorithms, the SVM is based on a theory for minimizing the upper bound on the prediction errors. LR is a generalized linear model which forms a predictor variable from the linear combination of the feature variables. Since these four algorithms show significantly different classification bias, we can verify the generality of the ability of the proposed combined measure to detect feature redundancy in this experiment.

4.3.4 A General Description of the Result on Each Data Set

In this section, we show the accuracy of classifications as a result of varying α in the above four microarray data sets.

When MRMR is applied to feature selection, we first choose the top 100 genes according to their t-test values. To eliminate the complexity due to unannotated genes, we remove genes without GO annotations from the top 100 genes and use the remaining x genes as the input features. As mentioned before, MRMR is simulated by a linear incremental process. For each α value between 0 and 1 with a step value of 0.1, starting from $i = 1$, we select the optimal feature set of size i, until $i = x$. This

[4] The implementation package used is Weka [57].

optimal feature set maximizes the value for Equation 4.6 among all feature sets of size i that is a superset of the optimal feature set of size $i-1$. Among the x feature sets for each α value, we then select the one with the highest accuracy for each classifier used, where the accuracy is obtained using leave-one-out cross validation.

When using HykGene for feature selection, the top 100 genes according to their information gain are selected. Then, we also remove genes without GO annotations from the top 100 genes and use the remaining y genes as the input features. For each α value between 0 and 1 with a step value of 0.1, starting from $j = 1$ until $j = y$, we apply the hierarchical clustering algorithm to partition these genes into j clusters, where the distance between genes is measured by the conjunctive distance. Then, we select a representative from each cluster, which is the gene with the minimum sum of squares of distances to all other genes in the cluster. These representative genes form the optimal feature set of size i. Among the y feature sets for each α value, we also select the one with the highest accuracy for each classifier used.

The experiment results are shown in Figures 4.2-4.5, where the horizontal and vertical axis indicate the values of α and the classification accuracies, respectively. For $\alpha = 0$, this indicates that only the expression similarity is used by feature selection methods. The label of each data point in these figures represents the size of the selected feature subset that makes the classification algorithm achieve the corresponding accuracy.

4.3.5 Comparing α for Different Data Sets

We first analyze the result on the leukemia data set. The leukemia data set has been documented as having a good separation behavior based on expression values (100% predicting accuracy reported in [56, 11]). This implies that the quality of the expression value is good and the expression similarity is able to accurately measure the redundancy among genes. Hence, as discussed in the previous section, the best α value should be close to 0, which makes the conjunctive similarity very close to the expression similarity. In Figure 4.2, for all the classifiers used, $\alpha = 0$ is one of the best values. Also, the curves of classification accuracy are decreasing functions of α in most cases. We also note that in many cases, such as (d), (e), (f), (g), and (h) in the figure, varying the values of α does not change the classification accuracy. This means that increasing the weight of the semantic similarity in the conjunctive similarity does not decrease the classification accuracy. However, in most of the cases, when the maximum accuracy is achieved at multiple α values, the size of the feature set tends to increase as α does (e.g. Figure 4.2(e)). This implies that the semantic similarity has a minor negative influence on the classification accuracy. This influence is counter-balanced by additional informative genes and, therefore, results in larger feature sets.

We now consider the results for the colon cancer data set. This data set is relative noisy. One notorious example is the type of genes responsible for cell composition. That is, the cancerous tissue generally contains many epithelial (skin) cells, while the normal tissue contains different kinds, including smooth muscle cells. This difference of tissue composition may appear to be good classification indicators for the cancerous and normal samples, but in fact are not informative biologically [2]. As we expected, the best α value fell in $(0, 0.5]$. In Figure 4.3, in all cases, $(0, 0.5]$ always contains a maximum accuracy. In only three of the cases, (a), (e), and (f), the maximum is also reached when $\alpha = 0$. In (b), (c), (d), (e), and (g), we also observe that the classification accuracy decreases after α reaches the optimal value. As we explained in Section 4.2.2, this is because when α approaches 1, the semantic similarity becomes dominate. Consequently, the semantic similarity values based on very general GO terms could play a considerable role in the conjunctive similarity. Since they are not informative, the value of the objective function, i.e. the classification accuracy, decreases.

The breast cancer data set also presents some complications. Samples are labeled based on whether or not they relapse after five years. In other words, the preparation of the expression levels of the genes and the labeling may be several years apart [50]. Therefore, unexpected conditions arising during the time duration that may contribute to the relapse cannot be taken into account at the time when the expression values of the genes are collected. Similar to the colon cancer data set, we also expect the best α value to be in $(0, 0.5]$ in this data set. In Figure 4.4, in the 7 cases of (a), (b), (c), (e), (f), (g), and (h), $(0, 0.5]$ contains a maximum accuracy. In (f), the maximum is also reached when $\alpha = 0$. (d) is an unexpected case, where the highest accuracy is reached when $\alpha = 0.6$. In most cases, we also note that the classification accuracy usually decreases after α reaches the optimal value. This also implies that increasing the effect of non-informative semantic similarity values negatively affects the objective function.

