USQ

UNIVERSITY
OF SOUTHERN
QUEENSLAND

PER STUDIA MENS NOVA

AUSTRALIA

Accurate and Robust Algorithms for Microarray Data Classification

Ms Hong Hu (Hons)

Department of Mathematics and Computing
Faculty of Sciences

Submitted in fulfilment of the requirements of the degree of
Doctor of Philosophy

November 2008

Statement of originality

This work has not previously been submitted for a degree or diploma in university. To the best of my knowledge and belief, the dissertation contains no material previously published or written by another person except where due reference is made in the dissertation itself.

Candidate:	Date:
Supervisor(s):	Date:

Acknowledgements

It is my great pleasure to thank my principal supervisor Dr Hua Wang (M&C) for his advice and encouragement during this research. My former supervisor, A/Prof. Jiuyong Li (UniSA), deserves my tremendous appreciation. A/Prof Li has provided guidance, encouragements, and thoughtful suggestions to my research work. Without his constant support and help, I would not imagine I could complete the research and the writing up of this thesis. I am also indebted to my former associate supervisor, A/Prof Grant Darggad (FoS), for his guidance especially during my early PhD research period.

I would also like to express my gratitude to the many staff in the Department for their unbounded help, and many invaluable friendships. I thank the Australian government for awarding me an Australian Postgraduate Scholarship (APA), and the Department of Maths and Computing for offering me a studentship, and the CBSi research centre for financial support to attend International conferences.

I owe a lot to Jingyan Li and Huiqing Liu for the offer of free software CS4 and Greg Otto for his critical proofreading of various publications and

this thesis.

Last but not least, I offer my heartfelt gratitude to my husband Zhongwei Zhang, my lovely children Jenny and Jason and my sister Ying for their endless support and enduring love.

I dedicate this thesis to the memory of my beloved father Qi Hu, mother Guilan Zhang and sister Yue Hu.

List of Publications

1. Hu, H., Li, J., Wang, H. and Daggard, G., Wang, L, Z. (2008). Robustness analysis of diversified ensemble decision tree algorithms for Microarray data classification. In Proc. 2008 International Conference on Machine Learning and Cybernetics (ICMLC2008), Kunming, China, Volume 1, 115-120, 2008.

2. Hu, H., Li, J., Plank, A., Wang, H. and Daggard, G. (2006). A Comparative Study of Classification Methods For Microarray Data Analysis. In Proc. Fifth Australasian Data Mining Conference (AusDM2006), Sydney, Australia. CRPIT, 61. Peter, C., Kennedy, P.J., Li, J., Simoff, S.J. and Williams, G.J., Eds., ACS. 33-37.

3. Hu, H., Li, J., Wang, H. and Daggard G. (2006) Combined Gene Selection Methods for Microarray Data Analysis,KES (1), Lecture Notes in Computer Science, Vol. 4251, pp. 976-983, Springer, 2006.

4. Hu, H., Li, J., Wang, H., Daggard, G. and Shi, M. (2006). A Maximally Diversified multiple decision trees Algorithm for Microarray Data Clas-

sification. In Proc. 2006 Workshop on Intelligent Systems for Bioinformatics (WISB 2006), Hobart, Australia. CRPIT, 73. Boden, M. and Bailey, T.L., Eds., ACS. 35-38.

5. Hu, H. and Li, J. (2005). Using Association Rules to Make Rule-based Classifiers Robust. In Proc. Sixteenth Australasian Database Conference (ADC2005), Newcastle, Australia. CRPIT, 39. Williams, H. E. and Dobbie, G., Eds., ACS. 47-54.

Awards

1. The Doctoral Consortium Award for Best Presentation. Australasian Computing Doctoral Consortium (ACDC). January 29-30 2007, University of Ballarat, Ballarat, Australia.

2. Consortium Financial Assistance (ACDC2007). Australasian Computing Doctoral Consortium (ACDC).January 29-30 2007, University of Ballarat, Ballarat, Australia.

3. International Centre of Excellence for Education in Mathematics (ICE-EM) Australian Travel Scholarships. BioInfoSummer 2005 ICE-EM Summer Symposium in Bioinformatics. 28 November - 2 December, 2005, Australian National University, Canberra, Australia

4. Australian Postgraduate Awards (APAs). March 2005- August 2007. USQ

Abstract

Microarray data classification is used primarily to predict unseen data using a model built on categorized existing Microarray data. One of the major challenges is that Microarray data contains a large number of genes with a small number of samples. This high dimensionality problem has prevented many existing classification methods from directly dealing with this type of data. Moreover, the small number of samples increases the overfitting problem of Classification, as a result leading to lower accuracy classification performance. Another major challenge is that of the uncertainty of Microarray data quality. Microarray data contains various levels of noise and quite often high levels of noise, and these data lead to unreliable and low accuracy analysis as well as the high dimensionality problem. Most current classification methods are not robust enough to handle these type of data properly.

In our research, accuracy and noise resistance or robustness issues are focused on. Our approach is to design a robust classification method for Microarray data classification.

An algorithm, called diversified multiple decision trees (DMDT) is proposed, which makes use of a set of unique trees in the decision committee. The DMDT method has increased the diversity of ensemble committees and therefore the accuracy performance has been enhanced by avoiding overlapping genes among alternative trees.

Some strategies to eliminate noisy data have been looked at. Our method ensures no overlapping genes among alternative trees in an ensemble committee, so a noise gene included in the ensemble committee can affect one tree only; other trees in the committee are not affected at all. This design increases the robustness of Microarray classification in terms of resistance to noise data, and therefore reduces the instability caused by overlapping genes in current ensemble methods.

The effectiveness of gene selection methods for improving the performance of Microarray classification methods are also discussed.

We conclude that the proposed method DMDT substantially outperforms the other well-known ensemble methods, such as Bagging, Boosting and Random Forests, in terms of accuracy and robustness performance. DMDT is more tolerant to noise than Cascading-and-Sharing trees (CS4), particulary with increasing levels of noise in the data. The results also indicate that some classification methods are insensitive to gene selection while some methods depend on particular gene selection methods to improve their performance of classification.

Table of Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Microarray data classification

The completion of the Human Genome Project [51] which generated a rough draft of the human genome sequence led us into a post-genomics era, and consequently has changed our perspectives of the genetics field. The human genome is all about biological information, and the information provided is on a grand scale. With advances in computing technology providing large capacity for storage, this biological data is able to be collected and stored. Since the size of the biological data is extremely large and is still increasing dramatically, biologists need to rely on computer analysis tools to convert this immense store of biological data into the knowledge we needed. The needs for computer science from the biological field has created a new field called *bioinformatics*. To date, bioinformatics has become essential for biological experiments, data management, and data analysis applications using available biological data.

In this chapter, we introduce bioinformatics and describe Microarray data analysis. Particularly we describe the concept of gene expression and Microarray technology. After the introduction of Microarray data analysis, we present our research problems and research objectives. Finally, we briefly list our contributions and the structure of the thesis.

1.1 Bioinformatics overview

Bioinformatics, a new research field which resorts to computer technologies to investigate biological problems at the molecular level, has grown up in the last twenty years or so due to the advance in computer technologies and the explosive growth of biological data [32]. The development of bioinformatics is driven by the accelerated growth of biological data, and the activities of bioinformatics involves researching and developing the application of computational tools for collecting, manipulating, analyzing, and transmitting huge quantities of biological data.

According to Genbank statistics from the National Center for Biotechnology Information website (NCBI: `http://www.ncbi.nlm.nih.gov/Genbank/genbankstats.html`), the number of entries of gene sequences in databases has grown from 606 in 1982 to 52 million in 2005. The production rate of biological data is largely driven by rapid changes in computer technologies such as larger data storage capacity and cheaper and faster computing hardware. Disk storage capacity doubles every 9 months, and processing capacity doubles every 18 months [58]. These technological advances have made available the storage capacity and processing power required for many applications including biological research.

Apart from the driving force of computer technologies, the development of bioinformatics has also been driven by the revolution in biology itself. The problem was, in the past, that scientists were able to study only one or a few genes at a time, and this severely restrained scientists from examining entire genomes. As a result, this made it impossible for scientists and biologists to reveal the interactions among the entire genome. Researchers demand the study of many genes together in order to discover useful information which is hidden among genes, for example the specific roles of individual genes in various genetic diseases, such as breast cancer or heart disease [144, 66]. Other useful information can be the gene patterns for the cause of a disease and the similarities among genes and the functionalities of genes [102]. Toward this end, the U.S. Department of Energy and the National Institutes of Health has launched the Human Genome Project (HGP) from the early 1990s which aims to identify the genes in human DNA and determine the sequences of the billions of chemical base pairs that make up human DNA. The project was successfully completed in year 2003. The Human Genome Project provided a large amount of genetic information facilitating our understanding of the genetic structure of human beings. We know now that there are around 20000 - 25000 genes in a human cell, according to the International Human Genome Sequencing Consortium [33]. The sequence of the human DNA is stored in databases which are available to the public. All these data provide a foundation for further study by other biological researchers. As a direct consequence, the complete genome information has provided the opportunities for biological researchers to study not just of single or a few genes, but the functions and interactions of all the genes in the whole genome. No doubt, these data are tremendously beneficial to biologists studying all aspects of human beings. The vast amount of valuable biological information provides an opportunity to analyze the interaction between genes; to detect possible mutation genes; to investigate certain genes which might cause certain cancers and compare genes between normal and cancer tissues. And most importantly, the discovered information will be useful for finding effective treatments and preventing the potential cancer patients from suffering the disease.

Despite the human genome project being finished, the problem to realize its full potential is challenging. From this moment on, biology research has entered into a post-genetic era. In the post-genetic era, It is noticeable that biology is no longer a traditional science which has little or no connection with computer science. Since the massive biological data can be extremely useful only if biological data can be converted into knowledge, biologists now need help from computer science to implement tools for effective accessing and maintaining the vast molecular biological database. In addition to computer capacity, biologists also need computing software such as biological database analysis algorithms that discover the information behind the generated biological data more accurately and consistently. This situation has created the need for a new generation of tools for extracting knowledge automatically and effectively in order to understand this biological information.

Biological databases are essential to Bioinformatics research. To be able to handle and share the wealth of biological data and support biological data analysis, these biological data are stored in different biological databases for easy access. Note that these biological databases contain necessary information to assist biological researchers to understand (a) the functions of proteins [120], (b)the evolution of living organisms [78], (c) the cause of diseases [16, 31]. Ultimately the discovered information will be beneficial for finding new strategies for improving human life. For example, the knowledge can be helpful for identifying genetic risk factors for genetic diseases. Consequently, these risk factors can assist doctors to detect the disease in the early stages. More ambitiously, the knowledge can be used to discover the groups of genes for causing breast cancer, we then might find a way to modify the identified genes. Furthermore, if we can determine which group of patients react positively to a certain drug with little side effects, then we should be able to apply different drugs according to which group the patient is in, in order to fight against cancer diseases more effectively.

Depending on the content of biological data stored, biological databases can be divided into: protein sequence databases, protein structure databases, genome sequence databases and Microarray databases [14]. As we are aware, in past decades these biological databases have been used in many bioinformatics research areas, such as comparative genomics, systems biology, structural biology and Microarray data analysis.

Comparative genomics [122] studies the available whole genome sequence of living beings by comparing with genetic material of itself or other different species. According to the theory of Charles Darwin's evolution, human beings are related to ape-like ancestors and it indicate that some species should be more or less related to each other. The living beings with common ancestors, homologous, should have similar DNA while the species with different ancestors should exhibit less similarities at the DNA level [32]. It is interesting to know how different or similar they are at molecular level and how major evolutionary changes take place [132]. Comparative genomics has been applied for revealing unusually long motifs in mammalian genomes [71]. It also been used for identifying the cold-induced genes among different crops in order to improve the freezing tolerance in plants [87].

Systems biology [8] aims to look at the big picture of everything going on in the cells and biological systems of living beings. Instead of studying protein activities, gene expressions and the variations proteins and genes in living beings in isolation, systems biology combines all proteomic, genomic and other information into an integrated picture of the logic of living beings. Many systems biology researchers have focused on building computer models for simulating biological system process [104, 37], such as the brain system [36] and the biological cell [129].

Structural Biology [56] is used to determine the structure of proteins from their amino acid sequences. Proteins carry out most biological activities in cells. Each protein folds into a unique three-dimensional structure held together by chemical interactions between the amino acids in order to perform its particular activity or functionality. In recent years, a flood of protein sequence data has been produced, and this creates the possibility for bioinformatics researchers to predict the structure of proteins in order to help biologists have a better understanding of the functionalities of proteins. Structural biology encompasses developing effective computer-based tools or methods for protein structure prediction. Structural biology is very useful for many research areas, such as drug design [143] and protein modeling [137].

Last but not least, Microarray analysis. Revolutionary breakthrough Microarray technologies provide an efficient, high-throughput way of producing vast of gene expression Microarray data. Microarray technologies enable scientists to study the whole genome with a single experiment by providing a broad snapshot of the state of a cell by measuring the expression levels of thousands of genes simultaneously. Those technologies accelerate the pace of every aspect of biological research. As more and more Microarray data become available, this enables biological researchers to obtain answers to many complex questions. For example, if it is a simple fact that a particular gene is known to be involved in a disease, then other unknown genes with similar expression patterns are very likely to have similar functionality. This definitely helps biology researchers to discover the genes of unknown function with co-regulate genes of known function. Generally, many diseases are affected by more than one gene. That means a group of genes normally acts together to contribute to a disease or change in state. With Microarray data, it is possible to reveal the relationship between those genes within a group as well as between groups. Ultimately, Microarrays create the potential to help discover the cause of human cancer diseases based on pattern differences between diseases and healthy people.

Microarray analysis focuses on identifying the sequences of genes and determining the expression of an abundance of genes. Its applications are mainly divided into single nucleotide polymorphisms Microarray analysis, and gene expression Microarray analysis.

Single nucleotide polymorphisms (SNPs) Microarray analysis focuses on genetic variations and mutation [25]. DNA contains genetic information coded by the bases A,G,T and C. A single mistake or change in the DNA code can therefore result in the expression of some important differences in living beings. A single nucleotide polymorphisms (SNP) occurs when a single nucleotide in the genome is substituted. For example, two sequenced DNA fragments from different individuals, AAGCCTA to AAGCTTA, contain a difference in a single nucleotide. The DNA at this locus is disrupted due to SNPs, and consequently, protein synthesis is also disrupted, producing the different phenotype, a disease for example. A very important task of SNPs analysis is to understand how the DNA sequence variation affects protein function, and to determine if a particular SNP is associated with disease susceptibility in a individual [110] . The results can help researchers identify useful markers and which regions of the genome are involved in certain diseases. It has also been used for genetic linkage analysis [61].

Gene expression Microarray analysis [130] aims to analyse gene expression profiling. Studying gene expression profiling in different cell types based on the entire genome helps scientists to identify novel genes associated with certain cancers, to answer questions such as: what kind of genes contribute to a normal cell turning to a certain type of tumor cell, how genes interact with each other, and predicting patient outcomes. In our research, we concentrate on gene expression Microarray analysis.

1.2 Gene expression Microarray analysis

In this section, we describe a few fundamental concepts including *gene expression*, *Microarray technology*, *gene expression Microarray data*. We also point out the potential applications of the gene Microarray data analysis in medical and cancer research.

1.2.1 Gene expression

As we know, the basic units of all living organisms are cells. Each cell can sustain and replicate itself. Moreover, proteins are the structural components of cells and perform most of the critical functions of biological systems. The construction of proteins is controlled by genes which are encoded in DNA. DNA, standing for deoxyribonucleic acid, contains the genetic instructions to direct the cell for making the required proteins. The instructions contained in DNA are deciphered by ribosome molecules in cells. After the decipher, various proteins are created accordingly to sustain live organisms.

DNA is made up of four building blocks called bases: adenine(A), thymine(T), cytosine(C) and guanine(G)(see Figure 1.1). These building blocks make up nucleotides, together with a 2'-deoxyribose sugar and a phosphate group. Nucleotides are linked to each other forming a DNA polynucleotide. As DNA is double-stranded, two polynucleotides are then joined together by hydrogen bonds between the bases by base-pairing rules to form a double helix. The base-pairing rules are: A is base-paired with T, and G is base-paired with C.

The information encoded in genes is ultimately used to construct proteins. The construction of proteins from genes is done in two steps: transcription and translation. The process of constructing proteins from genes is called gene expression.

Figure 1.2 shows the process of protein synthesis from DNA. During the transcription step, a single strand of messenger ribonucleic acid (mRNA) is synthesized using the gene where the genetic information needed to synthesize a protein encoded in DNA as a template. mRNA carries the information out

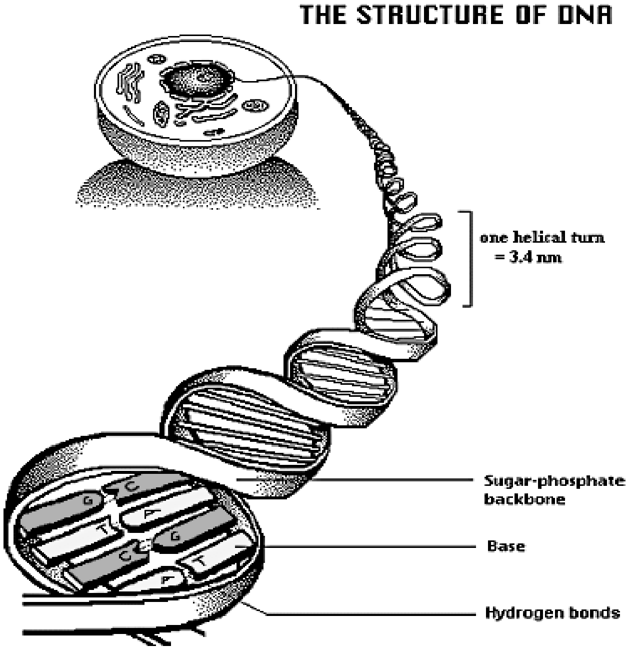


Figure 1.1: A sample of DNA helix from [1]

of the nucleus to the cytoplasm where protein synthesis takes place. The mRNA is similar to a single DNA strand with A, G, C bases, the difference is that uracil (U) replaces T. After the transcription step, the unique sequence of bases has been copied into mRNA.

Proteins are linear chains of arranged amino acids [23]. There are a total of 20 amino acids that can be incorporated into a protein. From the beginning of an mRNA strand, a sequence of three bases specifies a particular amino acid. For example, the amino acid glycine is coded on mRNA as GGU while lysine is coded as AAA. The genetic code is read during translation via transfer RNAs (tRNAs). Each tRNA contains three bases complementary to the corresponding three bases on mRNA and carries an amino acid encoded

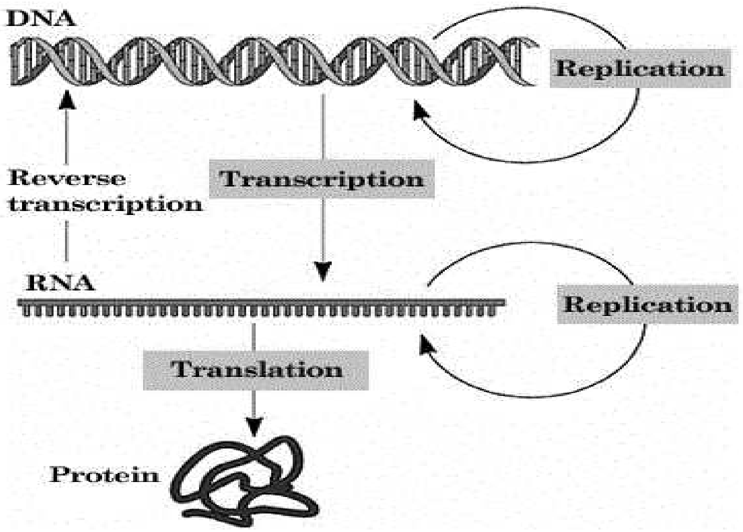


Figure 1.2: Protein Synthesis from DNA in [2]

in mRNA.

During the translation step, the unique sequence of bases copied from DNA is read by a ribosome molecule through mRNA. Ribosome reads three bases at a time, then picks up a tRNA which has a complementary sequence of bases. The matched tRNAs joins together with the mRNA chain. The ribosome takes the amino acid attached to the tRNA off after the mRNA and tRNA are joined together. The carried amino acid is then put into a chain of amino acids. After the entire mRNA chain has been read by the ribosome, the mRNA nucleotide sequence is translated into an amino acid chain, and this chain is called a protein.

Genes encoded in DNA are used to construct proteins which determine our inherited traits, such as hair color, behavior and our health. The proteins constructed then determine the traits of the living organisms which are inherited from their ancestors, such as hair and skin colors, body height, shape, and diseases. Due to genetic variations and mutations, some genes are not dominant. That means they are passed on to their offspring, but their traits are not expressed. These are called recessive genes. For example, a parent who carried breast cancer might have a very healthy baby. It does not mean the baby did not carry on the disease gene, but the gene was not dominant. In addition, some diseases are not determined by one gene, but several genes acting together. For instance in the case when several dominant and recessive genes act together to trigger the certain diseases such as breast cancer and heart disease.

Therefore, to discover cause of the cancer and other diseases, it is vital to know how genes react, such as over-expressed(dominant) or under-expressed(recessive), in those disease cells and what the relationship between them is when a disease occurs.

With all human genomes having been mapped and sequenced by the Human Genome Project, the next step would be how to apply this information for analysis and diagnosis of cancer diseases.

1.2.2 Microarray technologies

In order to determine the state of genes - over-expressed or under-expressed, new Microarray technologies [128] have been developed. These Microarray technologies are mainly used to conduct gene expression profiling. Such technologies have become a powerful tool in Microarray analysis. By analyzing microarray measurements of gene expression profiles we can achieve a better understanding of cancer diseases .

A gene expression Microarray is formed by placing tens of thousands of candidate gene sequences in discrete spots on a glass slide or a silicon chip. Each spot represents a known gene. Depending on how the length of DNA sequences is used and how DNA sequences are laid down, Microarrays can be divided into cDNA Microarrays [115] or Affymetrix Microarrays [91]. In this research, we focus on cDNA gene expression Microarray data.

cDNA, complimentary DNA, is a reverse transcribed DNA which is derived from mRNA shown in Figure 1.3. The level of gene expression measured in a Microarray is defined as the measure of the abundance of transcribed mRNA during the construction of protein. However, due to the unstable nature of mRNA which can easily degrade, cDNA is widely used in experiments since cDNA is more stable then mRNA.

Here we cite a comparative hybridization experiment with a cDNA Microarray. This experiment is conducted in six steps as shown in Figure 1.3.

1. Collecting normal and disease samples.

2. Collecting mRNA from two different samples.

3. Labeling mRNAs with different fluorescent dyes.

4. Mixing two samples on the Microarray slide.

5. Reading the results.

6. Storing gene expression profiling into an image.

To conduct a Microarray experiment, gene expression Microarray is prepared first by spotting cDNA gene probes to a glass slide using a robotic printer.

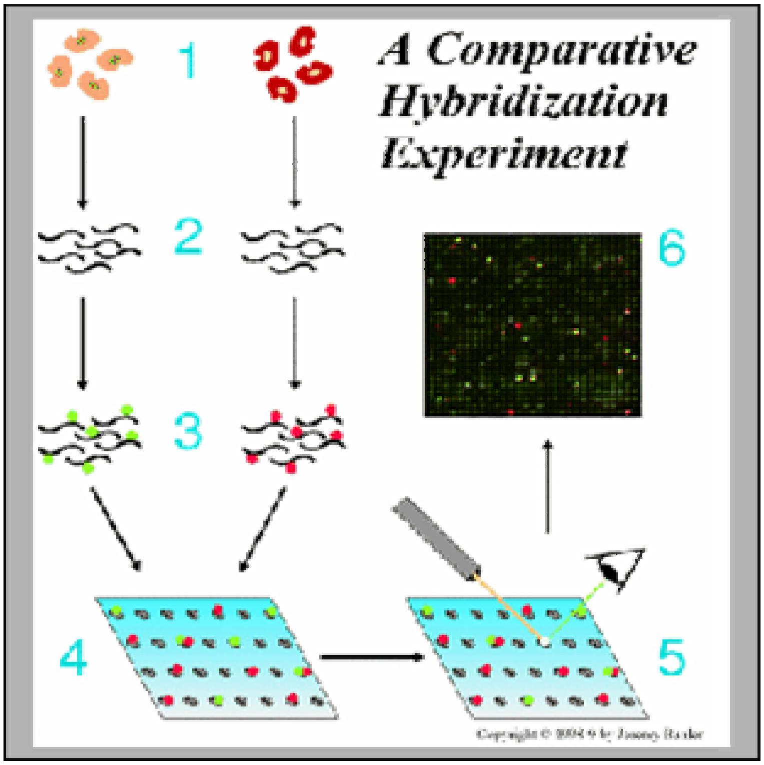


Figure 1.3: A comparative hybridization experiment with cDNA Microarrays from [3].

Next, mRNA are collected from two different cells (sample and test). The sample and test mRNA are then labeled with different fluorescent dyes, green and red respectively, and they are reverse transcripted to colored complementary DNA (cDNA). Two colored cDNA are mixed together. The mixed cDNA is incubated with Microarray on the slide. Each DNA spot on the slide can pair with a cDNA. Some of the samples will bind to spots where they find their complimentary pairs. Unbounded samples are washed off after a period of time. The Microarray slide then is scanned with a green laser and then a red laser to detect the abundance of cDNA. The two images are then merged together. Since we know which gene each spot represents, and the cDNA only sticks to the gene that is complimentary to it, we can determine which genes are turned on in the cells.

In summary, Microarray technology is a significant advance in biological research because it enables biologists to look into gene profiling to discover the relationship between genes, such as relationship between over-expressed genes, between over-expressed and under-expressed genes, and between under-expressed genes.

1.2.3 Gene expression Microarray data

Gene expression Microarray data is presented as a table with a collection of expression levels of genes under different samples (patients for example). An example of a gene expression Microarray dataset is shown in Table 1.1. Note that, such a sample may have thousands of attributes. The table organizes data into columns and rows (called samples). Within the table, a row represents a sample and a column represents an attribute. The columns contain a set of gene (or feature) expression values and a category (a.k.a class) value. Each feature contains the expression levels of a single gene for every sample. The category value divides the samples into different categories. Each sample in that table contains information about the expression levels of all genes with a consequent category.

For example, Table 1.1 represents a part of the breast cancer gene expression Microarray data set. There are total of 7 samples in the data set. Each sample contains the gene expression values of 7 genes such as the `NM_020120` gene. In Sample 1, the expression value of the `NM_020120` gene is 0.112. There are two categories, and they divides the samples into *relapse* and *non-relapse*.

Sample	NM_020120	NM_005744	NM_014003	NM_020974	NM_013438	D13540	NM_005915	Categary
Sample 1	0.112	0.039	0.002	-0.811	0.066	0.066	0.008	relapse
Sample 2	0.192	0.115	0.172	-0.981	0.101	0.293	0.487	relapse
Sample 3	-0.004	-0.045	-0.069	-0.124	-0.071	-0.042	-0.264	non-relapse
Sample 4	0.388	0.034	0	-1.194	0.171	0.088	0.066	relapse
Sample 5	-0.052	0.062	0.052	-0.628	-0.004	0.063	-0.104	non-relapse
Sample 6	-0.036	-0.005	-0.072	0.881	-0.097	-0.164	-0.293	non-relapse
Sample 7	-0.202	-0.075	-0.142	0.034	-0.003	-0.012	-0.054	non-relapse

Table 1.1: An gene expression Microarray dataset

Gene expression Microarray technologies can be used to compare gene expression in two or more different types of cells (eg. disease and normal) by measuring the level of expression of thousands of genes to allow identification of genes that are over-expressed or under-expressed. For example, gene expression Microarray technologies have brought great potential for cancer research. A practical application of gene expression Microarray technologies is

1. classify different diseases according to different expression levels in normal and tumor cells,

2. discover different subtypes of cancer,

3. reveal the function of novel genes based on similarities in expression patterns with those of known genes,

4. identify marker genes which play a critical role in the development of disease.

The research findings will ultimately play an important role in diagnosis of cancer patients, predicting new patients, and most importantly for discovering an effective treatments for the cancer patients.

The development of new gene expression Microarray technologies raises the need for new and more sophisticated bioinformatics methods to cope with the vast amount of data generated, and to convert useful information from those available Microarray data.

1.2.4 Gene expression Microarray data analysis applications

There are many active research applications currently aimed at taking advantage of the full potential of DNA Microarray technologies. The most prominent applications include cancer classification [117, 96], gene function identification [92, 114, 125], clinical diagnosis [138, 134], and drug discovery studies [94]. One of the most active practical applications of Microarray technology is cancer classification analysis [123, 124, 69, 57, 6]. Traditionally, scientists classify different types of cancers based on the organs in which the tumors develop, but gene expression Microarray technology can be used to classify subtypes of those cancers based on the patterns of gene expression level in the tumor cells. Note that such patterns could be used to diagnose possible future new cancer patients and design treatment strategies targeted directly to each specific type of cancer. In addition to cancer classification, gene expression Microarray technology also can assist scientists to develop more effective cancer treatments by analyzing the differences in gene expression levels among tumor cells using different drugs.

Gene expression Microarray data analysis has been the subject of intensive investigation and has been studied from different research fields such as data mining, machine learning, and statistics. As summarized by Sun-Bae Cho and Hong-Hee Won [29], the analysis of Microarray gene expression data is mainly about classification, gene selection and clustering.

It is common that a Microarray data contains a large number of genes. Not all genes however are relevant to the analysis. The fundamental task of gene selection is to select the most informative genes which are most predictive of their related class for classification. Many gene selection methods have been developed such as correlation coefficient [124], Markov blanket [75], and Chi-Squared [89].Gene selection methods have been widely used to reduce the dimensionality of gene expression data in order to improve the performance of classification [74]. Gene selection can also be applied for identification of novel gene markers [92, 125].

Clustering involves discovering new classes and grouping the genes with similarities in gene expression data where no categories are specified for samples. Some of these methods include hierarchical clustering [46], k-means algorithms [127], self-organizing map [68, 127] and etc. Clustering methods have been used to discover new classes in Microarray data analysis [46, 123, 118]. Clustering methods have also been used to cluster genes to find co-expressed genes and to support gene selection [103, 72].

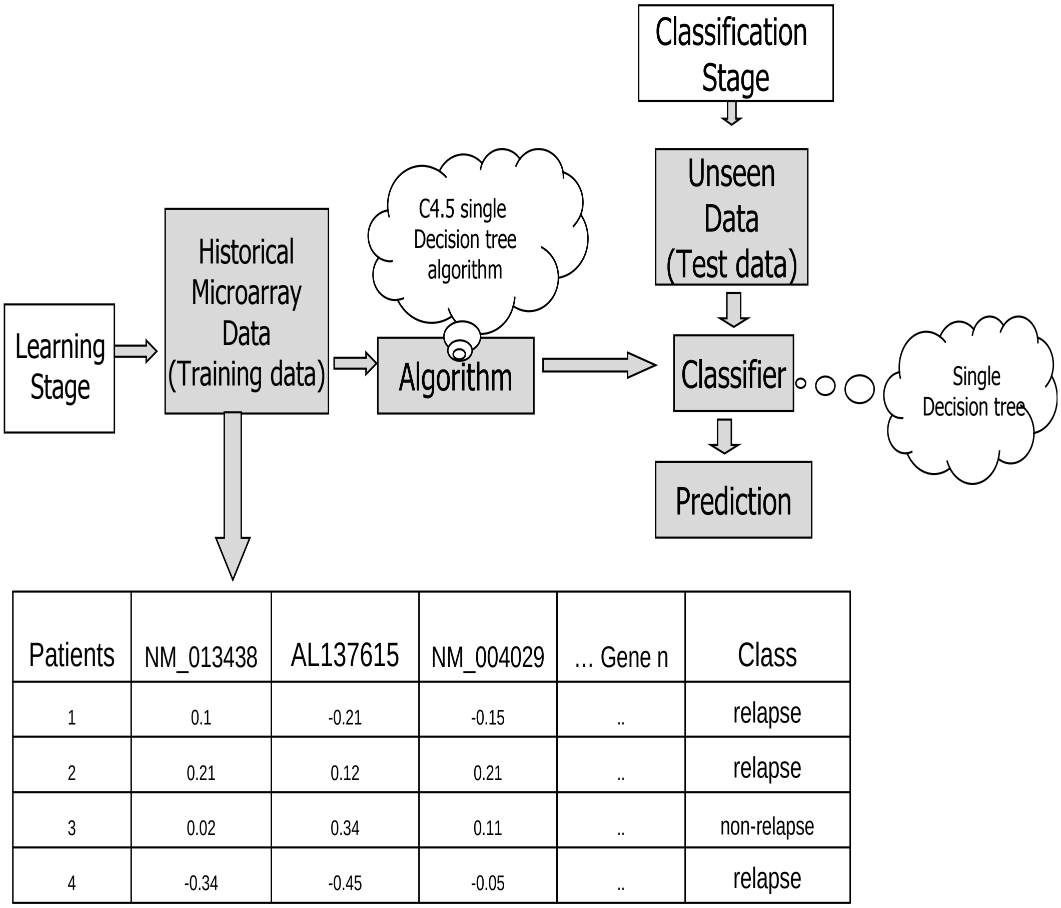
In this research, we will concentrate on gene expression Microarray data classification problems.

1.2.5 Microarray data classification

Microarray technologies have proved to be indispensable for providing new efficient and effective ways of diagnosing cancer diseases. The goal of Microarray data classification is to build a classifier from categorized historical Microarray gene expression data, and then to use the classifier to categorize future in-coming data or predict the future trend of data. These methods encompass support vector machines (SVMs) [21, 22, 59], k-nearest neighbor classifier [138], bagging and boosting [13], decision tree based methods [20] and Bayesian networks [44] etc. Classification has been extensively used in cancer research for classifying and predicting clinical cancer outcomes [123, 124, 69, 57, 6]. It is also applied to cancer diagnosis and prognosis [138, 134]. In addition, classification can help researchers to discover the drug response for particular patients in order to use appropriate treatment for individuals [94].

Like Machine learning and Data Mining [135], the objective of Microarray classification is to extract implicit, previously unknown, and potentially useful information from existing Microarray data. To apply gene expression Microarray data analysis, we must have historical gene expression data handy, as well as effective and efficient gene expression Microarray data classification algorithms.

Microarray data used for Microarray data classification is usually stored in relational databases or data sets as shown in Table 1.1. In this thesis, we will use the database and data set interchangeably. For the purpose of Microarray data classification, Microarray data will be divided into two sub databases - training data and test data.



Patients	NM_013438	AL137615	NM_004029	... Gene n	Class
1	0.1	-0.21	-0.15	..	relapse
2	0.21	0.12	0.21	..	relapse
3	0.02	0.34	0.11	..	non-relapse
4	-0.34	-0.45	-0.05	..	relapse

Figure 1.4: The process of single decision tree classification

Microarray data classification usually involves two stages: learning and classification. Figure 1.4 depicts the C4.5 decision tree algorithm, which is a benchmark classification algorithm used in the machine learning and data mining fields. Microarray data is divided into training and testing data before or during the training stage. In the training stage, a classifier is deduced from the training data. In the classification stage, previously unclassified data, test data, is subjected to the deduced classifier. The class values are removed from the test data set. By using the classifier which has been discovered from the training set, each record will be assigned a predicted class.

The quality of the classification is reflected in how accurate the predicted classes match the true classes of the tested records. The accuracy here is estimated by the percentage of the test records that are assigned correctly by the classifier.

1.3 Research problems

Microarray analysis technologies have opened up many windows of opportunity to investigate cancer diseases using gene expressions. Bioinformatics has bridged the gap between research in biology and novel computing technologies. Along with this have come many new research problem areas, such as: comparative genomics, systems biology, structural biology and Microarray data analysis. In our research, we will concentrate on Microarray data analysis and Microarray data classification in particular. The fundamental task of Microarray data classification is to find a computational model from the given Microarray data that can determine the category of unknown samples. The key elements of classification are accuracy and quality or robustness. The accuracy of Microarray data classification relies on both the quality of the provided Microarray data and the used algorithms for Microarray data classification. No doubt if the Microarray data provided are wrong or partially incorrect, the model deduced from it will be either false or less accurate. Microarray data analysis based on incorrect Microarray data will definitely mislead the users towards a wrong decision. Moreover, the quality of Microarray data classification also depends on the Microarray data classification algorithms used for the microarray data classification. With existing data, quality of Microarray data classification algorithms are required to interpret the raw biological data into meaningful information in order to assist biologists in fighting diseases or other biological problems. We can conclude that the ideal situation for Microarray data classification will be a high quality Microarray data classification algorithm along with an error free Microarray database.

However, the reality is that Microarray data is not error free. The most current Microarray data classification methods are struggling to deal with the curse of dimensionality and noise problems.

1. *Curse of dimensionality.*

   Microarray data contains large number of genes with a small number of samples. A real world gene expression Microarray data set suffers from a so called "curse of dimensionality" problem, ie. a huge number of genes (more than 20,000 for humans) with a small number of samples. Many existing classification methods such as association and classification methods [88, 86, 140] are prevented from directly dealing with this type of database due to computation costs and computer memory limitations. Furthermore, the insufficient number of samples is a major cause of low accuracy data classification. More attributes in a Microarray database require more samples to retain the accuracy of classification. However, Microarray databases normally contain less than two hundreds samples due to high experimental costs. This is far less than the minimal requirement for Microarray classification. As a matter of fact, Microarray classification based on a small number of samples can not generalize the classifier well, and this therefore decreases the accuracy of the Microarray data classification.

2. *Noise problem.*

   Microarray data contains errors including incorrect or missing data values. As existing Microarray data contains a massive number of genes, it is inevitable that some errors appear in the Microarray data sets. These errors can be introduced at any stage during generation of the raw Microarray data. For example, errors can be made by humans when they enter the data; some errors can also be introduced systematically if the inappropriate algorithm was applied when the raw data were transferred into the database from the Microarray data. Due to the prematurity of Microarray technology, it is common that some errors are introduced during the production of the Microarray data itself. The high level of noise is one of the key issues of Microarray data classification. Many efforts have been made by researchers from different fields, either to minimize or correct the errors being introduced during the production and maintenance of Microarray data. These attempts are certainly helpful for providing a reliable data source for Microarray data classification. However, no matter how hard we try to prevent the errors from being introduced, the truth is that some errors will always be present in the Microarray data. After the Microarray data has been generated and stored in the database, there is no way for the users to validate the quality of the Microarray data. In current practice, researchers normally apply databases from some trusted research institutes, or from high ranked journals or from trusted websites. But despite this it is still not possible to guarantee the quality of the database used for Microarray data classification. In reality, the quality of Microarray data we used for generating the models is not equal to the quality of Microarray data we are going to test on. That means the Microarray data sets contain different levels of noise. This quality uncertainty is one of the major considerations in the design of new Microarray data classification methods. And how to eliminate the effect caused by the uncertainty of noise data during the Microarray data classification stage remains a challenge for many existing classification algorithms.

In summary, accuracy and noise resistance are major issues for classifying gene expression Microarray data.

1.4 Objectives

To deal with the problems we identified in the last section related to gene expression Microarray data classification, we will design a novel Microarray data classification method that overcomes the problems associated with high dimensionality and high noise.

In this thesis, we are trying to focus on the following aspects to tackle the problems that exist in most current Microarray data classification. The objectives of our research are as follows.

1. to design and implement a robust Microarray data classification algorithm to classify gene expression Microarray cancer data more effec-

tively and efficiently. The proposed algorithm is based on traditional machine learning and data mining methods. Our algorithm should be robust enough to handle a high level of noise. It is also scalable to large Microarray data sets without any difficulties. The results of classification should be easily understandable by researchers.

2. to investigate the noise resistance capability of various gene expression Microarray data classification methods. As we know noise values do exist in all gene expression Microarray data, and robustness is a very important issue for all reliable gene expression Microarray classification algorithms. We compare the robustness of our newly developed algorithm to other gene expression Microarray classification methods.

3. to study the effectiveness of gene selections for various Microarray data classification methods. To improve the quality of Microarray data, the noise and irrelevant genes should be reduced to a minimal level before classification is carried out. The question is whether any given gene selection method can improve the performance of a Microarray classification algorithm. In this thesis, we study the dependency between gene selection methods and Microarray data classification methods in relation to enhancing the performance of Microarray data classification.

1.5 Contributions

We designed a diversified multiple decision trees algorithm. The proposed algorithm significantly improves the performance of existing classification algorithms in terms of accuracy and robustness.

The contributions of this research include:

1. *A robust Microarray classification algorithm.*

   The development of a robust decision tree-based classification prototype system for gene expression Microarray data analysis. This system is able to deal with noise data more effectively then current classification methods. The method is implemented in Perl and C programming language. In addition, this method has been integrated with the Weka package, and can be compared with other benchmark systems built in with the package.

2. *Examination of the robustness of gene expression Microarray classification algorithms.*

   We strongly believe that the robustness issue is equally as important as the accuracy issue in Microarray data classification. We confirmed that the diversified ensemble method by avoiding overlapping genes is more robust than other ensemble and non-ensemble methods in Microarray data classification, since the noise affect has been reduced to a minimum in this way.

3. *Comparison of dependence between gene selection methods and Microarray data classification methods.*

   We theoretically compared and revealed the relationship between gene selection methods and Microarray classification methods. We analyze various gene selection methods for improving the performance of Mircorarray classification algorithms. We compare the filter and wrapper gene selection methods. We discovered the relationships between gene selection methods and Microarray classification algorithms. We summarized several criteria for how to effectively choose and apply gene selection methods for Microarray classification.

1.6 Structure of the dissertation

This dissertation has been organized into 8 chapters. The rest of this dissertation is organized as follows.

In Chapter 2, we present the design of methods for comparing the accuracy and robustness of ensemble decision trees classification algorithms. We introduce the data sets that were used for the experiments. These all came from reliable sources and have been used for various publications. We also describe our noise generation method. This method is able to systematically generate noise data at any required level. To test the robustness of compared methods in different situations, we deliberately increase the noise level on both training and test data.

In Chapter 3, we introduce the tree based classification methods and problems. Robustness reflects the ability of a classification method to deal with noisy data. Single decision tree methods are not robust in Microarray data classification. In contrast, the traditional ensemble methods attempt to achieve diversified trees in an ensemble committee in order to increase the accuracy and robust performance of classification.