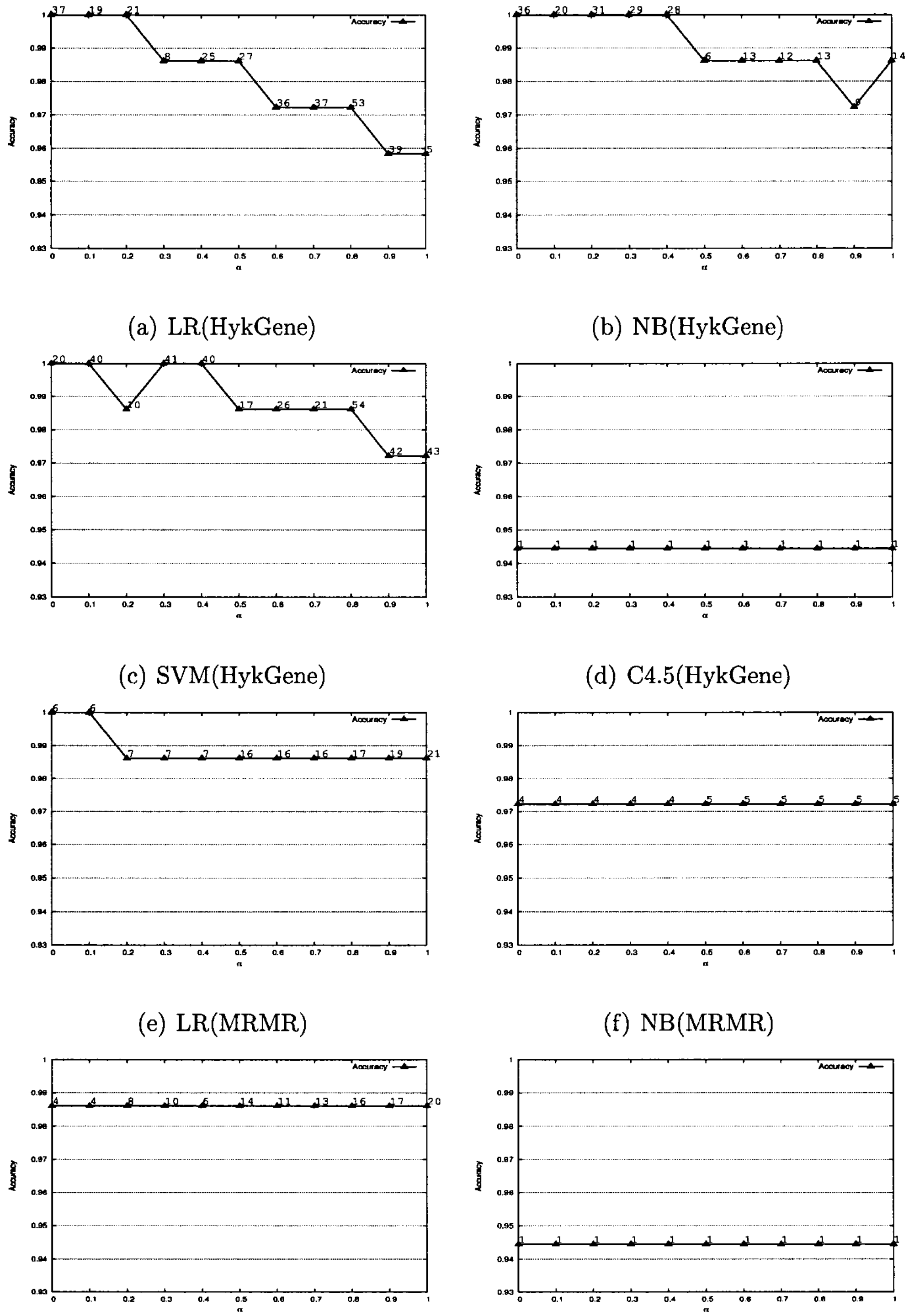
The experiment on the prostate cancer data set shows results similar to the previous two data sets. In Figure 4.5, in the 7 cases of (a), (b), (c), (d), (e), (f), and (h), $(0, 0.5]$ contains a maximum accuracy. In (d) and (h), the maximum is also reached when $\alpha = 0$. In (g), the highest accuracy is reached only when $\alpha = 0$. In most of cases, after α reaches the optimal value, the classification accuracy usually decreases as α increases.

In most cases, as we expected, the best value for the parameter α is located in interval $[0, 0.5]$. The decreasing property after the peak of the classification accuracy as a function of α is also observed in most of the cases. However, we also note that, in each data set, there are cases when the classification accuracy first decreases after the peak, and then jumps up at some points. For example, in the colon cancer data set for the HykGene-based feature selection classified by LR (Figure 4.3(a)), after reaching the peak at $\alpha = 0.5$, the classification accuracy decreases at $\alpha = 0.6$, but returns to the maximum again at $\alpha = 0.7$. However, for the other classifiers (NB, C4.5 and SVM), the classification accuracy shows a clear decreasing tendency after α reaches the optimum. Hence, we attribute this kind of behaviors to the specific combinations of the classifiers and the feature sets.

In Figure 4.4, we also observe that, in the breast cancer data set, the classification accuracy shows an unexpected increase around $\alpha = 0.6$ and then returns to an approximately decreasing tendency, such as in plots (a), (c), (d), (g), and (h). A possible reason for this phenomenon is the nature the data set. That is, the expression values may not reflect some factors that contribute to the labeling. As a result, the classification accuracy, which is based on these expression values, may demonstrate some complex behaviors.

Based on these results, we believe that in the general case, the best value for parameter α is most likely located in $[0, 0.5]$, although the exact value for any specific data set must be determined based on the specific case.

In summary, when a proper value of α is chosen, the conjunctive similarity can improve the expression similarity in classification accuracy. We also observe that in the general case such an improvement is not significant. This is partly due to the complexity of functional annotations, particularly in the multi-cellular organisms as discussed in Section 3.2.4.1, in which there exist non-informative semantic similarity values, such as those based on very general GO terms. However, although the proposed conjunctive similarity does not generate an overall significant improvement, the consistency of this improvement is very clear in all data sets. This proves the feasibility of integrating the expression similarity with the semantic similarity, and may disclose a direction for future work, using nonlinear model to combine these two types of similarities.



(a) LR(HykGene)

(b) NB(HykGene)

(c) SVM(HykGene)

(d) C4.5(HykGene)

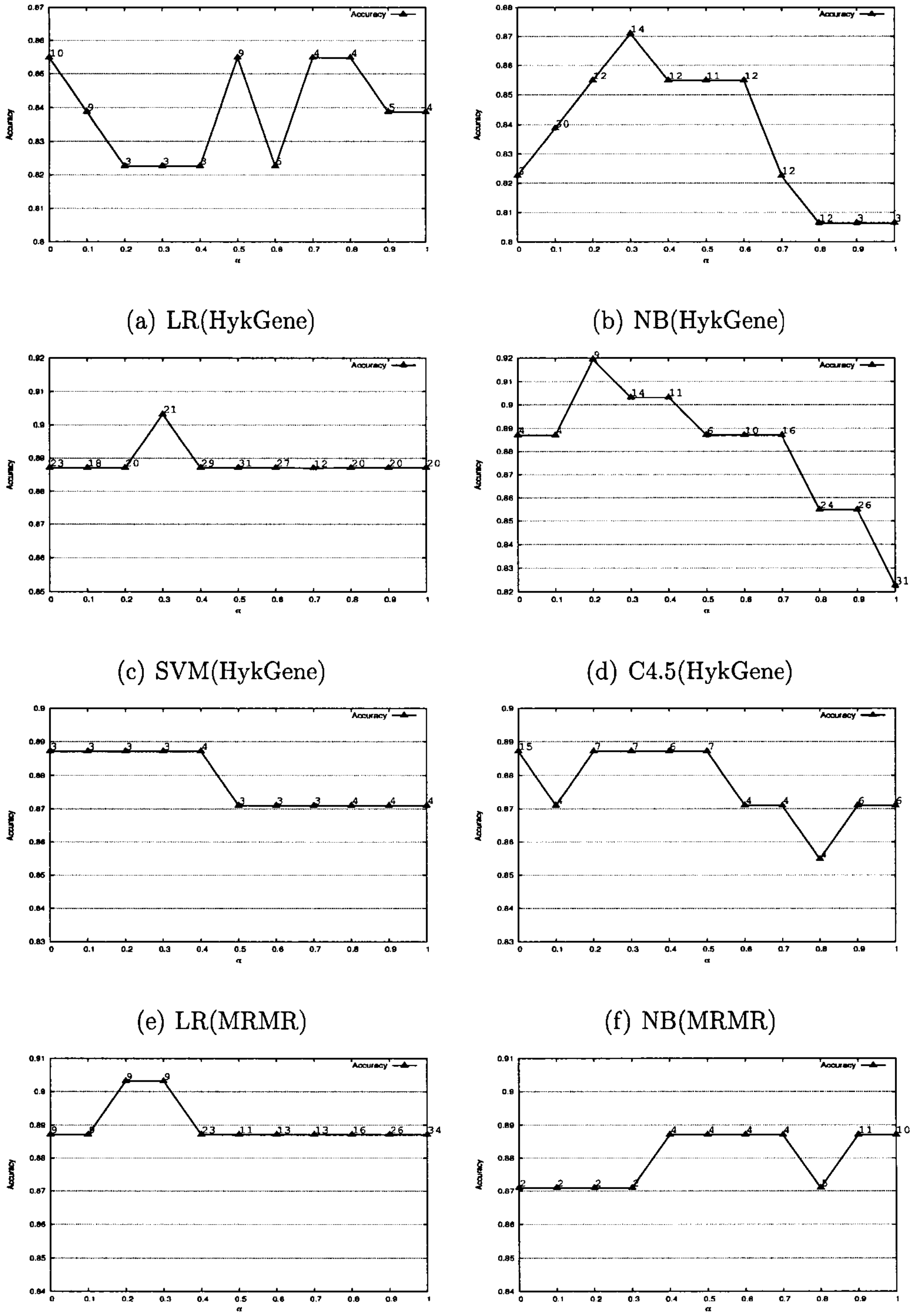
(e) LR(MRMR)

(f) NB(MRMR)

(g) SVM(MRMR)

(h) C4.5(MRMR)

Figure 4.2: Experiment Results on the Leukemia Data Set. The horizontal and vertical axis indicate the values of α and the classification accuracies, respectively.



(a) LR(HykGene)

(b) NB(HykGene)

(c) SVM(HykGene)

(d) C4.5(HykGene)

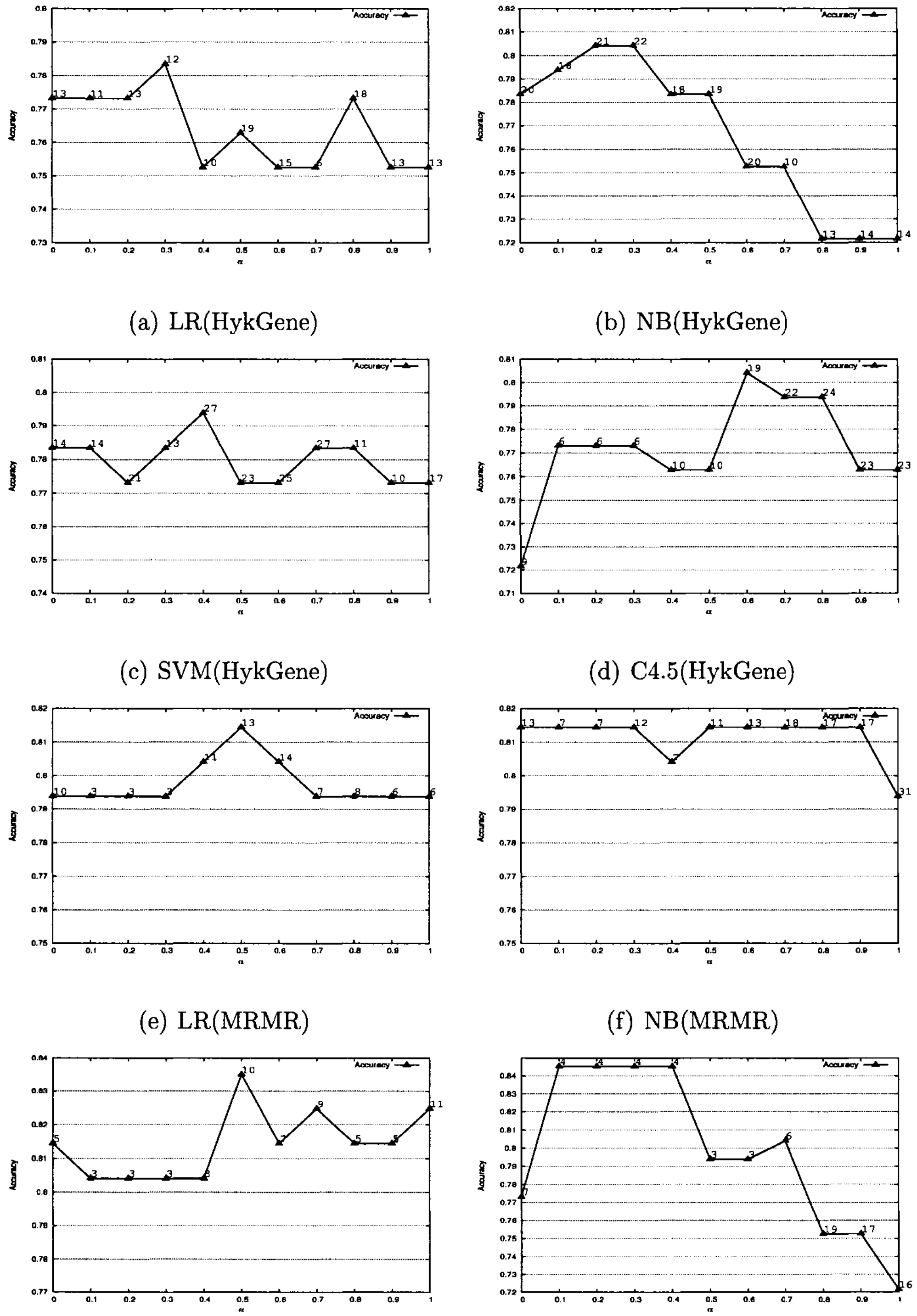
(e) LR(MRMR)

(f) NB(MRMR)

(g) SVM(MRMR)

(h) C4.5(MRMR)

Figure 4.3: Experiment Results on the Colon Cancer Data Set. The horizontal and vertical axis indicate the values of α and the classification accuracies, respectively.