In Chapter 4, we compare some existing ensemble decision tree methods. Two well-known ensemble methods using re-sampling samples, Bagging and boosting, are described first. We discuss the advantages and disadvantages of both methods. We conduct an experiment and point out the robustness limitation existing among those methods. We introduce some improved ensemble decision trees methods based on traditional single and ensemble decision tree methods. The huge number of genes in Microarray data poses a number of problems in Microarrray classification. However, they are useful if we use them to generate diversified trees. In fact, it is feasible to create aggregated ensemble decision trees through re-sampling genes instead of re-sampling scared samples. To date, many newly developed ensemble methods have constructed ensemble committees by using different gene selection methods in some classification stages. CS4 and Random forests are described in this chapter. We compare their robustness with traditional ensemble methods described in Chapter 3.

In Chapter 5, we present our newly developed ensemble decision trees method called diversified multiple decision trees algorithm (DMDT). Among the current Microarray classification methods, most of them have not given special consideration to the robustness issue of Microarray data classification in terms of dealing with the uncertainty of Microarray data in training and test data sets. Neglecting the robustness for the designing of Microarray data classification significantly impact the accuracy performance of Microarray classification. To address the disadvantage of Microarray classification existing in current ensemble classification methods, we implement a new robust Microarray classification algorithm with minimal impact on accuracy performance based on different levels of noise data in the same data set. This design ensures that more highly accurate classification performance is achieved not only on good quality data but also on the same data with decreased quality, or more noisy data.

In Chapter 6, we present our robustness evaluation for some well-known ensemble and single decision trees methods. We emphasize that the meaning of robustness has two fold - a good robust classification algorithm can not only resist noise data at the present level, but also tolerate noise data at an increased level. We conduct an experiment and discuss the results.

In Chapter 7, we investigate the dependence between gene selection and Microarray data classification algorithms. Gene selection method is helpful for increasing Microarray data classification performance. There is not a great deal of evidence to show the relationship between gene selection methods and Microarray data classification methods, but in the hope that gene selection methods can improve the accuracy performance of classification regardless. So we conduct an experiment to reveal the true relationships between gene selection methods and classification methods.

In Chapter 8, we conclude the dissertation and outline future work. We can see from this research that there are plenty of opportunities for further interesting research in this field.

1.7 Summary of the chapter

In this chapter, we introduced bioinformatics, Microarray technologies and gene expression Microarray data analysis methods. We have given an introduction to some problems in gene expression Microarray data and current Microarray data classification. We outlined our objectives, contributions and structure of this dissertation. In the next chapter, we will introduce the experimental design and methodology used in this thesis.

Chapter 2

Microarray datasets and research methodology

The feasibility of an ensemble decision tree method is determined by its accuracy and robustness. Our methodology is to compare the accuracy and robustness of these different classification methods on some gene expression Microarray data sets. In Section 1, we elaborate the gene expression Microarray data sets used for the experiments. In Section 2, we describe an estimation of accuracy performance of Microarray classification methods. In Section 3, we introduce the evaluation of robustness performance method. In Section 4, we describe the software used in the experiments. In Section 5, we summarize the chapter.

2.1 Gene expression Microarray data sets

In our research, we concentrate on the gene expression Microarray data sets which were introduced in Chapter 1. Six gene expression Microarray cancer data sets from Kent Ridge Biological Data Set Repository [82] are selected. Table 2.1 shows the summary of the characterustics of the six data sets. Each Microarray dataset is described by the following parameters.

- Genes: the number of genes or attributes,
- Class: the number of classes,
- Record: the number of samples in the dataset

Table 2.1: Gene expression Microarray data sets

	Dataset name	Genes	Class	Sample
1	Breast Cancer	24481	2	97
2	Lung Cancer	12533	2	181
3	Lymphoma	4026	2	47
4	Leukemia	7129	2	72
5	Colon	2000	2	62
6	Prostate	12600	2	21

Data set 1 is a Breast cancer data set [124] which contains 97 samples collected from patients with breast cancer. In this data set, 46 samples are from patients who had developed distance metastases within 5 years. The remaining 51 samples come from patients who remained free from the disease after their initial diagnosis for an interval of at least 5 years. The patients with distance metastases within 5 years are classified as relapse while the patients remaining free from the disease after their initial diagnosis for an interval of at least 5 years are classified as non-relapse. Each sample in this data set contains 24481 genes (attributes).

Data set 2 is a Lung Cancer data set [57] which contains 181 samples collected from lung cancer patients. In this data set, 31 patients have malignant pleural mesothelioma of the lung. The remaining of 150 patients have adenocarcinoma of the lung. The patients with malignant pleural mesothelioma of the lung are classified as Mesothelioma, while the patients with adenocarcinoma of the lung are classified as ADCA. Each sample in this data set contains 12533 genes.

The third data set relates to lymphoma disease. Each Lymphoma data set [6] has 47 samples collected from patients with different types of large B-cell lymphoma. In the data set, 24 patients are from germinal centre B-like type and 23 are from activated B-like type. The patients with germinal centre B-like type are classified as germinal while the patients with activated B-like type are classified as activated. Each sample in this data set contains 4026 genes.

Data set 4 is a Leukemia data set [123] which contains 72 bone marrow samples collected from patients who suffered from Leukemia. In the data set, 47 samples are ALL. The remaining 25 samples are AML. The samples with ALL are classified as ALL, while the samples with AML are classified as AML. Each sample in this data set contains 6817 genes.

The Colon data set [49] is collected from patients who suffer with Colon cancer. It contains 62 tumor samples collected from patients with Colon cancer. Among the 62 samples, 40 samples are from tumors and the remaining 22 samples are from healthy parts of the colons. The samples with tumor biopsies are classified as negative, while the samples with normal biopsies are classified as positive. Each sample in this data set contains 2000 genes.

Data set 6 is a Prostate data set [48] which contains 21 samples of patients with prostate cancer. In the data set, 8 samples are from patients who have relapsed within 4 years after surgery. The remaining 13 samples are from patients who have remained relapse free for at least 4 years or longer. The samples from relapsed patients are classified as relapse, while the samples from relapse-free patients are classified as non-relapse. Each sample in this data set contains 12600 genes.

2.2 Evaluation of accuracy performance

In this section, we give a definition of prediction accuracy. Then we review an accuracy estimation method: cross-validation.

2.2.1 Accuracy of test

In our research, a number of different ensemble Microarray data classification algorithms will be experimented on using the data sets described in Section 2.1. The objective of the experiments is to compare the estimated accuracy of the algorithms. Normally, we regard the algorithm with the highest accuracy as the best algorithm.

Definition 1. *Classification accuracy is defined as the percentage of correct classifications made from the total number of classifications by an algorithm* *based on a test dataset.*

$$\textit{classification accuracy} \; = \frac{\#\textit{correctly classified records}}{\#\textit{total records}} * 100(\%) \qquad (2.1)$$

Classification accuracy is an indicator of how many samples a classifier can correctly classify out of the total samples; # stands for the number of.

This definition is straightforward. For instance, if a classifier has a 99% accuracy, this means 99% of the records in the test dataset were predicted correctly.

The prediction accuracy[1] of a classifier has been seen as a major issue in Microarray data classification. Of course, a classification algorithm with higher accuracy is obviously more desirable. Prediction accuracy is a very important factor for measuring the performance of an algorithm. So it is vital that the estimated accuracy of a classifier approximate the true accuracy.

Note that a classification algorithm is usually trailed on some well-known Microarray data sets while under development. These well-known Microarray data are referred to as the training data sets. The classification accuracy derived from performing on the training data sets is called estimated accuracy.

2.2.2 Accuracy estimation

There are several reliable estimation of classification accuracy methods, such as cross-validation [116] and bootstrap [45, 47]. Among them, the cross-validation method has proven to be the more reliable method [19, 73].

[1] In much of the literature, the classification accuracy is also called predicative accuracy. Hence in this thesis we use them interchangeably.

The basic idea of cross-validation is to learn from some training data and then perform on the future as-yet-unseen data. This requires an independent data set to be used to estimate the accuracy performance of a classification algorithm. In our research, we adopt the ten-fold cross-validation technique, in which a data set is equally divided into ten folds(i.e. partitions) with the same type of distribution such as Normal distribution or Poisson distribution. After the partitioning, Nine folds of data are used as training data and one fold is for testing (unseen data set). The test procedure is repeated ten times. The classification algorithm will perform on the test data set ten times, and each run will generate a predication accuracy. The average of the ten prediction accuracies will be accepted as the final prediction accuracy of the classification algorithm.

2.3 Evaluation of robustness performance

In this section, we turn our attention to another performance index of classification methods - robustness. The objective of robustness analysis is to analyze how well a given algorithm can resist noise values, specially with increased levels of noise data.

Here for the sake of completeness, we present an informal definition of robustness.

Definition 2. *The robustness of a classification method is an indicator of how well the method can resist the noise in Microarray data. It is defined as the classification accuracy of the classification when the original data set are introduced with white noise at levels of 0%, 20% and 60%.*

We know the Microarray data sets described in Section 2.1 definitely contain various amount of noise data, but we do not know how accurate the data are, how badly the data have been affected by noise, and most importantly at what level the data contains noise. Therefore, in our experiments, we assume that the original data are perfect data with zero level of noise [65]. Then we add some noise to the original data. The way of adding noise onto the original data is to create a systematic algorithm which can generate different degrees of noise data, then add the noise data onto the original training and test data.

As we can see in Table 2.1, a Microarray data set organizes data into columns and rows (samples). The columns contain a set of gene values and a category value. Each column contains the expression levels of a single gene for every sample. Each row in that table contains sample information about the expression levels of all genes with a consequent class.

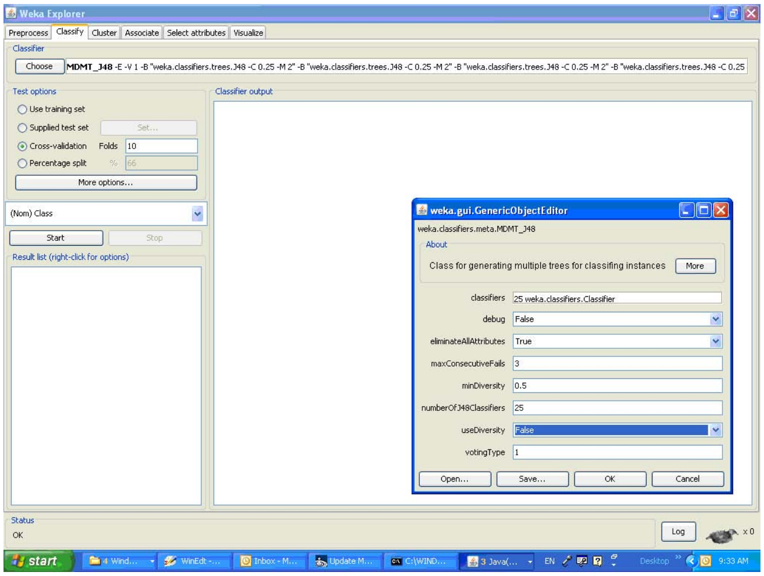
Here we elaborate the process of adding noise. In our experiments, the Polar form of the Box-Muller transformation method [39] has been used to generate noise data. The noise data is White Gaussian noise [2], that means the noise data is added independently to each gene in the original data set. In mathematical terms, let g be a gene expression level value of gene G in the original data set. The perturbed value of g will be $g' = g + n$, where n is generated using the Polar form of the Box-Muller method. Note that set of n has a mean of 0 and a variance of $d * \delta$ [98], where d represents the noise level while δ represents the variance of gene G in the original data set.

[2] White noise data is generated by a white random process. The white random process is represented as $n \sim N(0, d * \delta)$.

2.4 Related works and tools

Much work has been done in the data classification area. These efforts have resulted in many algorithms including decision tree algorithms. Decision tree algorithms are categorized into single decision tree and ensemble decision tree algorithms. In our research, we focus on 5 well-known single and ensemble methods, namely C4.5, Random Forests, AdaBoostC4.5, Baggingc4.5 and CS4. These algorithms have been implemented in perl or the Java programming language. We have done our experiments with all four algorithms apart from CS4 using the Weka-3-5-2 package which is available online (`http://www.cs.waikato.ac.nz/ml/weka/`).

Figure 2.1 is a snapshot of Weka.



MDMT_J48 -E -V 1 -B "weka.classifiers.trees.J48 -C 0.25 -M 2" -B "weka.classifiers.trees.J48 -C 0.25 -M 2" -B "weka.classifiers.trees.J48 -C 0.25 -M 2" -B "weka.classifiers.trees.J48 -C 0.25

○ Use training set

○ Supplied test set

● Cross-validation Folds 10

○ Percentage split % 66

(Nom) Class

OK

weka.gui.GenericObjectEditor

weka.classifiers.meta.MDMT_J48

About

Class for generating multiple trees for classifing instances

classifiers	25 weka.classifiers.Classifier
debug	False
eliminateAllAttributes	True
maxConsecutiveFails	3
minDiversity	0.5
numberOfJ48Classifiers	25
useDiversity	False
votingType	1

Figure 2.1: Weka Software GUI

We have done the experiments with CS4 using the software tool provided by Dr Jinyan Li and Huiqing Liu.

2.5 Summary of the chapter

In this chapter, we presented six Microarray data sets which have been collected by Kent Ridge Biological Data Set Repository. We described two performance indices of Microarray data classification algorithms - prediction accuracy and robustness, which are important factors affecting the practicality and feasibility of classification methods. After that, we listed some related software that supports single and ensemble decision tree algorithms and their tools. In the next chapter, we introduce the traditional ensemble decision tree methods which will be used for comparison with our newly developed method.

Chapter 3

Tree based classification and problems

In this chapter, we present a few Microarray data analysis methods. We also discuss their robustness of analysis when they are applied to the Microarray datasets we gave in Chapter 2. Although Microarray data analysis is a relatively new term, the technology which is used to do it is not. The purpose of gene expression Microarray data classification analysis is to diagnose and determine the prognosis of cancer patients. Therefore, it is crucial that the representation of classification is understandable or interpretable by users so they can use the discovered knowledge to help them in future decision making. This chapter is organized as follows. In Section 3.1, we overview the techniques used for data classification in the past. In Section 3.2, we account for one category of classification method: the single decision tree method. In Section 3, we discuss ensemble decision tree methods and the robustness of Microarray data classification. In Section 4, we introduce some traditional ensemble decision tree methods. In Section 5, we summarize the chapter.

3.1 Overview of classification techniques

Microarray data classification is a multidisciplinary field, and its techniques are drawn from areas such as machine learning, data mining, and artificial intelligence. There are many different classification methods for building classifiers such as decision tree based methods [107, 20], rule based methods [88, 95, 26], neural networks [112] and Bayesian networks [44]. *Neural networks* can be used as black box models but lack the transparency and interpretation of generated knowledge. The presentation of *Bayesian networks* is in the form of extremely complex graphs. This becomes cumbersome for gene expression Microarray datasets. *Rule based* methods use simple, intuitive and easily modified production rules, but they cannot handle databases such as gene expression Microarray data with huge attributes. In contrast, *decision tree* has a simple, intuitive representation of knowledge as a simple tree and is able to generate understandable rules, which are implications in the form of *if... then...*. They are therefore a popular basis for gene expression Microarray classification tasks.

3.2 Tree based classification method

According to the number of classifications generated, data classification methods can be catergorized into *single decision* and *ensemble tree method.* As the name suggests, a single decision tree classification generates one single classifier, which is usually represented as a decision tree for predicting the samples that have not been seen before. A decision tree is essentially a special data structure, which contains some decision nodes and leaf nodes [111].

To generate a decision tree, a classification algorithm recursively partitions a Microarray dataset into some disjointed subsets simultaneously, based on the values of an attribute. At each step in the construction of the decision tree, it selects an attribute which separates data with certain criteria such as the highest information gain ratio [108]. A good example is the C4.5 decision tree algorithm. We must be aware that the same process is repeated on all subsets until each subset contains only one class. In most cases such a decision tree is complex and redundant. To simplify a decision tree, the decision tree needs to be pruned using some criteria such as the pessimistic error estimation [108].

Figure 3.1 shows an example of a tree classifier. The C4.5 classification algorithm is used to classify the patients whose Microarray data were collected in the Breast Cancer data set that has been described in details in Section 2.1 of Chapter 2. Note that NM_* and AL* are all the attributes, and relapse or non-relapse are the classes. Each link from the root to a leaf in the decision tree is interpreted as a rule. Therefore, the tree contains three rules as follows.

1. `if (NM_013438 <= 0.14) and (AL137635 <= -0.12) then patient` = *relapse*. It means that if a patient has an expression level of NM_013438 under 0.14 while the expression level of AL137635 is under -0.12 , then the patient is likely to relapse after treatment.

2. `if (NM_013438 <= 0.14) and (AL137635 <= -0.12) and ( NM_004029 > -0.29) then patient` = $Non-relapse$. It means that if a patient has an expression level of NM_013438 under 0.14 while the expression level of AL137635 is above -0.12 and the expression level of NM_004029

is above -0.29, then the patient is unlikely to relapse of breast cancer after treatment.

3. if (NM_013438 > 0.14) then patient = *relapse*. It means that the patient is likely to relapse after treatment if the expression level of NM_013438 is above 0.14.

A decision tree classifier is often utilized to make a predictive class on a new record or sample. A decision tree classifies a new record by tracing it down the tree from the root to a leaf node choosing branches at each node according to the values contained in the new record. The specified class by the leaf node is assigned to the new record.

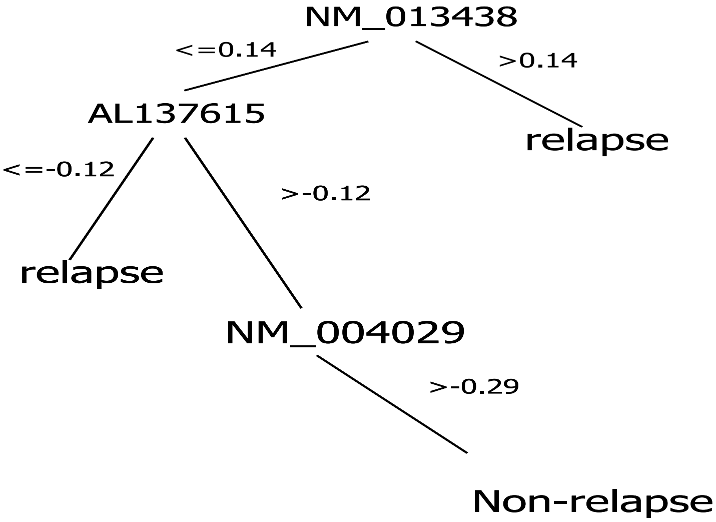


Figure 3.1: A tree classifier generated using the C4.5 algorithm

In the past few decades, many decision tree classifications [20, 11, 107, 24, 106], C4.5 [107] in particular, have become one of the most popular and successful classification methods in the machine learning and data mining fields [136]. In addition to its understandability, the C4.5 single decision tree algorithm has also several advantages as follows.

1. C4.5 is capable of dealing with different types of data. It can handle numerical values including continuous values and discrete values, as well as categorical values. This ability makes C4.5 more flexible for different types of data with less data preparation time for classification.

2. C4.5 is tolerant to and unaffected by redundant data. During the decision tree construction in C4.5, using the information gain ratio each node in the decision tree is selected if it can best separate the examples of different classes into different subsets and therefore can make the subsets as pure as possible. If one feature is selected, all other features in the data set are ignored. This characteristic helps eliminating the effect of redundant data in Microarray data classification.

3. C4.5 is more robust in handling data sets with missing values. For historic reasons, Microarray data often contains missing values. C4.5 handles this effectively by assigning missing data the value that is most common among training examples at the node [108]. In this way, C4.5 becomes more robust in handling data sets with missing values. To adapt C4.5 for Microarray data classification, we have to enhance this ability of Microarray data classification to deal with the missing data contained in Microarray data.

The good comprehensibility and reliability which the C4.5 decision tree method demonstrates, makes it a favorable choice which has been widely applied to many classification systems across different fields in the last few decades. For instance, C4.5 has been applied as an analysis tool to diagnosis skin cancer [9], breast cancer [124], lung cancer [57] and many other cancer diseases [145]. C4.5 also has been applied to machine fault diagnosis tasks such as bearing defects [79]. The application of the C4.5 algorithm can be seen in other domains. To name a few, C4.5 has been applied for recognition of printed text [7] and assessing the quality of certain types of text [67]. By using video information, C4.5 has been applied for detecting soccer goals [28].

We noticed, however, that there are a few disadvantages in the C4.5 single decision tree algorithm when used for Microarray data classification.

1. `C4.5 suffers the generalization problem`. As we saw in Section 3.2, to construct a decision tree C4.5 splits data into disjoint subsets until each subset contains only one class. As a result, each example cannot be covered by more than one rule. The decision tree generated from C4.5 is therefore relatively small. An argument for preferring a small tree is that they do not overfit the training data sets and this results in higher accuracies in test data sets. In data mining and machine learning domains, such as retail, financial, communication and marketing data, large numbers of samples are available for classification purposes. They are very easy to collect with very low cost. However, this becomes a major hurdle when it comes to Microarray classification applications. The nature of Microarray data is that it contains a large number of attributes with a very small number of samples. The number of samples is very small due to the cost of experiments and the source of samples etc. When the sample data is too small, it is more likely that the tree stops growing before it can be general enough to represent a reliable

classifier.