(a) LR(HykGene)

(b) NB(HykGene)

(c) SVM(HykGene)

(d) C4.5(HykGene)

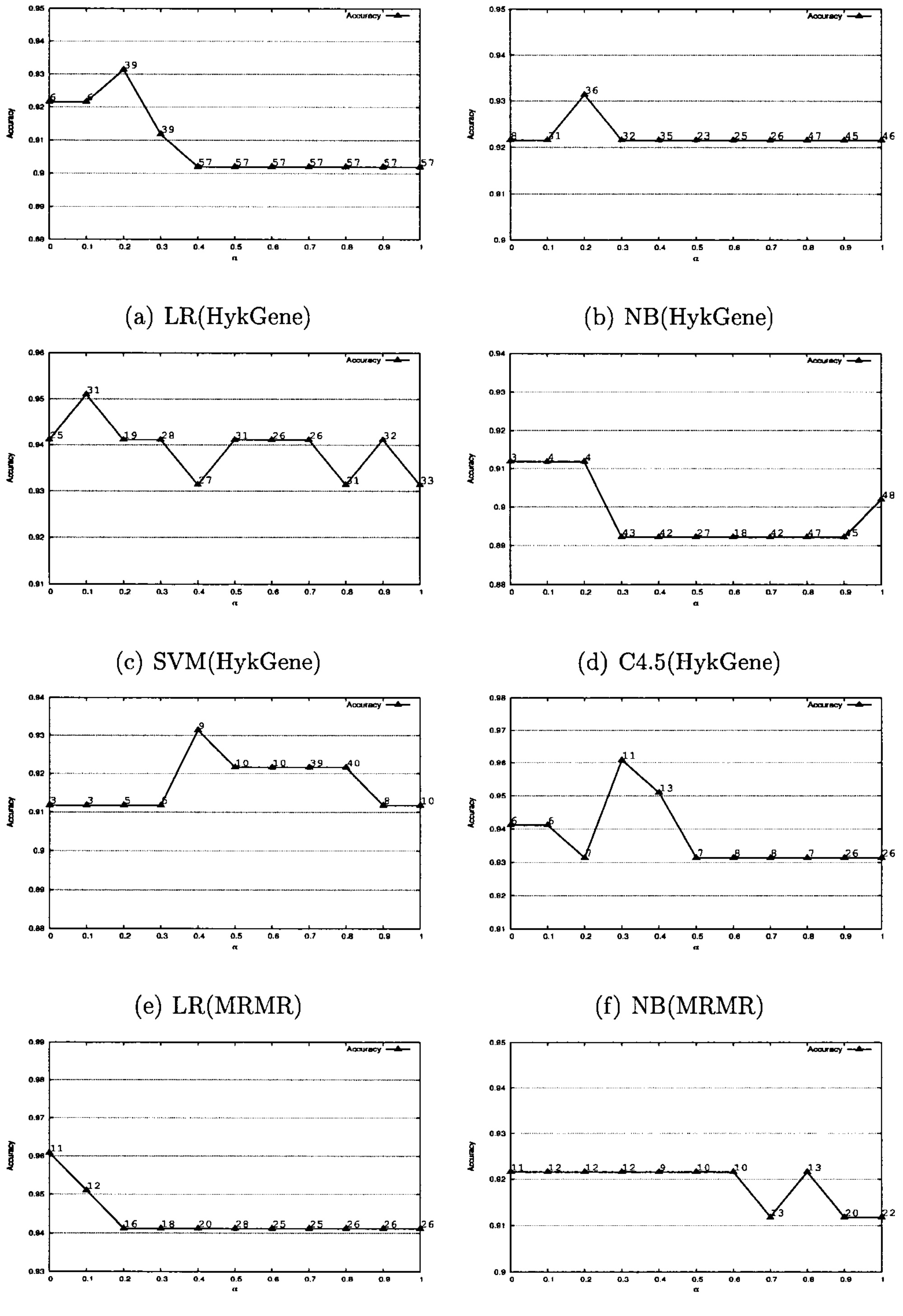
(e) LR(MRMR)

(f) NB(MRMR)

(g) SVM(MRMR)

(h) C4.5(MRMR)

Figure 4.4: Experiment Results on the Breast Cancer Data Set. The horizontal and vertical axis indicate the values of α and the classification accuracies, respectively.



(a) LR(HykGene)

(b) NB(HykGene)

(c) SVM(HykGene)

(d) C4.5(HykGene)

(e) LR(MRMR)

(f) NB(MRMR)

(g) SVM(MRMR)

(h) C4.5(MRMR)

Figure 4.5: Experiment Results on the Prostate Cancer Data Set. The horizontal and vertical axis indicate the values of α and the classification accuracies, respectively.

Chapter 5

Incorporation of GO Annotation to Gene Discriminative Power

In this chapter, we propose a method to determine the discriminative power of genes for feature selection. We first describe the algorithm and then present the related experimental results.

5.1 A Method to Integrate GO Annotation into Determining Discriminative Powers

In this section, we propose a GO-based method to select informative genes. The purpose of incorporating GO is to correct the invalid information caused by inaccurate measurement and the small number of samples in microarray data. In addition, we apply GO annotation to detect redundancy. This algorithm uses biological knowledge contained within GO annotations to process single-gene based discriminative values.

This leads to a gene ranking method different from using only single-gene based values.

5.1.1 Algorithm

The following is a high level description of our iterative algorithm.

1. Calculate the discriminative value for each candidate gene.
2. Retrieve the annotations for these genes in the biological process (BP) ontology.
3. Calculate the discriminative value of GO terms that are the averages of discriminative values of genes annotated by these terms.
4. Rank these GO terms by their discriminative value. The term with the highest value is considered as the most informative GO term in this iteration.
5. Choose the highest discriminative value gene from the most informative GO term.
6. Output this gene.
7. Remove this gene from the candidate gene set.
8. If the candidate gene set is not empty, re-calculate the discriminative value of all GO terms using the existing genes and return to Step 4 to select the next gene.

The calculation of the discriminative value can be done by any well developed traditional filtering methods. In addition to the discriminative value of individual genes, however, we also incorporate the discriminative value of GO terms, i.e. the average of the discriminative value of all the genes annotated by it. Genes annotated with same GO terms participate in same biological processes. Hence, if in a data set, most of the genes annotated with a GO term are relevant to a set of class labels, then we consider that the corresponding biological process represented by this term is also relevant to these class labels. This allow us to apply discriminative value of GO terms to verify the discriminative value of genes.

While the expression levels of a gene may be correlated with the labeling of sample classes incidentally, it is far less likely that the majority of the genes annotated by the same GO term are correlated with the sampling incidentally. In other words, in a group of genes annotated by the same term, if a majority of them have high discriminative values, then it is likely that they are indeed informative genes. This in turn implies that the likelihood that a random gene with a high discriminative value is truly a informative gene in that group will be high, compared to a gene with the same characteristic in the entire data set. This will reduce "false positives", i.e. "false informative."

On the other hand, assume that the expression level of a highly informative gene, g, reports a lower discriminative value compared to some outliers, and this gene is annotated with a GO term that is relevant to the class labels. Since most other genes annotated by this term participate in the same biological process as this gene, they are likely to be relevant to the class labels and report high discriminative values. Consequently, this GO term will have a good chance to be selected. This indicates that a gene participating in the same biological process as g will be selected. However, the selected gene would not be chosen by the traditional method because its value might not be higher than these outliers. Thus, the likelihood of missing an informative gene due to outliers is smaller than using a conventional method. Hence, the GO-based method is more robust against noise than traditional methods.