2. `The C4.5 method is susceptible to noise in the datasets` Furthermore, high-throughput DNA Microarray technologies generate complex biological Microarray data with a great deal of noise. It is a very dangerous situation for Microarray data classification when noisy data helped to predict data during the training stage, but failed at the testing stage, because the predictions are based on false Microarray data. False Microarray data inducts false conclusions. And the serious consequences are obvious.

   The vulnerability of C4.5 can be highlighted in the following example. The small rule sets included in the decision tree are too slim to tolerate the possible noisy values in the unseen test data. For example, Test A, Test B and Test C need to be taken before a patient is diagnosed with having diabetes. After applying a decision tree classification algorithm on past patients data, a single decision tree is generated with contains a rule

   ```
   Test A = high ==> diabetes.
   ```

   Other potential rules:

   ```
   Test B = high ==> diabetes
   ```

   and

   ```
   Test C = high ==> diabetes
   ```

are not be able to be included in the tree. Therefore patients who do not take Test A will miss the matching of the rule, and may be classified as normal by the default class even though the patients test B result is High. It is obvious that if we included other rules to cover the same patient sample, then it will not misclassify patients who do not take test A.

Like C4.5, many Microarray data classification methods which have one single decision tree suffer the same problem. In other words, a single decision tree Microarray data classification faces a great risk as it can be led astray by noisy data.

Now we seem to have a dilemma for Microarray data classification by using a single decision tree. One one hand, a single decision tree classification method is accurate and scalable with easily understandable representation of output. This is very desirable for Microarray data classification. On the other hand, most existing Microarray data would not be able to provide sufficient sample data. This situation has raised a great challenge for single decision tree classification methods in making reliable decision trees.

3.3 Ensemble decision tree classification

In this section, we focus on ensemble decision tree methods. As we discussed in Section 2.3 in Chapter 2, we refer to robustness as a good feature to tolerate noise in data, and it is associated with predictions based on data with noisy values. A Microarray classification algorithm which performs accurately and reliably even with increasing levels of noise data is considered robust. Hence, to increase the reliability of Microarray classification, we have to ensure that the classification algorithms are robust in tolerating the high level of noise. Otherwise, Microarray data classification based on Microarray data with high levels of noise will not necessarily lead to a reliable classification.

Naturally, an intuitive and simple way to achieve robustness is to take advantage of redundancy. This idea has been used in telecommunications for many years. A typical example of using redundancy in telecommunications is data transfer, in which some redundant bits are used for correction purposes in order to ensure the data transfer has minimum errors caused by missing or incorrect bits.

Instead of using one decision tree for prediction, ensemble decision tree classification methods combine the prediction of several decision trees for classification tasks [17].

Ensemble methods combine multiple classifiers or models, which are built on a set of re-sampled training data sets or generated from various classification methods on a training data set. For the sake of simplicity, this set of classifiers is referred to as a decision committee, by which future incoming samples are classified. The aggregation of the decision committee can be a simple vote or a weighted vote of individual classifiers in the committee.

Figure 3.2 shows the basic framework for an ensemble methods.

One approach to generating an ensemble committee with n decision trees is to obtain n different data sets derived from the original data set using different re-sampling methods. The re-sampling methods usually re-sample the original data sets, and each re-sampled data set is divided into training and test data sets. The selected single decision tree method then is applied

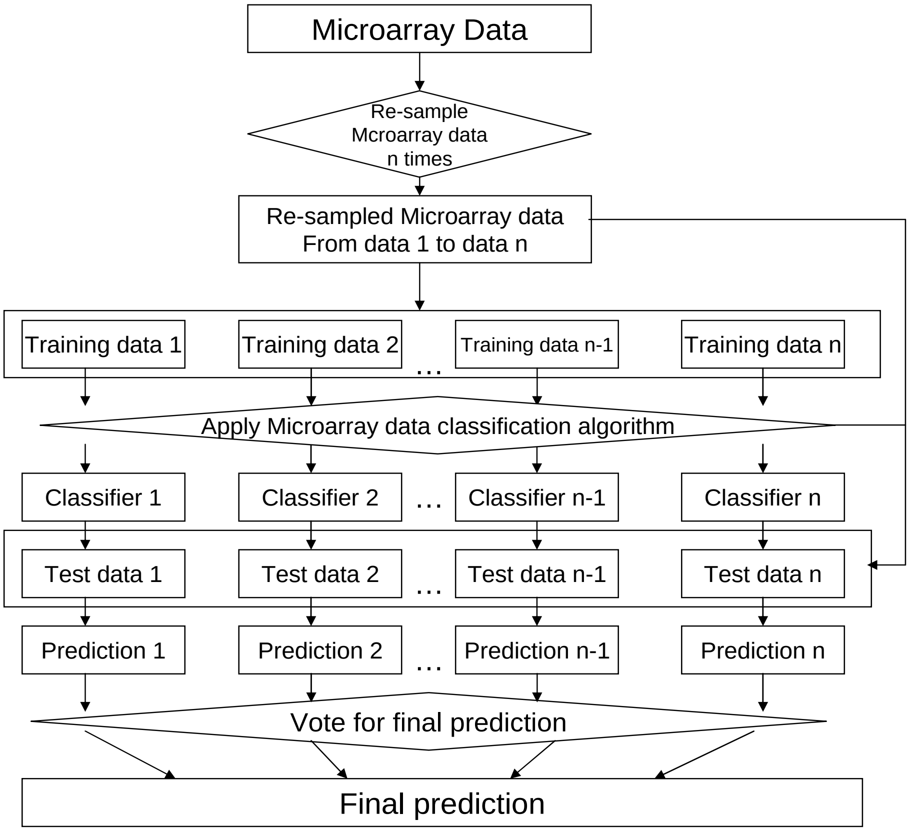


Figure 3.2: Ensemble classification flow chart

to each re-sampled training data set for n times. After the training stage, a decision tree is generated. During the test stage, each decision tree in turn is used to classify the samples in the test samples.

Ensemble methods which aggregate many diversified trees, are more effective than single decision tree methods in reducing bias that might exist in an individual tree, and consequently improve the robustness of Microarray classification. Compared with single decision tree methods, ensemble methods also show promise of achieving high classification accuracy and the classification from the ensemble tree methods are relatively easy to be interpret.

Because of those features, we will describe ensemble decision tree methods in more details in Chapter 4.

3.4 Summary of the chapter

In this chapter, we described single and ensemble decision tree methods. We discussed the understandability and robustness issues in Microarray data classification. In addition, we described the workings of C4.5 in details, which is a representative of the single decision tree methods. We also presented a brief flow chart of multiple classifiers.

In the next chapter, we introduce a few well-known ensemble decision tree methods and compare their performance.

Chapter 4

A comparison of ensemble decision tree algorithms

In past decades, many researchers have devoted their efforts to the study of combining decision trees for gene expression Microarray classification, in order to enhance the predictive accuracy and robustability of gene expression Microarray data analysis [17, 52, 18, 83].

The essence of ensemble tree methods is to generate alternative decision trees in the decision committee. In this Chapter, we describe a few ensemble decision tree methods which use different techniques for generating alternative trees. We also conduct a robustness analysis between the described methods.

This chapter is organized as follows. In Section 1, we introduce some ensemble decision tree methods by re-sampling samples. We show our experimental results and present discussions. In Section 2, we introduce some ensemble decision tree methods by re-sampling attributes. We also show our experimental results and present discussions. In Section 3, we summarize the chapter.

4.1 Ensemble decision tree methods by resampling samples

Traditional classification algorithms, such as Bagging and Boosting, re-sample samples in the training data to build multiple classifiers. In this section, we elaborate some of these ensemble decision tree methods.

Bagging and Boosting are well-known ensemble methods in the machine learning and data mining fields and have been extensively studied in gene expression Microarray data analysis.

4.1.1 Bagging

In this section, we briefly describe the Bagging algorithm. Let's assume a given relational data set D with n attributes and m samples. Let s be a sample containing a set of attribute-value pairs with a class value c which is one of C classes in D. The preset number of ensemble decision tree is denoted by K. The ensemble decision tree or classifier of T_i represents the tree generated on the i-th iteration while D_i represents the i-th data set used for generating T_i. C_i is a classifier.

The Bagging method is detailed in Algorithm 1

Bagging or Booststrap aggregation was proposed by Leo Breiman [17, 12] in 1990. Bagging(see Algorithm 6.4) uses a bootstrap technique to re-sample

Algorithm 1: The Bagging algorithm [12]

```
input  : training set $D$, Tree classifier $\mathcal{T}$, integer $K$ (number of
         ensemble trees)
for $i = 1$ to $k$ do
    $D_i$ = bootstrap sample from $D$ (sample with replacement);
    $C_i = \mathcal{T}_i(D_i)$
end
$C^*(s) = \arg max \sum_{i:C_i(s)=c} 1$(the most often predicted label $c$);
output: classifier $C^*$
```

the training data sets D. To form a re-sampled data set D_i, each sample is independently drawn from D with n samples. Each sample in D has a probability of $/1/n$ of being drawn in any trial. Note that D_i contains the same number of samples as the original data set D. However, in D_i, some samples may appear more than once, and some samples do not appear at all. T_i is built on a set of re-sampled D_i. The T_i will in turn be used to classify every sample in the testing data set. This process repeats for K times. The final prediction of a sample is determined by simple voting and each classifier has an equal weight of 1. The most often predicted label will be the final classification result.

4.1.2 Boosting

Another well-known ensemble tree method which uses re-sampling technique is the Boosting method. The Boosting method was first developed by Freund and Schapire [52] in 1996. Boosting uses a re-sampling technique which is different from Bagging. The re-sampling technique can be described simply as follows. A new training data set is generated according to its sample distribution. The first classifier is constructed from the original data set where every sample has an equal distribution ratio of 1.

In the following training data set D_i, the distribution ratios are made different among samples depending on their prediction accuracy in the previous data set D_{i-1}. If a sample has a lower prediction accuracy rate in D_{i-1}, it will be given a higher weight in D_i and therefore get a higher chance to be selected in D_i. The Boosting algorithm is described in Algorithm 2. The final prediction of a sample is determined by the weighted voting on all classifiers.

Although the Bagging method is slightly different than the Boosting algorithm, both of these ensemble decision tree methods can improve classification accuracy performance [109, 40, 27, 93, 121, 38]. Tan and Gilbert [121] used Bagging and Boosting C4.5 decision trees. For Microarray data classification, the results showed that both methods outperform C4.5 single tree on some Microarray cancer data sets. In the later 1990's, Statistik and Surich developed a new BagBoosting method [38]. Their experiments showed that BagBoosting outperforms the Boosting and Bagging methods and achieved a better accuracy result on some Microarray data sets compared with some well-known single classification algorithms such as C4.5 and Support Vector Machines(SVM)methods.

In addition to the comparison with single classification methods, many ensemble methods have been compared with each other, such as mentioned above. Note that these comparisons mainly focused on predictive accuracy. Robustness comparisons between existing ensemble methods have been so

Algorithm 2: The AdaBoost algorithm [131]

```
   input : training set $D$ of size $n$, Tree classifier $\mathcal{T}$, integer $K$ (number
           of ensemble trees)
1  $D_1 = D$ with instance weights assigned to be 1
2  for $i = 1$ to $k$ do
3      $C_i = \mathcal{T}_\rangle(D_i)$
4      $\epsilon_i = \frac{1}{n} \sum_{s_j \in D_i : C_i(s_j) \neq c_i} weight(s)$ (weighted error on the
       training set)
5      if $\epsilon_i > 1/2$ then
6          set $D_i$ to a bootstrap sample from $D$ with weight 1 for every
           instance and goto step 3
7      end
8      $\beta_i = \epsilon_i / (1 - \epsilon_i)$
9      foreach $s_j \in D_i$ do
10         if $C_i(c_j) = c_j$ then
11             weight($s_j$) = weight($s_j$) $\cdot \beta_i$
12         end
13     end
14     Normalize the weights of instance so the total weight of $D_i$ is $m$
15 end
16 $C^*(s) = \arg max \sum_{c \in C} \log \frac{1}{\beta_i}$
   output: classifier $C^*$
```

far ignored in most research literature. In the next section, we will conduct a series of robustness experiments to evaluate the robustness performance of Bagging and Boosting.

4.1.3 Comparison of robustness results with C4.5

To compare their robustness, we select a single decision tree method — C4.5 and two ensemble decision tree methods: Bagging and Boosting. All these methods are working on four Microarray data sets listed in Chapter 2. Here we emphasize that white noise will be added to the original data sets. These Microarray data sets are ALL-AML Leukemia, Colon , Lymphoma and Lung Cancer. White noise or Gaussian noise were generated based on the selected Microarray cancer data sets with 0%, 20% and 60% levels of noise.

Firstly we present Table 4.1, which shows the individual and average accuracy results of the compared methods based on four original data sets (ie. with 0% noise level) using ten fold cross-validation method. Note that the last row in Table 4.1 is the average.

	Data set	C4.5	AdaBoostC4.5	BaggingC4.5
1	Leukemia	79.2	87.5	86.1
2	Colon	82.3	77.4	82.3
3	Lymphoma	78.7	85.1	85.1
4	Lung Cancer	95.0	96.1	97.2
	Average	83.8	86.5	87.7

Table 4.1: Accuracy comparison of four original data sets of three classification methods.

Based on table 4.1, we make the following observations from comparing the performance between the ensemble decision tree methods.

Firstly, the two ensemble decision tree methods both outperform over the single decision tree C4.5 and improve the average accuracy by up to 3.9%.

These results demonstrate that ensemble decision tree methods generally can improve the accuracy on average over single decision tree methods on Microarray data sets. These results are consistent with most machine learning studies.

We also notice that interestingly, AdaBoostC4.5 decreases the accuracy performance on the Colon data set by 4.9%.

Secondly we present Table 4.2, Table 4.3 and Table 4.4 which show the details of the accuracy results with various levels of noise values for C4.5, AdaBoostC4.5, Baggingc4.5 respectively.

	Data set	C4.5 (with 0%)	20%	60%
1	Leukemia	79.2	70.4	66.7
2	Colon	82.3	72.5	47.9
3	Lymphoma	78.7	78.2	70.7
4	Lung Cancer	95.0	72.3	65.8
	Average	83.8	73.3	62.8

Table 4.2: Prediction accuracy of C4.5 with different levels of noise values

	Data set	original	20%	60%
1	Leukemia	87.5	80.0	76.7
2	Colon	77.4	82.5	64.6
3	Lymphoma	85.1	88.2	75.7
4	Lung Cancer	96.1	71.7	70.3
	Average	86.5	80.6	71.8

Table 4.3: Prediction accuracy of AdaBoostc4.5 with different levels of noise values

From the experimental results in Table 4.3 and Table 4.4, we make the following observations:

1. *Training datasets with a lower noise level of* 20%

   BaggingC4.5 performs the best with decreasing accuracy on average by 1.1%. AdaBoostC4.5 decreases the accuracy by 5.9% while single

	Data set	original	20%	60%
1	Leukemia	86.1	88.3	82.1
2	Colon	82.3	85.8	62.1
3	Lymphoma	85.1	85.7	84.1
4	Lung Cancer	97.2	86.7	81.9
	Average	87.7	86.6	77.5

Table 4.4: Prediction accuracy of Baggingc4.5 with different levels of noise values

tree method C4.5 decreases the accuracy on average by 10.5%. For self comparison to the original results, Baggingc4.5 increases its accuracy on the Leukemia data set by 2.2%,while all other algorithms decrease their accuracy by up to 8.8%; all algorithms increase their accuracy on the Colon data set by up to 5.1% except C4.5; Adaboostc4.5 and Baggingc4.5 increase their accuracy on Lymphoma despite C4.5 decreasing its performance; Both algorithms decrease this accuracy on the Lung cancer data set, while Adaboostc4.5 performs the worst among the ensemble methods with the biggest decrease of 24.4%.

2. *Training data sets with a relatively high noise level of* 60%

   All compared classification algorithms decrease the accuracy performance on average. Among the two ensemble methods, Baggingc4.5 and AdaBoostc4.5 decrease their accuracy by 10.2% and 14.7% respectively while C4.5 decreases the accuracy on average by 21%.

By comparing Table 4.2, Table 4.3 and Table 4.4, we notice that ensemble decision tree methods generally tolerate noise values better than single decision tree C4.5 does.

The reason the single decision methods have poor robustness on Microarray data with high levels of noise is that a single decision method is susceptible to noise and nothing dampens this adverse effect.

In contrast, an ensemble decision tree method, which usually contains a multiple of trees, always has a better robustness, since if one tree is affected by noise, the other trees might not be affected at all. The impact of noise is averaged down due to the ensemble classifier voting process.

However, the experimental results also revealed some limitations that Bagging and Boosting methods suffer in terms of dealing with high levels of noisy data. The limitations can be seen as follows.

1. *The Bootstrap method prevents the individual decision tree being generated from the entire training data set.*

   With the bootstrap method, only about 2/3 of the original training examples are used for constructing an individual decision tree. The accuracy of individual decision trees might not be affected if the training data set is big enough. However, as far as gene expression Microarray data analysis is concerned, gene expression Microarray data usually contains less than 100 samples. With the scarcity of the original training data set, the bootstrap method tends to decrease the accuracy of the decision tree compared to the one based on full training data set.

2. *The Bagging method does not guarantee the diversity of ensemble decision tree.*

   When re-sampled training datasets by the bootstrap process are identical or have similar class distributions, the decision trees generated from them are not so diversified. Intuitively, if individual decision trees in an ensemble committee are all identical, the ensemble committee is of

> little use in improving the prediction performance over a single decision tree algorithm. Ho [64] demonstrated that both the Bagging and the Boosting methods generated a high percentage of similar decision trees.

The Boosting method uses the entire training data set for constructing an individual decision tree, therefore the prediction accuracy of each individual tree tends to be more accurate than that of the Bagging method. However, it still has a disadvantage for gene expression Microarray data classification. It has the same risk as Bagging in terms of diversity. In addition, it is potentially not robust to noise data due to the re-sample technique used. The Boosting method assigns more weight to the samples with a higher prediction error rate. This is the case when a sample with a higher weight contains a high level of noise of genes or attributes. Since we know that Microarray data contains high levels of noise, as a result the re-sampled training dataset contains increased noise data, and the decision tree based on such a data set causes an overfit problem.

Therefore, re-sample examples are not good for Microarray data classification, and the boosting and bagging methods need to be improved with regard to diversity and robustability issues.

4.2 Ensemble decision tree methods by re-sampling attributes

The re-sample attributes technique is to vary genes instead of samples in order to maximally use the small number of samples and the large number of genes. In stead of re-sampling samples, a few new ensemble methods [142, 18, 83]have been proposed which construct decision tree ensembles by re-sampling genes since a Microarray data set normally has abundant genes, which can be used to counter the overfitting problem caused by a small sample size.

4.2.1 Random Forests

Random decision forests ensemble decision tree methods have been researched extensively [64, 63, 18, 142]. For example, Leo Breiman proposed a random decision forests method called Random Forests [18] in 1999. This early random decision trees method combines Bagging and random feature selection methods to generate multiple classifiers.

The Random Forests method is show in Algorithm 8. In Algorithm **??** first, bootstrap is adopted to form a re-sampled training data set D_i from which T_i will be constructed. During the T_i constructing stage, at each node a fixed number of features is selected randomly for splitting on. Two features are tried among the selected set of features and the one with the higher information gain ratio is selected to split the training data set.

Algorithm 3: The Random Forests [81]

```
Initially select the number of $K$ of trees to be generated; samples $s$ in
training set $D$; Tree classifier $\mathcal{T}$
for $i = 1$ to $K$ do
    $D_i$ = bootstrap sample from $D$ (sample with replacement) A
    Vector $\theta$ is generated
    (random selected genes for each node) Construct Tree $T_i = (s, \theta)$
    using any decisions tree algorithm
end
Each Tree casts 1 vote for the most popular class at $S$
$C^*(s) = \arg max \sum_{i:C_i(s)=c} 1$ (The class at $S$ is predicted by selecting
the class with max votes)
Outputclassifier $C^*$
```

Zhang and *et al.* [142] proposed a new ensemble decision tree method called deterministic forest which was a modified version of random forests. The modification is that instead of re-sampling the training data set, this method selects a specified number of the top splits of the root node and then generates a number of alternative trees, based on the selected top splits.

With another modified Random Forests, instead of using bootstrap method to re-sample the original training data set, Tin Kam Ho [64, 63] introduced a new method to generate random forests by re-sampling the features [1] from the original training data set. D_i contains all samples appearing in the original training data set with a randomly selected subset of features included in the original training data set.

4.2.2 CS4-Cascading-and-Sharing

CS4–cascading-and-sharing (simply called CS4) proposed by Jinyan Li and Huiqing Liu [83] makes use of gene selection at root nodes in their ensemble C4.5 algorithm for Microarray data classification. CS4(see Algorithm 4), first uses the information gain ratio to select the top n genes from the original data set. Then each of these n genes in turn is used as the root node of an alternative tree of ensemble trees. Root nodes of ensemble trees are not determined by C4.5, but the remaining parts of trees are constructed by C4.5. Based on the coverage of rules, the final predicted class is the class that receives the highest score. CS4 diversifies the roots of ensemble decision tree, but does not diversify all trees in the committee as does our proposed algorithm.

[1] Note that features can be different gene.

Algorithm 4: CS4 Algorithm [84]

```
Data: Given a training data set $D$, $k$ is significantly less than the
      number of features used in $D$
Use gain ratios to rank all the features into an ordered list with the
best features at the first position
$i = 1$
for $i = 1, i <= k$ do
    Use the $i$th feature as root node to construct the $i$th $rule_i$
end
Score the predicted class based on the coverages of the rules
Final predicted class is the class that receives the highest score
```

4.2.3 Comparison of robustness results

To compare the robustness of the newly developed ensemble methods, Random Forests and it variants, CS4 are selected for ensemble decision tree accuracy comparison on four data sets, namely Leukemia, Colon , Lymphoma and Lung Cancer. Ten fold cross-validation is used in this experiment. Note that for ease of comparison, we also place the accuracy results of the Boosting method and Bagging method as column 4 and 5 respectively in Table 4.5.

Like the Boosting or Bagging methods described in Section 3, white noise or Gaussian noise were generated based on the selected Microarray cancer data sets with 0%, 20% and 60% levels of noise.

Table 4.6, Table 4.7, Table 4.2, Table 4.3 and Table 4.4 show the details of the accuracy results with various level of noisy values for Random Forests, CS4, C4.5, AdaBoostC4.5, BaggingC4.5 respectively.

Table 4.5 shows the average accuracy results for the two selected algorithms: Random Forests, CS4 are selected along with BaggingC4.5 and AdaBoostingC4.5 over the four original data sets.

Table 4.6 shows the prediction accuracy of Random Forests over four data

	Data set	C4.5	Boost	Bagging	Random Forests	CS4
1	Leukemia	79.2	87.5	86.1	86.1	98.6
2	Colon	82.3	77.4	82.3	75.8	82.3
3	Lymphoma	78.7	85.1	85.1	80.9	91.5
4	Lung Cancer	95.0	96.1	97.2	98.3	98.9
	Average	83.8	86.5	87.7	85.3	92.8

Table 4.5: Average accuracy comparison

	Data set	original	20%	60%
1	Leukemia	86.1	80.8	83.3
2	Colon	75.8	80.0	76.3
3	Lymphoma	80.9	80.7	76.4
4	Lung Cancer	98.3	96.1	92.3
	Average	85.3	84.4	82.0

Table 4.6: Prediction accuracy of Random Forests

sets with different levels of noise values.

	Data set	original	20%	60%
1	Leukemia	98.6	95.0	89.6
2	Colon	82.3	81.3	70.4
3	Lymphoma	91.5	95.0	91.6
4	Lung Cancer	98.9	97.8	98.9
	Average	92.8	92.3	87.6

Table 4.7: Prediction accuracy of CS4 over four data sets with different levels of noise values

Table 4.7 shows the prediction accuracy of CS4 over four data sets with different levels of noise values.

From the experimental results presented in Table 4.5, Table 4.6 and Table 4.7, we have the following observations:

1. Based on the original data sets, compared to the single decision tree CS4 is the best ensemble method and outperforms C4.5 by 9.0%. Random Forests, AdaboostC4.5 and BaggingC4.5 also improve the accuracy on average by up to 3.9%. Among the four ensemble methods, CS4

is the most accurate classification method and improves the accuracy of classification on all cancer data sets by up to 19.4%. BaggingC4.5 also outperforms C4.5 on all data sets by up to 6.9%. Random Forests and AdaBoostC4.5 improve the accuracy on lung cancer, Lymphoma and Leukemia data sets by up to 8.3%, but fail to improve the accuracy on the Colon data set.

2. With a lower noise level of 20%, CS4 and Random forests perform the best with a slight change over the original data by up to 0.9% on average. BaggingC4.5 performs well with decreasing accuracy on average by 1.1%. AdaBoostC4.5 decreases the accuracy by 5.9% while single tree method C4.5 decreases the accuracy on average by 10.5%.

3. With a high noise level of 60%, Random Forests perform the best with a smallest decrease of 3.3% compared with the other ensemble methods. CS4 decreases the accuracy by 5.2% while BaggingC4.5, AdaBoostingC4.5 and C4.5 decrease their accuracy by 10.2%, 14.7% and 21%, respectively.

4.3 Discussions

The robustness of Microarray classification is affected by data re-sampling methods. A traditional ensemble method assumes that a training data set has a large number of samples with small numbers of attributes. The re-sampled data set is only slightly different from the original data set. The trees constructed on those re-sampled data are still reliable. However, most Microarray data contains less than 200 samples, a slight change of samples may cause a dramatic structural change in the training data set. The trees constructed on such unreliable data sets are more likely to lead to a higher risk of the unreliability problem. This risk affects the performance of classification. In contrast, CS4 is designed specially for Microarray data analysis. It kept the structure of the original data sets and therefore the generated trees are more reliable and the ensemble committee is more robust.

4.4 Summary of the chapter

In this chapter, we conducted a comparative study of a few newly developed ensemble classification methods by re-sampling samples and attributes. In particular, we introduced Bagging, Boosting, Random Forests and CS4. We compared these methods on four Microarray data sets with various level of noise. We discussed the the comparison results. In the next chapter, we introduce our newly developed ensemble decision decision trees method.

Chapter 5

DMDT—A novel robust Microarray classification method

From the observations of previous chapters, we noticed that there are a number of factors which affect the accuracy and robustness of Microarray classification. As we discussed in Chapter 1, Microarray data commonly contains a high level of noise. High noise in Microarray data is not helpful in improving the accuracy of Microarray data classification. On the other hand, a Microarray data set normally has a small number of samples, and the re-sampling samples method does not improve the accuracy performance of classification significantly on Microarray data. In the worst case, it decreases the performance of Microarray data classification especially with increased noise levels.

In this chapter, we propose a new Microarray data classification method, based on diversified multiple trees. The new method contains features that,

(1) make most use of the information from the abundant genes in the Microarray data, and (2) maximize diversity in the ensemble decision committee. This chapter is organized as follows. In Section 1, we discuss the measurement of diversity. In Section 2, we apply the concept of diversity measurement to evaluate the diversity among compared ensemble decision tree methods. In Section 3, we propose a new Microarray data classification method called d̲iversified multiple decision trees algorithm (DMDT). In Section 4, we show experimental results. In Section 5, we present a discussions on the DMDT algorithm. This chapter is concluded by a short summary in Section 6.

5.1 Measurement of diversity

Apart from the generally high noise level in Microarray data, the level of noise also varies from one data set to another. This data uncertainty is another characteristic of Microarray data. To cope with this data uncertainty, ensemble methods aim to improve their predictive power by combining diversified or alternative trees in order to increase the robustness of Microarray classification. To achieve this goal, many methods have been applied to diversify the ensemble committee, such as re-sampling samples using bootstrapping or re-sampling attributes with gene selection.

Ensemble decision tree classification methods all generate a set of decision trees to form a committee. Due to the different approaches used to generate the committee, the decision trees in the final ensemble committee could be diverse from each other in certain ways. In the past decades, measuring diversity has become a very important issue in the research of Microarray ensemble classification methods [77, 5, 131].

Measuring outputs is a most natural way to measure the diversity of ensemble classifiers [5]. The output from measuring the classifiers in a committee may give a result of total different, partially different or identical with each other. If the classifiers in a committee are all identical in a committee, we can say these classifiers are not diversified; if the classifiers are partially different, we can say they are diversified. When the classifiers are totally different or unique to each other, we say the classifiers are maximally diversified.

There are also many statistical diversity measures available, such as diversity of errors [101, 5], and pairwise and non-pairwise diversity measures [77, 55, 5]. It is desirable if every classifier in an ensemble committee can agree on most samples which are predicted correctly. At the same time, we also expect that they do not make same incorrect predictions on testing samples. Those methods are also very important measurements of diversity, because if their errors were correlated, classification prediction would not lead to any performance gain by combining them.

The approach of measuring diversity based on statistical methods has drawbacks. There is a lack of robustness consideration in Microarray classification in terms of incorrect and missing data values. Identical trees are excluded from the ensemble committee since they are not helpful in improving the prediction accuracy of classification. However, this measurement allows overlapping genes among diversified trees. Overlapping genes are a problem for reliable Microarray data classification.

Let's use an example to illustrate the problem. For simplicity purposes, we assume that each ensemble decision tree generated contains only one rule. a,b,c,d,e,f,g,h represent attributes, while z represents a class. In our example, we have an ensemble committee containing three trees.

- tree1: $g\ and\ h \Rightarrow z$
- tree2: $a\ and\ b$ and $c \Rightarrow z$
- tree3: $b\ and\ c \Rightarrow z$

b had values in training data but is missing in the unseen test data. The problem occurs when the ensemble committee applies to the unseen test data since the missing value b paralyze both tree2 and tree3. In this case having redundant rules does not help the ensemble classifier to outperform the accuracy performance of a single classification regardless how diversified they are. This is not good for overall robustness. In contrast, this problem can be easily dealt with by measuring the outputs using ensemble classification methods. We simply require tree1, tree2 and tree3 to be disjunctive by containing totally different genes:

- tree1 $g\ and\ h \Rightarrow z$,
- tree2 $a\ and\ b\ and\ c \Rightarrow z$ and
- tree3 $d\ and\ e\ and\ f \Rightarrow z$.

In this case, if one gene of a, b, c, d,e, f, g or h is missing, only one rule is paralyzed. The other rules still work.

In addition, measuring outputs takes advantage of the abundant genes contained in Microarray data. This character of Microarray data ensures that completed different outputs can be generated from any give Microarray data without any difficulties.

In summary, Microarray data suffers from the curse of dimensionality and a high degree of data uncertainty. Ensemble decision tree formed by way of measuring the outputs increases the possibility of tolerating incoming unknown data, which involves a higher degree of noise than the data used for training ensemble decision tree. That is to say, a general Microarray data analysis technique requires the disjointed outputs to be generated and be able to take full advantage of abundant genes.

To the best of our knowledge, so far there is still not a generally accepted formal definition of measurement of diversity [77]. There is no such general diversity measurement which is best for improving the classification performance of ensemble methods in all applications cross different research fields.

In this thesis, diversity is measured by the difference of outputs for Microarray data classification problems. The degree of diversity is dependent on how many overlapping genes are included between the decision trees of an ensemble committee.

Definition 3 (Degree of diversity)**.** *Given a data set D with n attributes, $A = \{att_1, \cdots, att_n\}$; C is an ensemble decision tree committee with $k(k > 1)$ individual decision trees generated from D, $C = \{c_1, \cdots, c_k\}$; $c_i \in C$ and $c_j \in C$ are any single decision trees; c_i contains a set of attributes $A_{c_i} =$*

$\{att|att \in A\}$ *and* $A_{c_j} = \{att|att \in A\}$*;*

let $|A_{c_i} \cap A_{c_j}|$ $=$ *number of element contained in* $A_{c_i} \cap A_{c_j}$*,*

$|A_{c_i} \cup A_{c_j}|$ $=$ *number of element contained in* $A_{c_i} \cup A_{c_j}$

degree of diversity between c_i *and* c_j *is*

$$DD = 1 - \frac{|A_{c_i} \cap A_{c_j}|}{|A_{c_i} \cup A_{c_j}|} (\ 0 \leq DD \leq 1)$$

When an ensemble committee contains only decision trees which have totally different outputs, or unique trees with no overlapping genes, we say that the ensemble committee is maximally diversified. According to Definition 3, the DD of the unique decision trees is 1.

Definition 4 (Unique decision trees)**.** c_i *and* c_j *are called unique decision trees, if* $A_{c_i} \cap A_{c_j} = \phi$*.*

We say an ensemble decision tree classification method has greater diversity when its decision trees have a higher degree of different outputs with less overlapping genes. It is clear that diversified decision trees have a DD value awhich is between 0 and 1.

Definition 5 (Diversified decision trees)**.** *if* $A_{c_i} \neq A_{c_j}$ *and* $A_{c_i} \cap A_{c_j} \neq \phi$*, then* c_i *and* c_j *are called diversified decision trees.*

Similarly, if all decision trees in an ensemble decision tree committee are identical, the degree of its diversity would be 0.

Definition 6 (Identical decision trees)**.** *We call* c_i *and* c_j *are identical decision trees,if* $A_{c_i} = A_{c_j}$*.*

5.2 Diversity of current ensemble decision tree methods

In this section, we look into diversity measurement in some specific ensemble decision tree methods. Up to date, all ensemble decision trees methods have kept diversity in mind. However, among those methods, Boosting and Bagging do not guarantee that each ensemble decision tree in the committee is different from outputs, namely identical trees and overlapping genes are not prohibited from an ensemble committee. Identical trees decrease the diversity of an ensemble committee, and noise in one gene may affect a number of ensemble decision tree; the noise will ultimately affect the reliability of Microarray classification. Therefore, committees built on Bagging and Boosting methods may not be as effective as a committee that contains no identical trees and overlapping genes.

A quick fix to improve diversity in the ensemble decision tree committee is to include a set of diversified decision trees with no overlapping genes. If classifiers in the ensemble decision tree committee are not guaranteed to be different to each other, the committee must be very large, in order to create certain diversity in the committee. This behoves us to pay special attention while designing our algorithm. One concern for such a split is that it might break down some attribute combinations that are good for classification. However, an apparent benefit of such trees is that a noise attribute cannot affect more than one tree in the committee. Considering that Microarray data normally contains much noise and many missing values, the idea of using diversified trees with no overlapping genes may provide a better solution.

CS4–cascading-and-sharing trees [83] is a diversified decision trees ensemble. CS4 selects the n top genes and then builds n trees from the roots of these n top genes. Apart from the root of the tree being fixed, other levels of trees are constructed by using a normal tree construction method. CS4 has been show to achieve a higher classification accuracy than Bagging and Boosting. It has also been reported that CS4 is better than other ensemble decision tree methods for Microarray data analysis. However, CS4 has some limitations. For instance, apart from the top level genes, other genes in the decision trees are shared. And a number of trees may use some genes repeatedly. Consequently, noise from one gene may affect most trees. Also, the performance of CS4 largely relies on the selection of top genes.

Generally, CS4 is better than other ensemble methods in term of diversity. Despite using traditional and most newly developed tree construction methods, the constructed alternative trees may share the same genes at either root nodes or branch nodes. To increase the diversity of all constructed trees, CS4 tries to guarantee no overlapping genes among the root nodes. Unfortunately, CS4 is unable to prevent overlapping genes from the second level of a tree and onwards.

If we want to further increase the diversity of the alternative trees, we need to make sure that all alternative trees are truly unique. That is to say, if we want to ensure truly unique trees are constructed from the ensemble decision tree method, we need to guarantee that no overlapping genes exist among the constructed trees.

A distinction between CS4 and our proposed algorithm is that there are no common genes in our trees in the decision committee whereas genes in trees of CS4 are overlapping except the root genes. We will compare these two diversified decision tree approaches in this chapter, and compare them with other traditional ensemble methods.

5.3 Diversified multiple decision trees algorithm

On the preceding section, we investigated the limitations existing in ensemble decision tree methods. In this section, we design a new diversified multiple decision trees algorithm (DMDT), which is capable of dealing with the problem of small samples versus high dimensions in Microarray data. The objective of DMDT is to improve the accuracy and reliability of ensemble decision tree methods. Our DMDT algorithm is presented in Algorithm 5. The DMDT is designed based on the C4.5 single decision tree method. We design a new Microarray data re-sampling method with the concept of robustness in mind. The features of DMDT include guaranteeing that constructed trees are truly unique, and maximizing the diversity of the final classifiers. To achieve this, DMDT reduces the instability caused by overlapping genes in current ensemble methods. For example, if the expression level of one gene is read wrongly, it only affects one tree and all other trees are unaffected.

DMDT algorithm consists of the following two steps:

1. Tree construction

   The main idea is to construct multiple decision trees by re-sampling genes. All trees are built on all of the samples but with different sets of genes. We conduct re-sampling data in a systematic way. First, all samples with all genes are used to build the first decision tree. The decision tree is built using the c4.5 algorithm. After the decision tree is built, the used genes appearing in the decision tree are removed from the data. All samples with the remaining genes are used to built the second decision tree. Then the used genes are removed and so on. This process repeats until the number of trees reaches a preset number. As a result, all trees are unique and do not share common genes.

2. Classification

Since the k-th tree has only used the genes that have not been selected by the previously created $k-1$ trees, the quality of the k-th tree might be decreased. To fix this problem, we take a vote approach; that is to say, the final predicted class of an unseen sample is determined by the weighted votes from all constructed trees. Each tree is given the weight of its training classification accuracy rate. When the vote is a tie, the class predicted by the first tree is preferred. Since all trees are built on the original data set, all trees are accountable on all samples. This avoids the unreliability of voting caused by sampling a small data set. Since all trees make use of different sets of genes, trees are independent. This adds another merit to this diversified committee. One gene containing noise or missing values affects only one tree, and not multiple trees. Therefore, it is expected to be more reliable in Microarray data classification where noise and missing values prevail.

Algorithm 5: Diversified multiple decision trees algorithm (DMDT)

```
1. TREECONSTRUCTION($D, \mathcal{T}, n$)
   INPUT: A Microarray data set $D$, and the number of trees $n$.
   OUTPUT: A set of disjointed trees $\mathcal{T}$
   let $\mathcal{T} = \emptyset$
   for $i = 0$ to $n - 1$ do
      call c4.5 to build tree $T_i$ on $D$;
      remove genes used in $T_i$ from $D$;
      $\mathcal{T} = \mathcal{T} \cup T_i$.
   end for
   Output $\mathcal{T}$;
2. CLASSIFICATION($\mathcal{T}, x, n$)
   INPUT: A set of trained trees $\mathcal{T}$, a test sample $x$, and the number of trees $n$.
   OUTPUT: A class label of $x$
   let vote($i$) = 0 where $i = 1$ to $c$ = the number of classes.
   for $j = 1$ to $n$ do
      let $c$ be the class outputted by $T_j$;
      vote($c$) = vote($c$) * accuracy($T_j$);
   end for
   Output $c$ that maximizes vote($c$);
```

We give some explanation of the algorithms in the following.

As we know, C4.5 is itself a gene selection algorithm based on information gain ratio. Within DMDT no gene selection algorithm is required. In addition, C4.5 discretizes continuous values by an information gain ratio. No discretization pre-process is required for this algorithm.

In DMDT, The required input is a Microarray data set and a preset number of trees. Within the tree construction stage, the first tree (T_1) is constructed based on the original training data set. The second tree (T_2) is based on a re-sampled training data set where genes used in T_1 are removed. As a result, T_1 and T_2 share no common genes and hence are unique. The process repeats until the required number of trees k is generated.

5.4 Experimental methods

The detailed experimental methods have been introduced in Chapter 2. To evaluate the performance of the ensemble decision tree methods, five data sets, namely Breast Cancer, Lung Cancer, Lymphoma, Leukemia and Colon, are selected for the experiment. Table 2.1 shows the summary of the characters of the five data sets. We conduct our experiments by using tenfold cross-validation on the merged original training and test data sets.

Our developed DMDT algorithm is compared with five well known single and ensemble decision tree algorithms, namely C4.5, Random Forests, AdaBoostC4.5, Baggingc4.5 and CS4. Our experiments with all four algorithms apart from CS4, were done using the Weka-3-5-2 package. The experiments with CS4 were done using the software tool provided by Dr Jinyan Li and Huiqing Liu as mentioned in Section 2.4 in Chapter 2. In our examples, the default settings are used for all compared ensemble methods. We were aware that the accuracy of some methods on some data sets can be improved when the parameters are tuned. However, it was difficult to find another uniform setting good for all data sets. Therefore, we did not change the default settings since the default produced high accuracy on average. We set the number of trees at 25 for the tenfold cross-validation test since further increasing the number of ensemble trees does not help to improve the average prediction accuracy of classification significantly for most of the Microarray data sets we used.

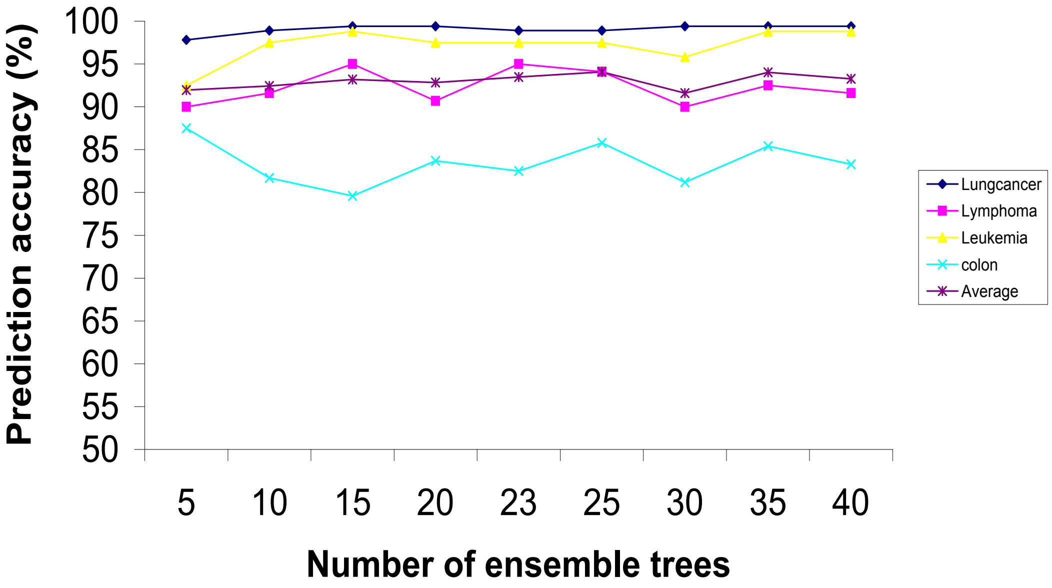


Figure 5.1: Prediction accuracy vs number of ensemble trees using DMDT

Figure 5.1 shows the individual and average accuracy results of the DMDT algorithm with different numbers of decision trees based on Leukemia, Colon, Lymphoma and Lung cancer data sets.