We further explain the above statement using a schematic example. According to their biological relevance to class labels, genes can be broadly categorized into two groups: informative and non-informative. Let T be a GO term, and $G(T)$ be the group of genes that are annotated by T. For any group of genes $G(T)$, let $N(G(T))$ be the percentage of $G(T)$ that are non-informative. Assume that there are three terms, T_{i-1}, T_i and T_{i+1} in the annotation. T_i is the parent of T_{i-1} and T_{i+1}. Figure 5.1 describes this structure. These three terms represent a miniature of the entire DAG in GO, since each of them may represent an agglomeration of several terms with similar biological functions. Furthermore, GO terms can be also broadly divided into two groups: informative and non-informative. Such a categorization is based on the biological relevance to class labels. It is understood that the percentage of the informative genes in an informative GO term is much higher than that in a non-informative GO term. We assume that T_{i+1} is informative and T_{i-1} is non-informative. Here, we assume the transitivity of gene annotation. This means that,

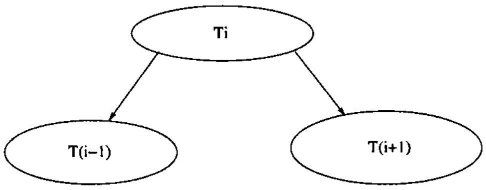


Figure 5.1: A Miniature of the Entire DAG

if a gene is annotated by a term, all of its parent terms also annotate that gene[1]. Hence, we have $G(T_{i+1}) \subseteq G(T_i)$ and $G(T_{i-1}) \subseteq G(T_i)$. Suppose that

$$\{g_1, g_2, \ldots, g_{k-1}, g_k, g_{k+1}, \ldots, g_{l-1}, g_l, g_{l+1}, \ldots, g_n\}$$

is the group of genes that are annotated by T_i, where these genes are ranked by their discriminative value. We assume that g_k is a "false positive" gene and it is annotated by T_{i-1}. In addition, g_l is the last informative gene in this sequence. Since T_{i+1} is informative, it holds that most informative genes should be in $G(T_{i+1})$. Hence, we suppose that

$$G(T_{i+1}) = \{g_1, g_2, \ldots, g_{k-1}, g_{k+1}, \cdots\ g_l, \ldots\}$$

and

$$G(T_{i-1}) = \{g_k, \ldots, g_{l+1}, \ldots, g_n\}.$$

Since $N(G(T_{i+1})) < N(G(T_i)) < N(G(T_{i-1}))$, we can then establish that

$$(AverageScore)_{G(T_{i+1})} \quad > \quad (AverageScore)_{G(T_i)} \quad > \quad (AverageScore)_{G(T_{i-1})}.$$

Remember that our method is to select the highest ranked gene from the GO term with the highest average score. Hence, in the first iteration, T_{i+1} is chosen and g_1 is the output. After the $(k-1)^{st}$ iteration, we expect that

$$G(T_{i+1}) = \{g_{k+1}, \ldots, g_l, \ldots\}$$

and

$$G(T_{i-1}) = \{g_k, \ldots, g_{l+1}, \ldots, g_n\}.$$

[1] The motivation of the assumption of transitivity is introduced in Section 5.1.2.

Since we assume that the majority of the genes in $G(T_{i+1})$ have high discriminative values, after removing a number of highly expressed genes, we expect that the average score of $G(T_{i+1})$ is still higher than $G(T_{i-1})$. Hence, in most cases, $N(G(T_{i+1})) < N(G(T_i)) < N(G(T_{i-1}))$. Thus, we still can establish that

$$(AverageScore)_{G(T_{i+1})} \quad > \quad (AverageScore)_{G(T_i)} \quad > \quad (AverageScore)_{G(T_{i-1})}.$$

In the k^{th} iteration, it is likely that T_{i+1} is still chosen and g_{k+1} is the output. Hence, we expect that the ranking of genes generated by our method resembles the following:

$$\{g_1, g_2, \ldots, g_{k-1}, g_{k+1}, \ldots, g_{k+d}, g_k, \ldots, g_n\}.$$

It is known that this "false positive" gene, g_k, will be ranked lower than that in the traditional single-gene based method.

One common way of feature selection is to select the top m genes, say 100. So we define the false positive rate as the percentage of the false positive genes among these m genes. As we demonstrated above, the GO-based method ranks the false positive genes lower than the traditional method. Hence, it is likely that the number of the false positive genes among these m genes decreases and this leads to reducing the false positive rate.

It is possible that some non-informative genes are annotated by T_{i+1}. We suppose that a non-informative gene g_j is annotated by T_{i+1} $(1 < j < k)$. In the j^{th} iteration, our method selects this gene, but the traditional algorithm will rank it at the same position. Hence, our algorithm is as good as the traditional algorithm in light of these non-informative genes.

In addition to verifying gene expression values, the other purpose of incorporating GO annotation is to eliminate redundancy. It is very likely that genes that are annotated by the same term have certain degrees of similarities. Since among all the genes annotated by a term, we select only one, we get rid of redundancy in a single iteration. Moreover, in the next iteration, the likelihood of choosing the same group will be smaller, since the highest ranking gene has been removed from that group. This also reduces the chance of selecting a redundant gene.

5.1.2 Determine the Biological Soundness of GO Terms

A subtle point in dealing with annotations is how to choose the proper degree of the biological soundness of these annotations. This degree could influence the effectiveness of our algorithm. In GO DAG, the lower a term is, the more biological information it presents. This means that a term at a very high level should not be considered informative even though it has a high discriminative value, because it does not contain enough biological information relevant to the sample labeling. However, it would be too restrictive to be practically useful if we consider terms only at low levels, since many genes are not annotated with low level terms at current stage. One straightforward approach to cope with this problem would be to declare a parameter to indicate the level at which the terms should be considered. However, this will introduce a complicated question of how to determine this parameter. (If this approach is used, we suspect that the values of this parameter will depend heavily on data sets, so they will have to be determined by experiment.)

In this work, we adopt a more efficient approach, the transitivity of gene annotation. This means that if a gene is annotated by a term, then we assume all its parent terms also annotate that gene. By this assumption, a general term may be taken as an informative term only when all of its children terms have high discriminative values. Thus, it is harder for higher level terms to be informative than it is for lower level terms. The advantage of this approach is that it does not compromise our projected effectiveness of the algorithm, while at the same time keep the extra overhead low.

In the next subsection, we give the probabilistic argument to demonstrate that our method outperforms the traditional one.