5.5 Experimental results

In this section, we first present the accuracy of individual methods and the average prediction accuracy of the six methods, which are all based on the ten-fold validation technique.

Table 5.1 shows the individual and average accuracy results of the six methods based on tenfold cross-validation method.

Data set	C4.5	Random Forests	AdaBoostC4.5	BaggingC4.5	CS4	DMT
Breast Cancer	62.9	61.9	61.9	66.0	68.0	64.3
Lung Cancer	95.0	98.3	96.1	97.2	98.9	98.9
Lymphoma	78.7	80.9	85.1	85.1	91.5	94.1
Leukemia	79.2	86.1	87.5	86.1	98.6	97.5
Colon	82.3	75.8	77.4	82.3	82.3	85.8
Average	79.62	80.6	81.6	83.3	87.9	88.1

Table 5.1: Average accuracy of five data sets with six classification algorithms based on tenfold cross-validation

Figure 5.2 shows the average prediction accuracy of the six methods based on tenfold cross-validation methods. The individual accuracy results are shown in Figure 5.3, Figure 5.4, Figure 5.5, Figure 5.6 and Figure 5.7.

Figure 5.3 depicts the accuracy of six classification algorithms performing on the Breast Cancer data set.

Figure 5.4 depicts the accuracy of six classification algorithms performing on the Lung Cancer data set.

Figure 5.5 depicts the accuracy of six classification algorithms performing on the Lymphoma Cancer dataset.

Figure 5.6 depicts the accuracy of six classification algorithms performing on the Leukemia dataset.

Figure 5.7 depicts the accuracy of six classification algorithms performing

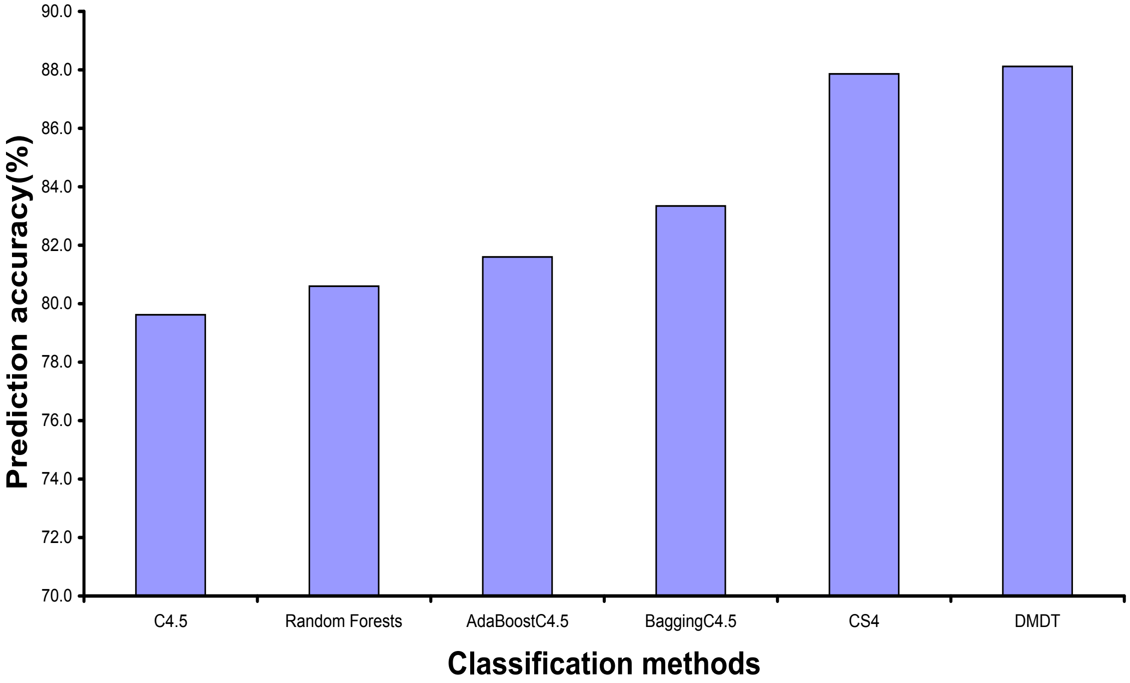


Figure 5.2: Average accuracy of six classification algorithms

on the Colon dataset.

Based on the tenfold cross-validation test, our DMDT outperforms other ensemble methods. For instance, compared to the single decision tree, DMDT is a more favorable ensemble method and outperforms C4.5 by 10.0% on average.

From Fig 5.2, we notice that CS4 also performs very well and improves the accuracy by 8.4% on average. Random Forests, Adaboostc4.5 and BaggingC4.5 improves the accuracy on average by up to 4.3%.

More specifically,

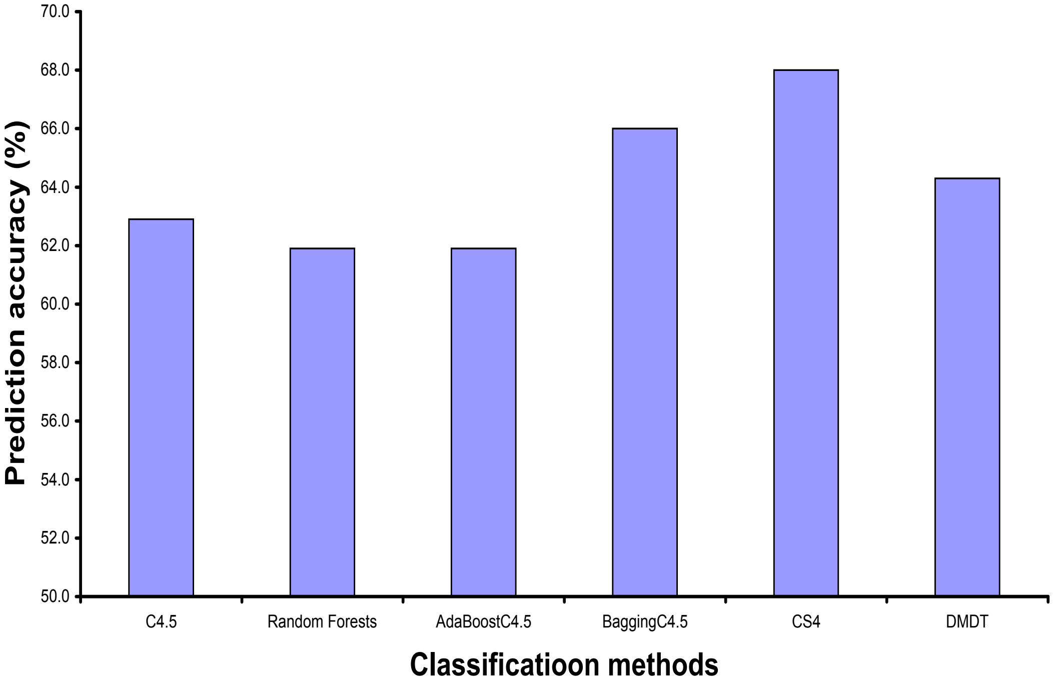


Figure 5.3: Accuracy of Breast cancer data set with six classification algorithms

1. Among the five ensemble methods used in our experiments, DMDT turns to be the most favorable classification algorithm with the highest accuracy, which improves the accuracy of classification on all cancer data sets by up to 26.7% as shown in Figure 5.3.

2. As shown in Figure 5.2, CS4 is comparable to DMDT in the test which improves the accuracy of classification on all data sets by up to 17.4%.

3. Baggingc4.5 also outperforms C4.5 on all data sets by up to 9.6%.

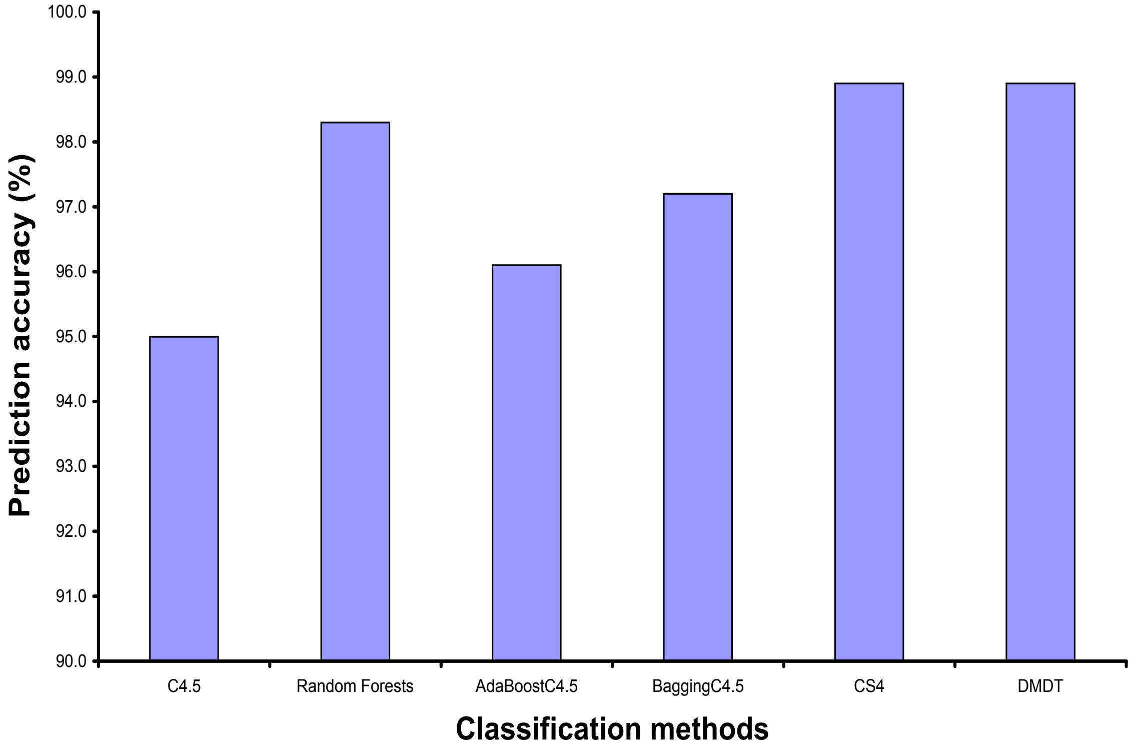


Figure 5.4: Accuracy of Lung cancer data set with six classification algorithms

4. Random Forests improves the accuracy on lung cancer, Lymphoma, Leukemia and Prostate data sets by up to 19.1%, but fails to improve the accuracy on breast cancer shown in Figure 5.3, Colon and Ovarian data sets in Figure 5.7. AdaBoostc4.5 can only improve the accuracy on Lung Cancer,Lymphoma and Leukemia and decreases the accuracy performance on the Breast Cancer and Colon data sets shown in Figure 5.7.

It is interesting to see that traditional ensemble decision tree algorithms

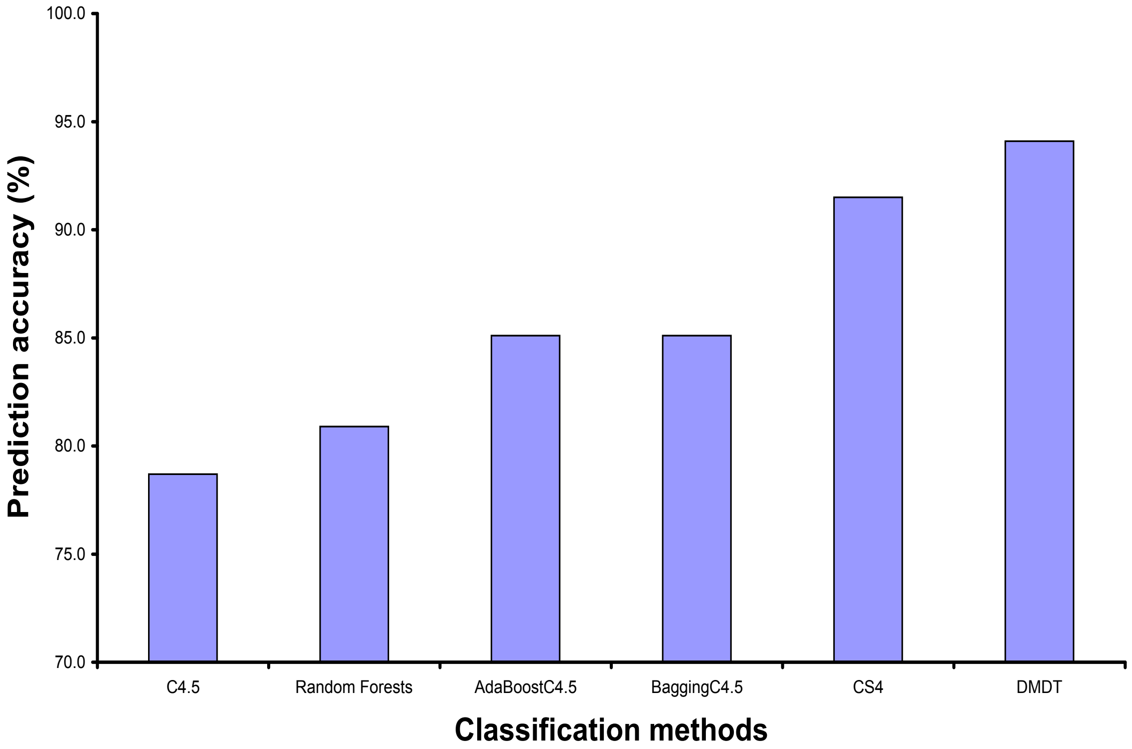


Figure 5.5: Accuracy of Lymphoma data set with six classification algorithms

do not always outperform a single tree algorithm. This is because the traditional ensemble methods assume that a training data set has a large number of samples with small numbers of attributes. As a result, the re-sampled data set is only slightly different from the original data set. The trees constructed on those re-sampled data are still reliable. However, in Microarray data analysis, the problem that we are facing is completely the opposite: a small number of samples with large numbers of attributes (genes). As most Microarray data contains less than 200 samples, a slight change of samples may cause a dramatic structural change in the training data set. The trees

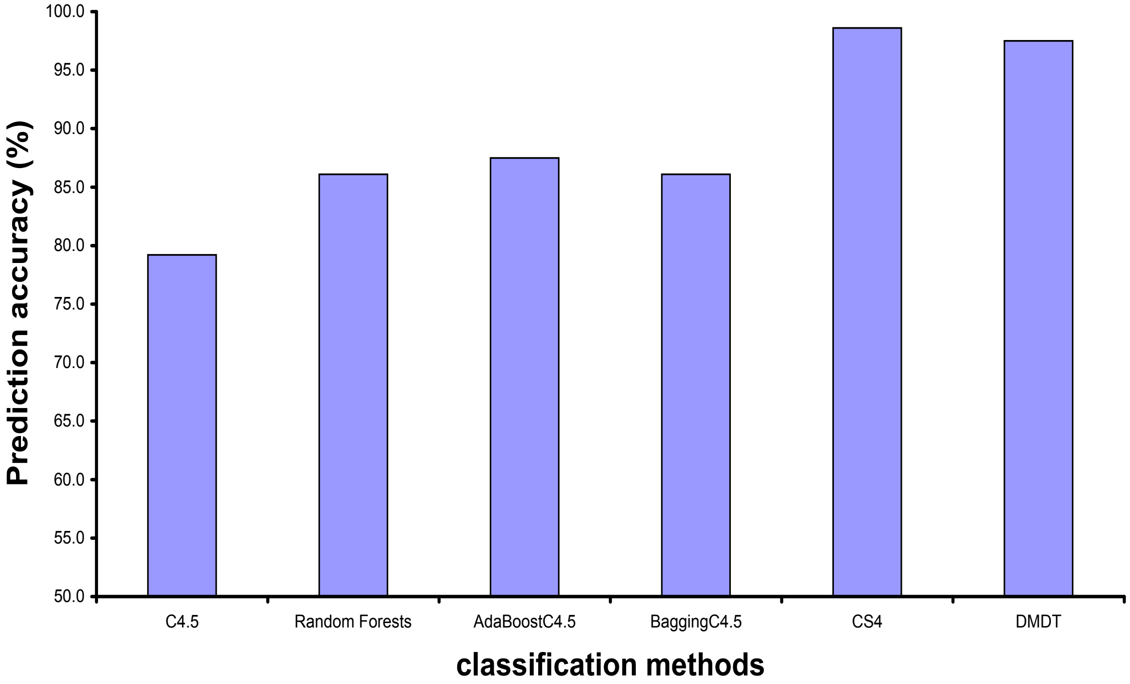


Figure 5.6: Accuracy of Leukemia data set with six classification algorithms

constructed on such unreliable data sets are more likely to lead to higher risk of the problem of unreliability. This risk affects the performance of classification. In contrast, DMDT and CS4 are designed specially for Microarray data analysis. DMDT keeps the alternative trees using all available samples in order to minimize the impact of the unreliability problem.

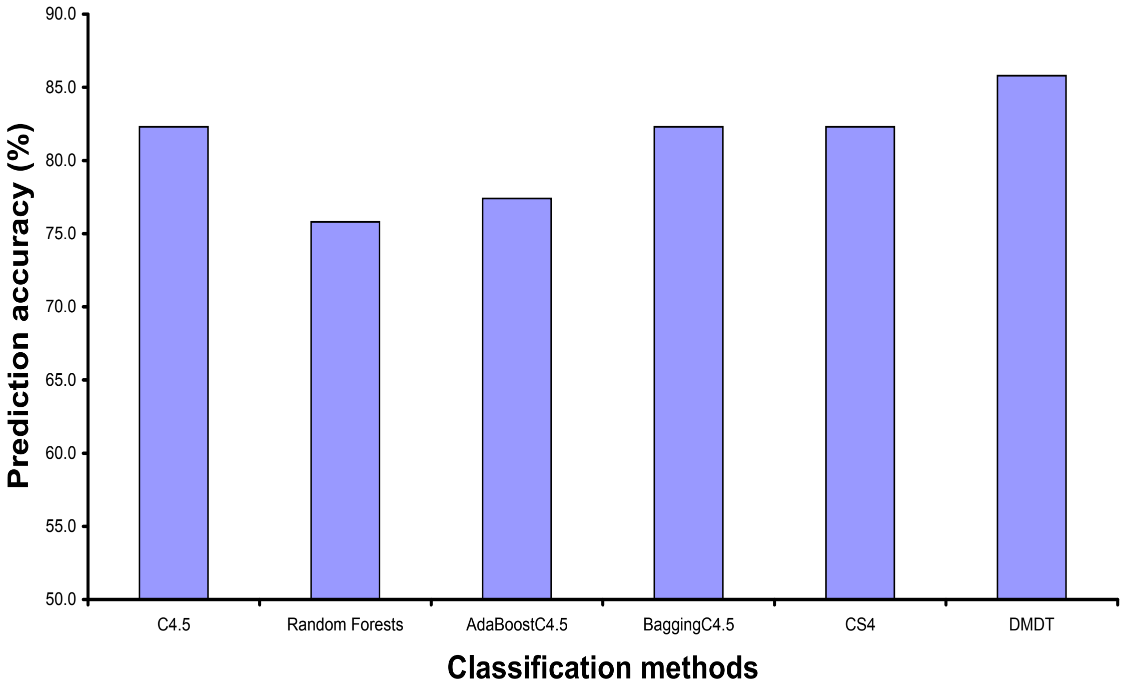


Figure 5.7: Accuracy of Colon data set with six classification algorithms

5.6 Discussions

The results of our experiments show that our proposed diversified multiple decision trees method outperforms the majority of traditional ensemble methods. This reveals that diversity improves the classification accuracy of ensemble classification. It also implies that the robustness of Microarray classification relies on the diversity of ensemble decision tree. Hereinafter, to increase the performance of ensemble classification, the ensemble decision tree algorithms must be able to generate a number of individual trees that are distinguished from (diverse to) each other [40]. For example, DMDT and

CS4 are designed to guarantee diversified trees in an ensemble committee. The results proved that DMDT and CS4 performs reasonably well in dealing with noise data. In contrast, all other ensemble decision trees method do not guarantee that all trees in an ensemble committee are different from each other.

Although the performance of CS4 is compatible to DMDT, CS4 has some weakneses which DMDT does not necessarily have.

CS4 includes a set of decision trees in the decision committee, and the decision trees have a set of distinct top genes at roots. The top genes are identified using an information gain ratio in the current CS4 algorithm. Apparently, other criteria can be used to find top genes too. If top genes are biologically meaningful, this algorithm is very useful for biologists. It groups genes by some informative genes and builds a classifier based on meaningful gene groups. However, if the top genes are mis-identified due to noise, the classifier committee will mislead the prediction results. In addition, apart from the top genes, other genes in decision trees may overlap. One noise gene may affect a number of trees. We refer to this phenomenon as ”propagation”.

That is not a problem in the DMDT algorithm. A noise gene affects only one tree, which would not be propagated to other decision trees in the decision committee. Hence DMDT should be more tolerant to noise than CS4. One concern associated with DMDT is that the enforcement of unique trees could break up some gene combinations that are good for classification. However, the experimental results shown in Section 5.5 do not indicate that this is an issue. DMDT probably affects finding some combinations of highly informative genes with less informative genes. Nevertheless DMDT is capable of finding some combinations of less informative genes that are missed by CS4. Therefore, this is a plus. Keep in mind that many biologists believe that many "uninformative genes" play an important role in diseases. DMDT has the advantage of finding such gene combinations which are missed by CS4.

In short, CS4 is capable of finding informative genes and the combinations of informative genes with informative genes, and combinations of informative genes with less informative genes, whilst DMDT is capable of discovering combinations of informative genes with their informative genes, and combinations of less informative genes with other less informative genes. In addition, DMDT has the potential of being less insensitive to noise data than CS4. Note that informative or less informative genes may only make sense to data analyzers. For biologists, the two methods use different gene sets and different combinations to equally explain a Microarray data. Both diversified ensemble methods have the potential of offering biologists some interesting discoveries.

5.7 Summary of the chapter

In this chapter, we studied the concept of diversity measurement in ensemble classifiers. We then proposed an algorithm that maximally diversifies trees in the ensemble decision tree committee. Decision trees in the committee should share no common genes. We conducted experiments on six Microarray cancer data sets. The experimental results show that DMDT and CS4, which diversifies trees by using distinct tree roots, are more accurate on average than other well-known ensemble methods, including Bagging, Boosting and Random Forests. The experiments indicate that the diversity in decision trees improves the classification accuracy of ensemble classification on Microarray data. Finally we discussed the relative strengths and weaknesses of both diversified ensemble classification methods.

Chapter 6

Robustness analysis of DMDT and other ensemble methods

DMDT has shown promise for achieving higher classification accuracy for Microarray data classification analysis based on existing Microarray data. However, a robust Microarray data classification algorithm should be able to handle the noise data well and produce reliable results from lower quality Microarray data sets. Robustness is therefore another very important criteria - in addition to accuracy - for evaluating reliable Microarray classification algorithms. Robustness is associated with predictions on data with noise. The noise can be missing data, redundant data or errors. The objective of robustness analysis is to analyze how well a given algorithm can resist noise values, specially with increased levels of noise data. Microarray data sets used for experiments may contain various amount of noise data. It is impossible for us to know how accurate the data is, how badly the data has been affected by noise, and most importantly at what level the data contains noise.

This chapter is organized as follows. In Section 1, we introduce the causes of noise data and its side effects on Microarray data classiciation. In section 2, we test the robustness of selected algorithms. The results are summarized into figures and tables. In section 3, we discuss the results. In section 4, we conclude the paper.

6.1 Characteristics of Microarray gene data

High-throughput DNA Microarray technologies generate complex biological Microarray data with a great deal of noise. DNA Microarray production involves many processes [100], such as sample preparation, spotting samples on a chip, hybridization, results collection, and data transformation [10, 100, 43] etc. Unfortunately every process can potentially bring in errors or noise due to the quality of DNA samples, such as during experimental set up, different treatment of chips during hybridization, and finally the quality of reading equipment and statistical methods [4, 50]. Although much research has been conducted to eliminate gene expression level errors through control of image processing [105] and normalization for Microarray data [126], noise data is still a key issue for Microarray data.

As a result, Microarray data classification faces a great risk as it can be led astray by these noise data. With overwhelming gene information waiting to be discovered behind the data, discovering truthful information effectively from the available abundant yet imperfect data is a great challenge for Microarray data classification.

Unlike missing data which can be seen, incorrect data hides in the existing database. It is very hard to know how much noise data is included in a database. In particular, it is dificult to detect which genes are affected. To avoid the effect of noise data, we need to ensure that Microarray data classification algorithms are robust enough to tolerate noise data contained in the Microarray data.

Robustness refers to the toleration of noise data and it is associated with predictions on data with noise values. A robust Microarray classification algorithm should perform accurately and reliably even with increasing levels of noise data. Hence, to increase the reliability of Microarray classification, we have to ensure that the algorithms we apply are robust for tolerating high levels of noise. Otherwise, Microarray data classification based on Microarray data with high levels of noise will lead to unreliable and low accuracy analysis.

Robustness therefore is one of the most important criteria for judging Microarray classification algorithms due to the nature of Microarray data. Robust ensemble classification algorithms improve the accuracy performance of Microarray data classification.

In this chapter, we focus on the robustness comparison between existing single and ensemble decision tree methods including our newly developed diversified multiple decision tree algorithm (DMDT). We test and evaluate how well a given single or ensemble decision tree classification method can tolerate noise values, particularly with increasing levels of noise.

6.2 Experimental results

In our experiments, five data sets are selected, namely the cancer, ALL-AML Leukemia [123], Colon [49], Lymphoma [6] and Lung Cancer [57]data sets. The same algorithms selected for ensemble decision tree accuracy comparison in Chapter 3 are used for robustness comparison. They are: C4.5, Random Forests, AdaBoostC4.5, Baggingc4.5 and CS4. The Polar form of the Box-Muller transformation method [39] has been used to generate additional White Gaussian noise. Please refer to Chapter 2 for details.

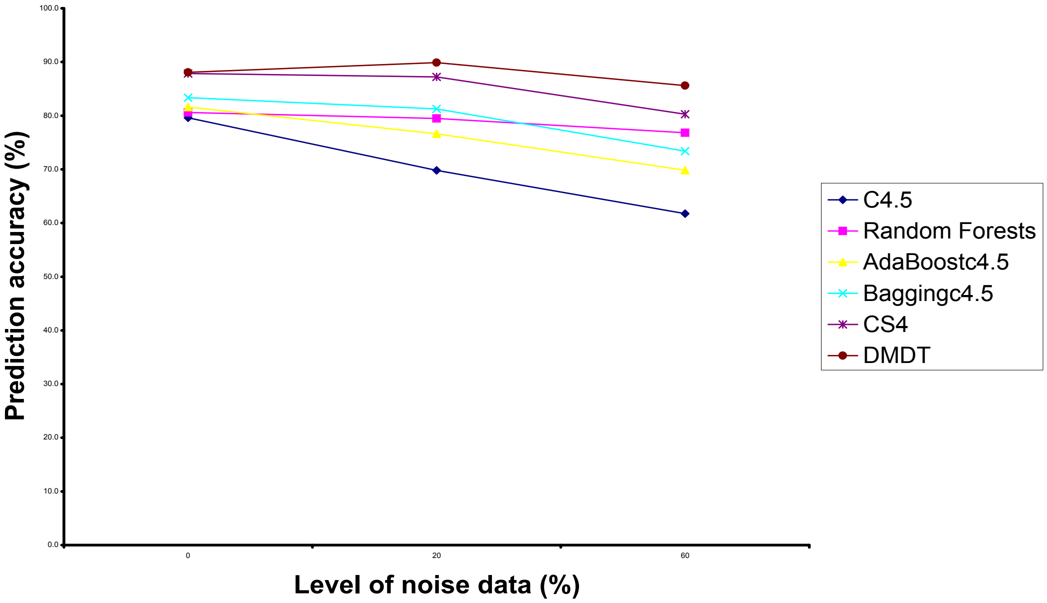


Figure 6.1: Average prediction accuracy over five data sets

Figure 6.1 shows the average accuracy results for the six selected algorithms over the four data sets with noise level of 0%, 20% and 60%. Table 6.1, Table **??**, Table 6.3, Table 6.4, Table 6.5 and Table 6.6 show the details of the accuracy results for C4.5, Random Forests, AdaBoostC4.5, Baggingc4.5, CS4 and DMDT respectively.

Data set	original	20%	60%
Breast Cancer	62.9	55.7	57.7
Leukemia	79.2	70.4	66.7
Colon	82.3	72.5	47.9
Lymphoma	78.7	78.2	70.7
Lung Cancer	95.0	72.3	65.8
Average	79.6	69.8	61.8

Table 6.1: Prediction accuracy of C4.5 over five data sets with different level of noise values

From the experimental results, we make the following observations:

1. Based on the original data sets with 0% level of noise, compared to the single decision tree DMDT and CS4 are the best ensemble methods and outperform C4.5 by up to 10.2% on average. Random Forests, Adaboostc4.5 and BaggingC4.5 improve the accuracy on average by up to 3.9%. Among the five ensemble methods, DMDT and CS4 are the most accurate classification algorithms and improve the accuracy of classification on all cancer data sets by up to 19.4%. CS4 is comparable to DMDT in the test. DMDT performs better than CS4 on Colon and Lymphoma by up to 3.2% while CS4 outperforms DMDT on Leukemia by 1.1%. And DMDT and CS4 perform equally well on the Lung cancer

Data set	original	20%	60%
Breast Cancer	61.9	59.8	55.7
Leukemia	86.1	80.8	83.3
Colon	75.8	80.0	76.3
Lymphoma	80.9	80.7	76.4
Lung Cancer	98.3	96.1	92.3
Average	80.6	79.5	76.8

Table 6.2: Prediction accuracy of Random Forests over five data sets with different level of noise values

data set. Baggingc4.5 also outperforms C4.5 on all data sets by up to 6.9%. Random Forests and AdaBoostc4.5 improve the accuracy on lung cancer, Lymphoma and Leukemia data sets by up to 8.3%, but fails to improve the accuracy on the Colon data set.

2. With a lower noise level of 20%, DMDT performs the best with no decreasing accuracy on average. CS4 and Random Forests also perform well with a slight decrease on average by up to 1.1%; BaggingC4.5 decreases the accuracy on average by 2.1%. AdaBoostC4.5 decreases the accuracy by 5.9% while single tree method C4.5 decreases the accuracy on average by 9.8%. DMDT keeps the accuracy unchanged on average while all other algorithms decrease their accuracy by up to 9.8%; all algorithms increase their accuracy on the Colon data set by up to 5.1% except C4.5 and CS4; Adaboostc4.5, Baggingc4.5 and DMDT increase

Data set	original	20%	60%
Breast Cancer	61.9	60.8	61.9
Leukemia	87.5	80.0	76.7
Colon	77.4	82.5	64.6
Lymphoma	85.1	88.2	75.7
Lung Cancer	96.1	71.7	70.3
Average	81.6	76.6	69.8

Table 6.3: Prediction accuracy of AdaBoostc4.5 over five data sets with different level of noise values

their accuracy on Lymphoma despite C4.5, while Random Forests and CS4 decrease their performance; All algorithms decrease the accuracy on the Breast cancer and Lung cancer data sets while Adaboostc4.5 performs the worst among the ensemble methods with the biggest decrease of 24.4%.

3. With a high noise level of 60%, all compared classification algorithms decrease the accuracy performance on average. Among the six methods, DMDT and Random Forests perform the best with the smallest decreases of 2.5% and 3.8% respectively. CS4, Baggingc4.5 and AdaBoostc4.5 decrease their accuracy by 7.6%, 10.0% and 11.8% respectively while C4.5 decreases the accuracy on average by 17.9%.

Data set	original	20%	60%
Breast Cancer	66	59.8	56.7
Leukemia	86.1	88.3	82.1
Colon	82.3	85.8	62.1
Lymphoma	85.1	85.7	84.1
Lung Cancer	97.2	86.7	81.9
Average	83.3	81.3	73.4

Table 6.4: Prediction accuracy of Baggingc4.5 over five data sets with different level of noise values

6.3 Discussions

1. Ensemble methods increase the robustness of decision tree classification. Experimental results show that ensemble decision tree methods tolerate noise values better than single tree C4.5. Since we know that Microarray data contains a huge number of noise values, it can be very difficult for a small tree to tolerate noise, hence it is not robust. For example, if a single tree is affected by noise, the whole classifier is affected and this leads to and unreliable and lower accuracy result. In contrast, an ensemble decision tree method contains multiple trees. When one tree is affected by noise, other trees might not be affected at all, and the impact of noise is reduced due to the ensemble classifier voting process.

2. The robustness of Microarray classification is affected by the diversity of ensemble methods. The essence of ensemble methods is to generate

Data set	original	20%	60%
Breast Cancer	68	67	50.8
Leukemia	98.6	95.0	89.6
Colon	82.3	81.3	70.4
Lymphoma	91.5	95.0	91.6
Lung Cancer	98.9	97.8	98.9
Average	87.9	87.2	80.3

Table 6.5: Prediction accuracy of CS4 over five data sets with different level of noise values

diversified classifiers in the decision committee. Intuitively, if individual trees in an ensemble committee are all identical, the ensemble committee is of little use for improving prediction performance over a single decision tree algorithm.

To increase the power of ensemble classification, ensemble decision tree algorithms must be able to generate a number of individual trees that are distinguished from each other [40]. Bagging does not guarantee the diversity of an ensemble tree. In contrast, CS4 and DMDT are designed to guarantee diversified trees in an ensemble committee. CS4 guarantees the diversified trees by selecting the top n unique genes from the original data set. Then each of the n genes in turn is used as the root node of an alternative tree of ensemble trees. DMDT guarantees that constructed trees are truly unique by using disjointed genes among alternative genes. The results indicate both methods perform very well

Data set	original	20%	60%
Breast Cancer	64.3	71.1	61.8
Leukemia	97.5	97.5	92.5
Colon	85.5	87.5	82.1
Lymphoma	94.1	95.0	93.2
Lung Cancer	98.9	98.3	98.4
Average	88.1	89.9	85.6

Table 6.6: Prediction accuracy of DMDT over five data sets with different level of noise values

in dealing with noise data.

The Random Forests method combines Bagging and random feature selection methods to generate alternative classifiers. Decision trees generated in this way increase the diversity among alternative trees. It still does not guarantee that every decision tree in the committee is unique. However, due to the enormous number of genes existing in Microarray data sets, Random forests has a good chance of generating a higher degree of diversified trees with little or no overlapping genes among them. So the Random decision forests algorithm should be more robust or more resistance to noise data than Bagging. The results prove that DMDT, CS4 and Random Forests outperform BaggingC4.5 and BoostingC4.5.

3. Regarding the degree of diversity of ensemble decision tree, we can

see from the results that DMDT and CS4 perform similarly on the original test data. However, when test data contains more noise values, DMDT performs better than CS4 and other ensemble methods. In CS4 ensemble trees, apart from the top genes, other genes in the trees might overlap. One noise gene may affect a number of trees. In contrast, in the DMDT algorithm, a noise gene affects only one tree, and hence DMDT should tolerate more noise than CS4. The results indicate that avoiding overlapping genes among the ensemble trees is an intuitive, simple and effective way to achieve a higher degree of diversity for ensemble decision tree methods.

4. From the results, we observe that Random forests performs similar to DMDT from the perspective of robustness. One of the possible reasons is that it is beneficial in the way it constructs the alternative trees. Unlike Bagging, Random forests constructs a tree by using random selected genes at each node. It therefore greatly increases the chance of obtaining unique trees without overlapping genes.

6.4 Summary of the chapter

In this paper, we explored the robustness of ensemble decision tree methods. Perturbed data sets with increased noise data level were used to test the robustness of the ensemble decision trees generated from C4.5, Random Forests, AdaBoostC4.5, Baggingc4.5, CS4 and DMDT. We discussed the experimental results. In the next chapter, we will investigate the dependency between gene selection methods and Microarray data classification.

Chapter 7

Gene selection for Microarray classification

In past decades, gene selection technology has been used to select the most effective genes from high dimensional Microarray data. In this Chapter, we look at how the gene selection technology affect the performance of the Microarray classification methods, such as the ensemble descision tree methods, SVMs.

Diversified multiple decision trees(DMDT)method is designed to directly deal with high dimensional Microarray data. It takes into account the nature of Microarray data which often contains high levels of noisy data. Noisy data has a large impact on the prediction performance of robust Microarray data classification. Using noise genes for classification causes a risk of decreasing the accuracy of Microarray classification. The curse of dimensionality problem slows down or even prevents the Microarray data classification process, and this is one of the major hurdles for Microarray data classification.

The basic requirement for effective Microarray classification is to work on a quality Microarray data source. To improve the performance of Microarray data classification, we should remove the noise genes as much as possible from Microarray data before classification takes place. In this chapter, we investigate the effectiveness of gene selection methods and the relationships between gene selection methods and Microarray data classification algorithms.

This chapter is organized as follows. In Section 1, we identify problems in gene expression Microarray data classification and highlight the importance of gene selection for gene expression Microarray data. In Section 2, we review a number of gene selection methods. In Section 3, we present the design of methods for comparing the accuracy of SVMs and C4.5 using different gene selection methods. In Section 4, we test four different gene selection methods with six data sets. In Section 5, we present a discussion of the results. In Section 6, we summarize the chapter.

7.1 Review of Microarray gene data

The Microarray data we acquire from Microarray technology is quite different from normal relational databases. Normal relational databases contain a small number of attributes and a large number of samples. In contrast, gene expression Microarray data usually contains a very large number of attributes but a small number of samples. With a large number of genes, it is desirable to have a large number of samples accordingly in order to build reliable Microarray classification models. The reality is that for most

Microarray experiments, a limited number of samples are available due to the huge cost of producing such Microarray data and other factors, such as privacy and availability. As an example, for cancer Microarray data, the number of samples is usually less than 100.

High dimensionality causes many problems in Microarray data analysis as follows.

1. Irrelevant and noise genes can significantly decrease the quality of Microarray data classification. Among the huge number of genes, there is unavoidably a great deal of noise which may be caused by human error, malfunctions and missing values. Apart from that, it is common that not all genes in a dataset are informative for classification. Using irrelevant and noise genes for classification causes a risk of decreasing the accuracy of classification.

2. Processing the huge number of genes causes great computational complexity in building classifiers. High dimensionality is the main cause of inefficient Microarray data classification. Actually, it renders many classification algorithms inapplicable [76]. For instance, it could take days for some classification algorithms to analyze a Microarray cancer data. Such a speed of Microarray data classification is not acceptable to any practical users. In the worst case, some classification algorithms can not be used for analying original Microarray data if the dimensions of the Microarray data are too high.

In short, high dimensionality renders many classification methods not applicable for analyzing raw gene expression Microarray data. Furthermore, high dimension Microarray data with noisy attributes leads to unreliable and low accuracy analysis results. Consequently, reducing irrelevant and removing noise gene expression values from the original Microarray data are crucial for applying classification algorithms to analyze gene expression Microarray data.

To deal with the problems caused by high dimensionality and noise data, a preprocessing phase should be introduced to reduce the noise and irrelevant genes before the Microarray data classification. As a preprocessing method of Microarray data Classification, gene selection is a very effective way for eliminating the noisy genes. In essence, gene selection aims to select a relatively small set of genes which can be used to improve the accuracy and efficiency of Microarray data classification from a high dimensional gene expression data set. Gene selection helps to clean up the existing Microarray data and therefore improve Microarray data quality. In other words, removing irrelevant and noisy genes is helpful for improving the accuracy of Microarray data classification. The resultant Microarray data classification models would therefore better characterize the true relationships among genes and hence be easier to be interpreted by biologists. Arguably, a good gene selection method not only increases the accuracy of classification through the improvement of the Microarray data quality, but also speeds up the classification process through the cutdown of high dimensionality.

7.2 Gene selection methods

In this subsection, we overview a number gene selection methods. Gene selection strives to reduce the risk of an overfitting problem, enhances the efficiency of the classification process, and increases understandability of the result [113]. Gene selection essentially is a process of selecting a subset of genes from the original data which are most predictive of categorized classes [35, 42]. Many gene selection methods have been applied to Microarray data classification in past decades [80, 90, 54, 62, 139]. Many of these have been proven to be very effective in eliminating noise data, efficient in speeding up the process of Microarray classification, and capable of increasing the accuracy performance of classification prediction.

Based on the dependency on classification algorithms, gene selection methods can be roughly divided into wrapper and filter methods [74].

A filter method performs gene selection independently from a classification method. It preprocesses a Microarray data set before the data set is used for classification analysis. Some filter gene selection methods are: ranking gene selection methods [97], and information gain gene selection method [90], Markov blanket-embedded genetic algorithm for gene selection [146], and so on. One-gene-at-a-time filter methods, such as ranking [97], signal-to-noise [123], information gain [108], are fast and scalable but do not take the relationships between genes into account. Some genes among the selected genes may have similar expression levels among classes, and they are redundant since no additional information is gained for classification algorithms by keeping them all in the dataset. To this end, Koller and Sahami [75] developed an optimal gene selection method called *Markov blanket filtering* which models feature dependencies and can eliminate redundant genes. Further to this method, Yu and Liu [141] proposed the Redundancy Based Filter(RBF) method, which is able to deal with redundant problems. Favorable results have been achieved.

In contrast, a wrapper method embeds a gene selection method within a classification algorithm. An example of a wrapper method is SVMs [59], which uses a recursive feature elimination(RFE) approach to eliminate the features iteratively in a greedy fashion until the largest margin of separation is reached. Wrapper methods are not as efficient as filter methods due to the fact that they usually run on the original high dimensional Microarray dataset. However, Kohavi and John [74] discovered that wrapper methods could significantly improve the accuracy of classification algorithms over filter methods. This discovery indicates that the performance of a classification algorithm is largely dependent on the chosen gene selection method. Nevertheless, no single gene selection method can universally improve the performance of classification algorithms in terms of efficiency and accuracy.