5.1.3 Probability Derivation

In the follows, we present a semi-formal probabilistic argument to justify the use of our algorithm. We concentrate on the analysis of "false positives". We assume that for a given labeling, the genes are categorized into two classes: *informative* and *non-informative.* Conceptually, a gene is informative if and only if it is highly correlated with the labeling in a *biological sense.* Exactly how to determine whether or not a gene is informative is not important for our derivation. We use two values, 'infor' and 'non-infor', to indicate informative and non-informative genes, respectively. Also, we use the terms 'high' and 'low' to denote high and low measured discriminative scores, respectively, without caring about their exact numerical values. Thus, we approximate our algorithm by randomly selecting a gene with a high discriminative score in the selected group. We consider any particular iteration i. Let G be the entire group of genes, and T be the group with the highest average measured expression value in iteration i. We use $P_X(r)$ to denote the probability that a random gene selected from group X has a property of r. We make the following assumptions:

1. $P_T(non-infor) < P_G(non-infor)$

2. $P_T(high|non - infor) = P_G(high|non - infor) = P(high|non - infor)$

3. $P_T(high|infor) = P_G(high|infor) = P(high|infor)$

4. $P_G(high|non - infor) < P_G(high|infor)$

The first condition claims that it is less likely for a random gene to be non-informative in group T than it is in the entire group. The second condition implies that the likelihood that a gene has a high discriminative score given that it is non-informative is the property of the gene, not a property of any group to which the gene belongs, and therefore, it is independent of the group to which it belongs. The idea for the third condition is similar. The fourth condition should be intuitively clear. If a gene is informative, then it is more likely for it to have a high discriminative score than if it is non-informative.

Consider the value of $P_X(non - infor|high)$. This is the conditional probability that a randomly selected gene is non-informative given that it has a high discriminative score in group X. Therefore, we use $P_T(non - infor|high)$ and $P_G(non - infor|high)$ as an approximation of the false positive rate of our algorithm and that of the traditional filtering algorithm, respectively. We shall now compare these two values.

First, by the Bayesian Theorem, we have

$$P_T(non - infor|high) = \frac{P(high|non - infor) \cdot P_T(non - infor)}{P(high|non - infor) \cdot P_T(non - infor) + P(high|infor) \cdot P_T(infor)}$$

and

$$P_G(non-infor|high) = \frac{P(high|non-infor) \cdot P_G(non-infor)}{P(high|non-infor) \cdot P_G(non-infor) + P(high|infor) \cdot P_G(infor)}.$$

Second, the following is true:

$$\begin{aligned} P(high|non-infor) \cdot P_T(non-infor) + P(high|infor) \cdot P_T(infor) &= \\ P(high|non-infor) + (P(high|infor) - P(high|non-infor)) \cdot P_T(infor) &> \\ P(high|non-infor) + (P(high|infor) - P(high|non-infor)) \cdot P_G(infor) &= \\ P(high|non-infor) \cdot P_G(non-infor) + P(high|infor) \cdot P_G(infor). \end{aligned}$$

Also, we have the following:

$$P(high|non-infor) \cdot P_T(non-infor) < P(high|non-infor) \cdot P_G(non-infor).$$

The above two inequalities imply

$$P_T(non-infor|high) < P_G(non-infor|high).$$

This means that our algorithm has a smaller chance to select false positives than the traditional filtering method under assumptions 1 to 4. In the next section, we will use experimental results to further verify this point.

5.2 Experiments

5.2.1 Data Sets and Experiment Setup

In the experiment, we use the leukemia [16], colon [2], lung [17] and breast cancer [50] data sets. Table 5.1 shows the simple description of these data sets. The detail

Title	#Genes	#Samples	#Samples per class	
Leukemia	7129	72	ALL	AML
			47	25
Colon Cancer	2000	62	tumor	normal
			40	22
Lung Cancer	12533	181	MPM	ADCA
			31	150
Breast Cancer	24481	97	relapse	non-relapse
			46	51

Table 5.1: Summary of microarray data sets used to examine the effectiveness of GO-based discriminative power

of these microarray data sets, such as the data source and the meaning of the class labels, has been introduced in Chapter 1. We replace the missing value of a gene by the mean value of that gene. Gene expression levels for each gene are normalized by subtracting their means and dividing by their standard deviations.

In these experiments, we use information gain to calculate individual discriminative values of genes which can be decided by Equation 3.3. In the data preprocessing, we removed all genes whose discriminative values are equal to zero since they do not contain any information relevant to class label. Furthermore, for simplicity, we ignore genes without gene annotations. We notice that some unannotated genes are relevant to class labels, so we may lose some chances of functional gene discovery. However, the primary goal of these experiments is to determine whether or not the GO-based method is more effective than the single-gene based method.

5.2.2 Experimental Result

In this subsection, we compare the effectiveness of the GO-based method with that of the single-gene based method. We use two criteria to evaluate the effectiveness: the number of selected genes and the predictive accuracy. It is desirable to select the smallest number of genes which can achieve the highest predictive accuracy. LR, NB, and SVM were selected as the classification algorithms in these experiment. These classifiers have different classification bias, so we can sufficiently test the effectiveness of the proposed method. Details about these classifiers are reviewed in Section 3.1.

In the single-gene based method, genes are ranked by their information gain, while in the GO-based method, genes are ranked by the algorithm proposed in the previous section. For both of these two methods, starting $i = 1$, we select the optimal feature set of size i which contains the top i genes according to the corresponding ranking, until $i = 100$. Among the 100 feature sets for these two methods, we then select the one with the highest accuracy for each classifier used, where the accuracy is obtained using leave-one-out cross validation. The results are shown in Tables 5.2, 5.3, 5.4, and 5.5.

In most of these data sets, the GO-based method attains promising results. Compared to the single-gene based method, this method attains better accuracies or attains the same accuracy with fewer genes. The above results sufficiently demonstrate the effectiveness of the proposed GO-based algorithm.

One exceptional case happens for the breast cancer data set for the NB algorithm

Classifier	GO		Single-Gene	
	Accuracy	#Genes	Accuracy	#Genes
NB	100%	9	98.61%	63
LR	100%	15	97.22%	5
SVM	98.61%	19	98.61%	26

Table 5.2: Comparison of accuracy between the GO-based and single-based methods on the leukemia data set

Classifier	GO		Single-Gene	
	Accuracy	#Genes	Accuracy	#Genes
NB	65.98%	1	61.86%	58
LR	78.35%	5	73.20%	8
SVM	79.38%	30	76.28%	12

Table 5.3: Comparison of accuracy between the GO-based and single-based methods on the breast cancer data set

Classifier	GO		Single-Gene	
	Accuracy	#Genes	Accuracy	#Genes
NB	100%	81	100%	28
LR	100%	7	100%	45
SVM	100%	27	99.44%	11

Table 5.4: Comparison of accuracy between the GO-based and single-based methods on the lung cancer data set

Classifier	GO		Single-Gene	
	Accuracy	#Genes	Accuracy	#Genes
NB	83.87%	14	82.26%	14
LR	88.71%	9	88.71%	9
SVM	88.71%	18	90.32%	15

Table 5.5: Comparison of accuracy between the GO-based and single-based methods on the colon cancer data set

(Table 5.3). The GO-based method attains a 65.98% predicting accuracy with one gene, while the best accuracy of the single-gene based method is 61.86% with as many as 58 genes. We further investigate this result due to this large gap between the performance of the two methods.

The gene in the singleton set selected by the GO-based method is $ALDH6A1$. This gene has the 7^{th} highest information gain. We also observe that adding any genes to this singleton set will dramatically reduce the classification accuracy. For example, in our experiment, when we add one more gene to the set, the accuracy is 49.5%. When we add two or three more genes to the set, the accuracies are 53.6% and 52.6%, respectively. On the other hand, the single-gene based method selects genes sequentially. It can not select this 7^{th} gene without including the first six genes. Thus, when it continues to select more gene sequentially, it terminates with 58 genes when it can reach the maximum accuracy of 61.86%. The above results imply that gene $ALDH6A1$ is the best gene and combing this gene with any other gene will reduce the classification accuracy. This phenomenon should be attributed to the special combination of the classifier and this gene. We speculate that Gene $ALDH6A1$ is a really biologically meaningful gene and plays a predominant role in the classification. However, NB is based on the arguable realistic assumption that all attributes are independent. This means that for any feature set that contains this gene, NB multiplies the individual probability of all genes in the feature set together to determine the posterior probability. Thus, no genes can be ignored; even they are noisy or redundant. As a consequence of the predominant predictive power of gene $ALDH6A1$, the addition of other genes negatively affects the classification and decreases the prediction accuracy (Table 5.3).

From the perspective of biology, $ALDH6A1$ is annotated with GO terms "Pyrimidine nucleotide metabolism" and 'Valine metabolism" in the biological process ontology. This gene is associated with metabolic diseases generally characterized by neurologic complications and developmental delay [51]. Since no previous work shows that this gene is involved in development and diagnosis of breast cancer, it is worth further investigation by biologists.

Chapter 6

Conclusion

Microarray technology is one of the most powerful and versatile tools for functional genomic studies. The most attractive application of this technology is to simultaneously monitor the expression of thousands of genes. In the analysis of microarray data, one major challenge is the small number of samples compared with the huge number of genes. To reduce the computational cost and identify genes relevant to the biological process of the microarray data measures, feature selection becomes an essential step in the study of microarrays.

In this thesis, we present some approaches to integrate Gene Ontology into the feature selection in microarray data. Due to the limitations of microarray technology, gene expression values may contain noise or they may not provide a good estimation of the underlying distribution. It would be very difficult to resolve this problem solely by expression value-based methods. Hence, we propose to use biological knowledge contained in GO annotations to verify inaccurate measures.

In particular, we demonstrate the intrinsic capability of the semantic similarity in detecting redundancy among genes. Moreover, we introduce a new similarity metric which is defined as the convex combination of the semantic and expression similarities. Using feature selection methods of HykGene and MRMR, we test this conjunctive similarity in four widely used datasets. These experiment results show that this new measure is more effective than the traditional expression similarity.

Furthermore, we propose a novel feature selection algorithm which combines individual discriminative powers with GO annotations of genes. This integration enables the proposed method to be more robust against noise than the traditional single-gene based method, since the proposed method evaluates the discriminative power of both genes and GO terms annotating them. We also test this method in several data sets and our method outperforms the conventional method in most classifiers.

In the study of microarray data, most existing work using GO annotations treats them as auxiliary tool to interpret the final results. One typical example is to evaluate the biological soundness of the result of gene-expression-value-based clustering. In reality, as we discussed in Section 3.2.4.2, the gene expression value is one of the 14 major sources of GO annotations. Thus, GO annotations are affected in part by gene expression values. In this thesis, however, we integrate GO annotations into feature selection for microarray data. This indicates that GO is, oppositely in some sense, allowed to direct the analysis of gene expression values. Although these two styles of work may seem opposite in direction, we believe that they reinforce each other. That is, on one hand, analyzing gene expression values help biologists to understand the biological roles of genes and, consequently, lead to more specific and informative GO annotations. On the other hand, GO annotations are extracted from other sources besides gene expression values, such as protein sequence analysis and genetic interaction. Therefore, the biological information represented by GO annotations is far more comprehensive than that represented by expression values. Hence, GO annotations provide a wider scope of useful information for feature selection, which is not included by expression values alone. Consequently, such supplemental information leads to improved effectiveness of feature selection.

Bibliography

[1] D. W. Aha, D. Kibler, and M. K. Albert. Instance-based learning algorithms. *Machine Learning*, 6(1):37–66, 1991.

[2] 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. *Proceedings of the National Academy of Sciences*, 96(12):6745–6750, 1999.

[3] A. Benard and E. Bos-Levenbach. The plotting of observations on probability paper. *Statistica Neerlandica*, 7:163–173, 1953.

[4] A. Brazma and J. Vilo. Gene expression data analysis. *FEBS Letters*, 480(1):17–24, 2000.

[5] M. Brown, W. Grundy, D. Lin, N. Christianini, C. Sugnet, M. Jr, and D. Haussler. Knowledge-based analysis of microarray gene expression data using support vector. *Proceedings of the National Academy of Sciences of the United States of America*, 97(1):262 - 267, 2000.

[6] A. Budanitsky and G. Hirst. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In *Proceedings of Workshop on WordNet and Other Lexical Resources*, 2001.

[7] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[8] S. L. Cessie and J. C. V. Houwelingen. Ridge estimators in logistic regression. *Applied Statistics*, 41(1):191–201, 1992.

[9] P. Clark and T. Niblett. The cn2 induction algorithm. *Machine Learning*, 3(4):261–283, 1989.

[10] M. Diehn and G. Sherlock. Source: a unified genomic resource of functional annotations, ontologies, and gene expression data. *Nucleic Acids Research*, 31(1):219–223, 2003.

[11] C. Ding and H. Peng. Minimum redundancy feature selection from microarray gene expression data. In *Proceedings of the Computational Systems Bioinformatics Conference*, pages 523–529. IEEE Computer Society, 2003.

[12] U. Fayyad and K. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 1022–1027, 1993.

[13] Gene Ontology Consortium. Creating the gene ontology resource: design and implementation. *Genome Research*, 11(8):1425–1433, 2001.

[14] Gene Ontology Consortium. The gene ontology (GO) database and informatics resource. *Nucleic Acid Research*, 32(Database issue):D258–D261, 2004.

[15] D. Gershon. Microarray technology: an array of opportunities. *Nature*, 416(6883):885–891, 2002.

[16] T. R. Golub and et al. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537, 1999.

[17] G. Gordon and R. Jensen. Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma. *Cancer Research*, 62(17):4963–4967, 2002.

[18] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.

[19] J. Han and M. Kamber. *Data mining: concepts and techniques.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000.

[20] B. Hanczar, M. Courtine, A. Benis, and C. Hennegar. Improving classification of microarray data using prototype-based feature selection. *SIGKDD Explor. Newsl.*, 5(2):23–30, 2003.

[21] H. L. Harter. Expected values of normal order statistics. *Biometrika*, 48(1-2):151–165, 1961.

[22] I. Inza, B. Sierra, R. Blanco, and P. Larra. Gene selection by sequential search wrapper approaches in microarray cancer class prediction. *Journal of Intelligent and Fuzzy Systems*, 12(1):25–33, 2002.

[23] D. Jiang, C. Tang, and A. Zhang. Cluster analysis for gene expression data: a survey. *IEEE Transactions on Knowledge and Data Engineering*, 16(11):1370–1386, 2004.

[24] J. J. Jiang and D. W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of International Conference on research in Computational Linguistics*, 1997.

[25] J. Knight. When the chips are down. *Nature*, 410(6831):860–861, 2001.

[26] R. Kohavi and G. H. John. Wrapper for feature subset selection. *Artificial Intelligence*, 97(1-2):273–274, 1997.

[27] D. Koller and M. Sahami. Toward optimal feature selection. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 284–292. Morgan Kaufmann, 1996.

[28] R. Kothapalli, S. Yoder, S. Mane, and T. Loughran. Microarray results: how accurate are they? *BMC Bioinformatics*, 3(1):22–31, 2002.

[29] A. Lagreid, T. R. Hvidsten, H. Midelfart, J. Komorowski, and A. K. Sandvik. Predicting gene ontology biological process from temporal gene expression patterns. *Genome Research*, 13(5):965–979, 2003.

[30] J. Li, H. Liu, and LimsoonWong. Mean-entropy discretized features are effective for classifying high-dimensional biomedical data. In *Proceedings of 3rd ACM SIGKDD Workshop on Data Mining*, pages 17–24, 2003.

[31] D. Lin. An information-theoretic definition of similarity. In *Proceedings 15th International Conference on Machine Learning*, pages 296–304. Morgan Kaufmann Publishers Inc, 1998.

[32] H. Liu, J. Li, and L. Wong. A comparative study of feature selection and classification methods using gene expression profiles and proteomic patterns. *Genome Informatics*, 13:51–60, 2002.

[33] H. Liu and R. Setiono. Chi2: Feature selection and discretization of numeric attributes. In *Proceedings of 7th IEEE International Conference on Tools with Artificial Intelligence*, page 388. IEEE Computer Society, 1995.

[34] T. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[35] P. Pavlidis, D. Lewis, and W. Noble. Exploring gene expression data with class scores. In *Proceedings of the Seventh Annual Pacific Symposium on Biocomputing*, pages 474–485. World Scientific, 2002.

[36] J. C. Platt. *Advances in kernel methods: support vector learning*, chapter Fast training of support vector machines using sequential minimal optimization, pages 185–208. MIT Press, Cambridge, MA, USA, 1999.

[37] P. Pudil, J. Novovicova, and J. Kittler. Floating search methods for feature selection with nonmonotonic criterion functions. In *Proceedings of the 12th IAPR*

International Conference on Computer Vision & Image Processing, pages 279–283. IEEE Computer Society, 1994.

[38] J. Qi and J. Tang. Gene ontology driven feature selection from microarray gene expression data. In *Proceedings of 2006 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, pages 1–7. IEEE Computer Society, 2006.

[39] J. Qi and J. Tang. Integrating gene ontology into discriminative powers of genes for feature selection in microarray data. In *Proceedings of the 2007 ACM symposium on Applied computing*, pages 430–434. ACM, 2007.

[40] J. Quackenbush. Computational analysis of microarray data. *Nature reviews. Genetics*, 2(6):418–427, 2001.

[41] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 2003.

[42] R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1993.

[43] M. Robnik, Ikonja, and I. Kononenko. Theoretical and empirical analysis of relieff and rrelieff. *Machine Learning*, 53(1-2):23–69, 2003.

[44] S. S. Shapiro and M. B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3-4):591–611, 1965.

[45] L. Shen and E. C. Tan. Dimension reduction-based penalized logistic regression for cancer classification using microarray data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2(2):166–175, 2005.

[46] C. Sima and E. R. Dougherty. What should be expected from feature selection in small-sample settings. *Bioinformatics*, 22(19):2430–2436, 2006.

[47] D. Singh, P. G. Febbo, K. Ross, D. Jackson, J. Manola, C. Ladd, P. Tamayo, A. A. Renshaw, A. V. D'Amico, J. P. Richie, E. S. Lander, M. Loda, P. W. Kantoff, T. R. Golub, and W. R. Sellers. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 1(2):203–209, 2002.

[48] A. Tefferi, M. E. Bolander, S. M. Ansell, E. D. Wieben, and T. C. Spelsberg. Primer on medical genomics part iii: Microarray experiments and data analysis. *Mayo Clinic proceedings*, 77(9):927–940, 2002.

[49] J. Tuikkala, L. Elo, O. Nevalainen, and T. Aittokallio. Improving missing value estimation in microarray data with gene ontology. *Bioinformatics*, 22(5):566–572, 2006.

[50] L. J. van 't Veer and H. Dai. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415(6871):530–536, 2002.

[51] V. Vasiliou and A. Pappa. Polymorphisms of human aldehyde dehydrogenases consequences for drug metabolism and disease. *Pharmacology*, 61(2):192–198, 2000.

[52] H. Wang and F. Azuaje. Gene expression correlation and gene ontology-based similarity: An assessment of quantitative relationships. In *Proceedings of IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, pages 25–31. IEEE Computer Society, 2004.

[53] H. Wang and F. Azuaje. An ontology-driven clustering method for supporting gene expression analysis. In *Proceedings of the 18th IEEE International Symposium on Computer-Based Medical Systems*, pages 389–394. IEEE Computer Society, 2004.

[54] H. Wang, H. Zheng, F. Azuaje, O. Bodenreider, and A. Chesneau. Linking gene ontology-driven similarity and gene co-expression in mouse. Available at `http://mor.nlm.nih.gov:8000/pubs/alum/2005-azuaje.pdf`.

[55] Y. Wang and F. Makedon. Application of relief-f feature filtering algorithm to selecting informative genes for cancer classification using microarray data. In *Proceedings of the 2004 IEEE Computational Systems Bioinformatics Conference*, pages 497–498. IEEE Computer Society, 2004.

[56] Y. Wang, F. S. Makedon, and J. C. Ford. Hykgene: a hybrid approach for selecting marker genes for phenotype classification using microarray gene expression data. *Bioinformatics*, 21(8):1530–1537, 2005.

[57] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2003.

[58] E. Xing, M. Jordan, and R. Karp. Feature selection for high-dimensional genomic microarray data. In *Proceedings of the 18th International Conference on Machine Learning*. Morgan Kaufmann, San Francisco, CA, 2001.

[59] X. Xu and A. Zhang. Selecting informative genes from microarray dataset by incorporating gene ontology. In *Proceedings of the Fifth IEEE Symposium*

on Bioinformatics and Bioengineering, pages 241–245. IEEE Computer Society, 2005.

[60] L. Yu and H. Liu. Redundancy based feature selection for microarray data. In *Proceedings of the Tenth ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 737–742. ACM, 2004.

[61] J. Zhu and T. Hastie. Classification of gene microarrays by penalized logistic regression. *Biostatistics*, 5(3):427–443, 2004.