In past years, some filter and wrapper methods have been widely applied by many researchers in various cancer research applications, such as Signal-to-Noise ratio filter method being used for leukemia cancer research [123]; and a correlation coefficient filter method being applied for breast cancer analysis [124]. Support Vector Machines [60, 133], wrapper Approaches [15] and ensemble of neural networks [30] have also been used for cancer classification.

Many researches have shown that gene selection can improve the performance of Microarray classification [41, 85, 119, 123, 124]. The current research does not answer the question: Is there any given gene selection method which can enhance the prediction performance of different types of Microarray classification methods?

As stated in the beginning of the chapter, one objective of our research is to find out the relationship between gene selection methods and classification methods. It is undesirable if we apply a gene selection method and eventually the accuracy of Microarray data classification is decreased.

In the following sections, we investigate the dependency between gene selection methods and Microarray data classification methods.

7.3 Experimental design and methodology

Our approach is to use different existing gene selection methods to preprocess Microarray data for classification. We have carried out our experiments by comparing with benchmark algorithms SVMs and C4.5 . Note that this choice is based on the following considerations.

Consideration of benchmark systems: For a number of years, SVMs and C4.5 have been regarded as benchmark classification algorithms. SVMs was proposed by Cottes and Vapnik [34] in 1995. It has been one of the most influential classification algorithms. SVMs has been applied to many domains, for example, text categorization [70], image classification [99], cancer classification [53, 21]. SVMs can easily deal with high dimensional data sets with a wrapper gene selection method. SVMs also can achieve a higher performance compared to most existing classification algorithms.

Considering of wrapper methods: SVMs and C4.5 are not only benchmark classification systems, but each of them contains a wrapper gene selection method. SVMs uses a recursive feature elimination(RFE) approach to eliminate the features iteratively in a greedy fashion until the largest margin of separation is reached. Decision tree method can also be treated as a gene selection method. It selects the gene with the highest information gain at each step and all selected genes appear in the decision tree.

A ranking method identifies one gene at a time with differentially expressed levels among predefined classes and puts all genes in decreasing order. After a specified significance expressed level or number of genes is selected, the genes lower than the significance level or given number of genes are filtered out. The advantages of these methods is that they are intuitive, simple and easy to implement. In this study, we choose and implement four popular ranking methods collected by Cho and Won [29], namely Signal-to-Noise ratio (SNR), correlation coefficient (CC), Euclidean (EU) and Cosine (CO) ranking methods.

To evaluate the performance of different gene selection methods, three datasets from Kent Ridge Biological Data Set Repository [82] were selected. These data sets were collected from some influential journal papers, namely the breast cancer, lung cancer, Leukemia, lymphoma, colon and prostate data sets which we described in Chapter 2.

During the gene expression Microarray data preprocessing stage, we define the number of selected genes as 20, 50 and 100 and 200 for all filter gene selection methods. In our experiments, a tenfold cross-validation method is also carried out for each classification method to test its accuracy.

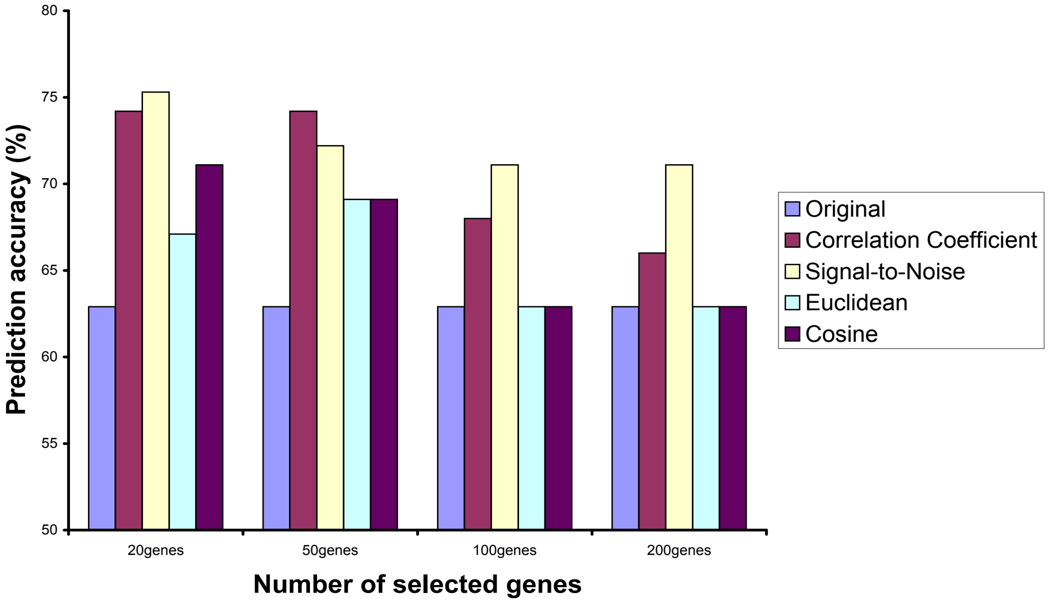


Figure 7.1: The accuracy results for C4.5 tested on Breast cancer data set

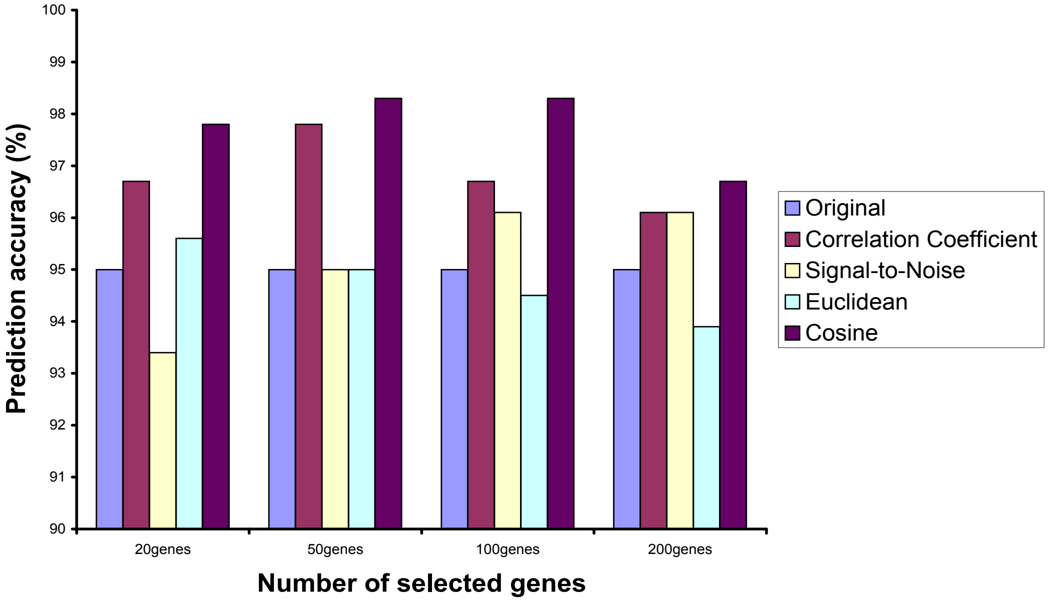


Figure 7.2: The accuracy results for C4.5 tested on lung cancer data set

7.4 Experimental results and discussions

Figure 7.1, Figure 7.5, Figure 7.4, Figure 7.2, Figure 7.3, Figure 7.6,Figure 7.7, Figure 7.11, Figure 7.10, Figure 7.8, Figure 7.9 and Figure 7.12, show the detailed results for SVMs and C4.5 tested on six different datasets preprocessed by four different filter gene selection methods.

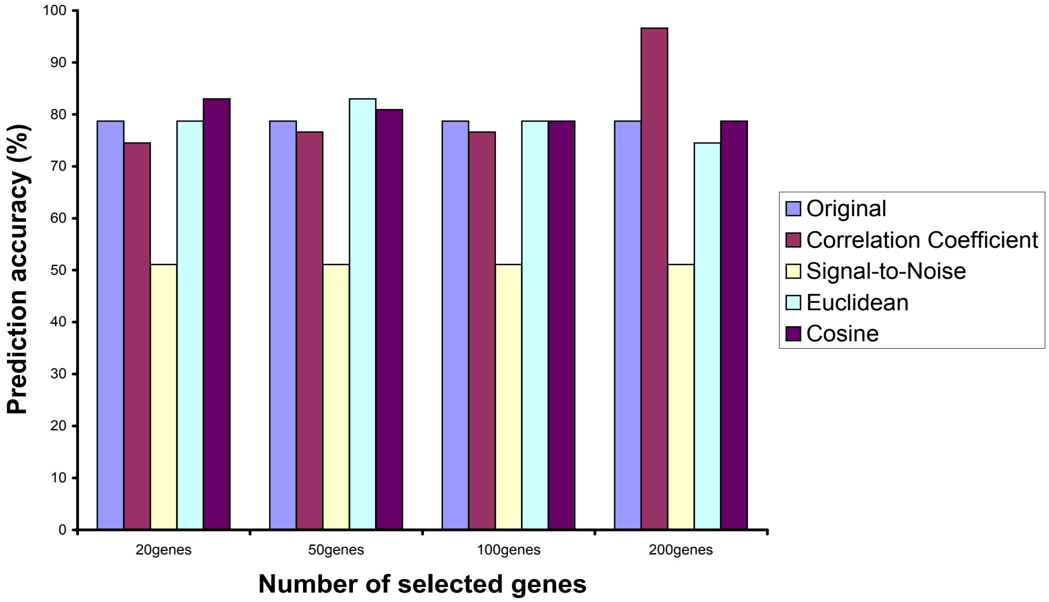


Figure 7.3: The accuracy results for C4.5 tested on Lymphoma data set

From these experimental results, we make the following observations.

1. When Microarray data sets are preprocessed, SVMs improves its prediction accuracy on Breast Cancer and Lymphoma data sets only. Signal-to-Noise and Correlation coefficient methods performed best and improved the accuracy up to 16.5% and 15.5% respectively on Cancer data. The Cosine method also improved the accuracy by up to 7.2%

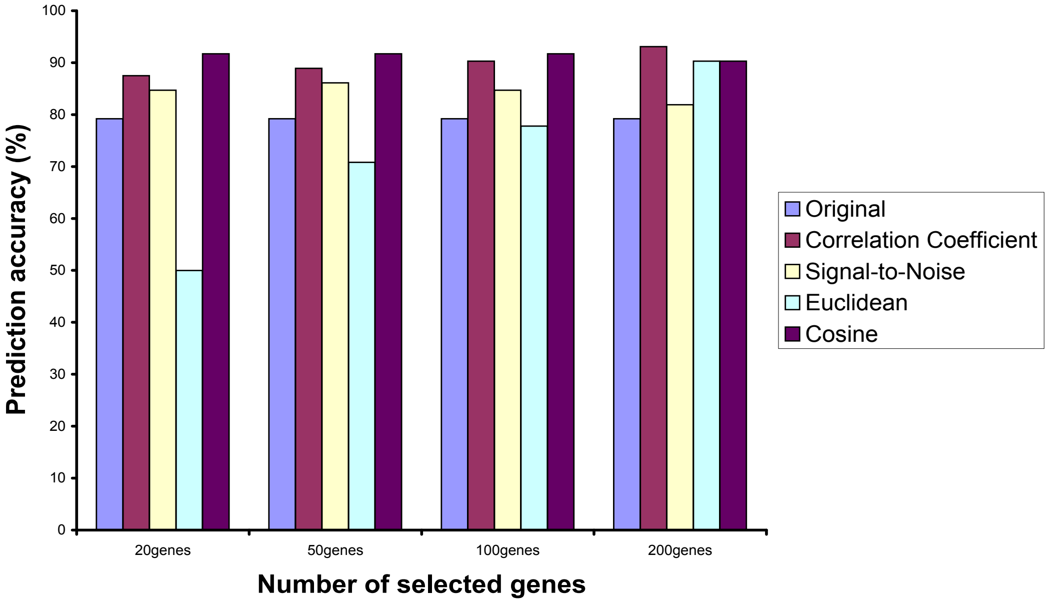


Figure 7.4: The accuracy results for C4.5 tested on Leukemia data set

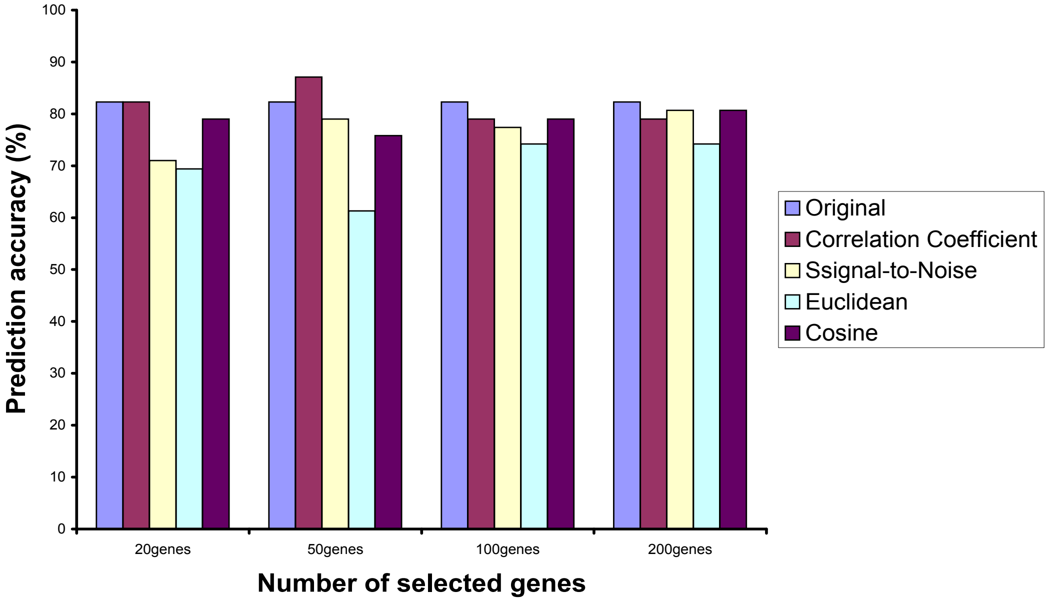


Figure 7.5: The accuracy results for C4.5 tested on Colon data set

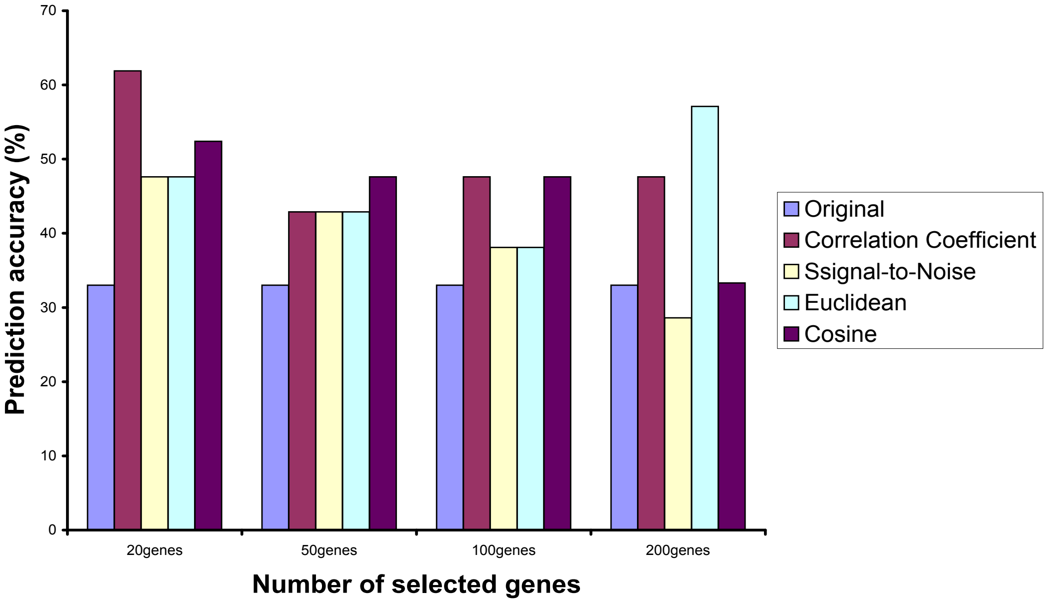


Figure 7.6: The accuracy results for C4.5 tested on Prostate data set

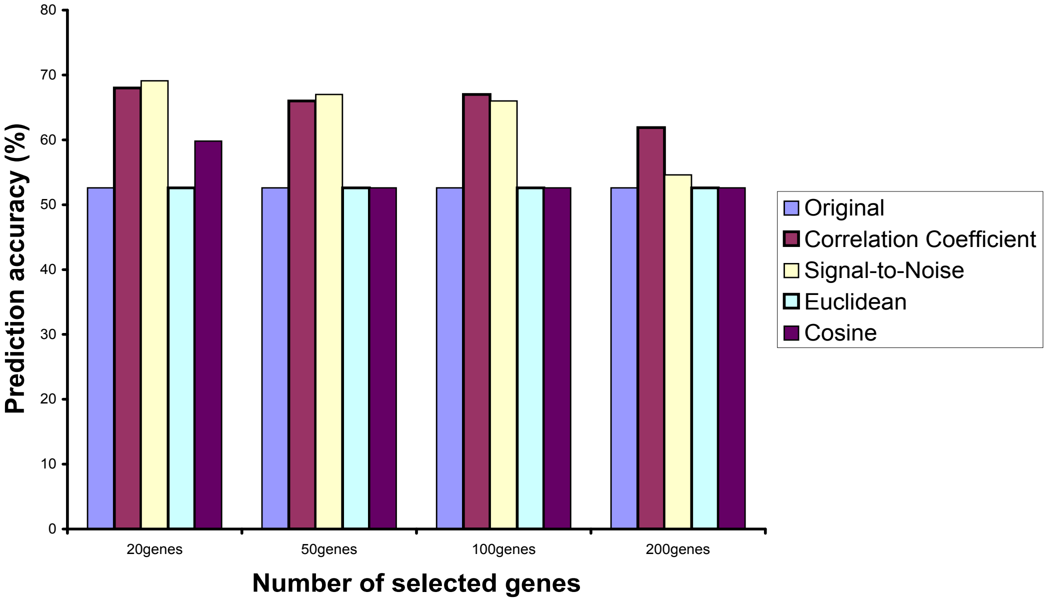


Figure 7.7: The accuracy results for SVM tested on Breast cancer data set

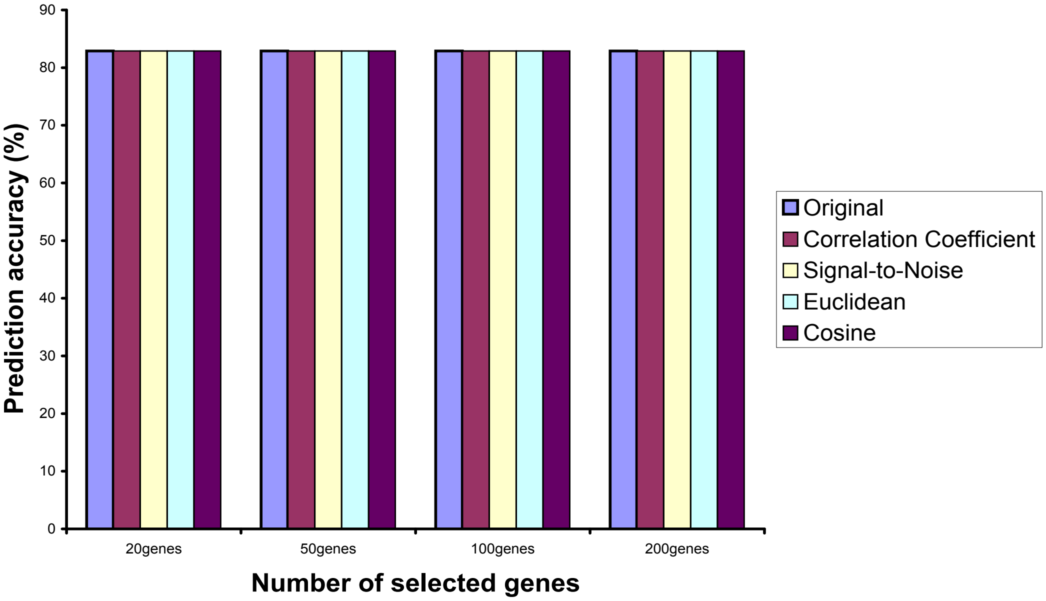


Figure 7.8: The accuracy results for SVM tested on lung cancer data set

on Cancer data. On the Lymphoma data set, the Correlation coefficient method is the only method which improved the accuracy performance over original data set by up to 2.2% while other methods were not able to improve the accuracy. None of the gene selection methods improved nor decreased the prediction accuracy based on Lung Cancer, Leukemia, Colon and Prostate data sets with 200, 100, 50 and 20 genes. Instead, the accuracy performance is kept unchanged.

2. The performance of C4.5 improves its prediction accuracy by up to 28.6%. Among the four gene selection methods, Correlation coefficient is the most effective preprocessing method with an improvement of accuracy up to 7.6% on average, followed by Cosine 7.3%, and Signal-to-Noise 6.0%. The Signal-to-Noise gene selection method performed

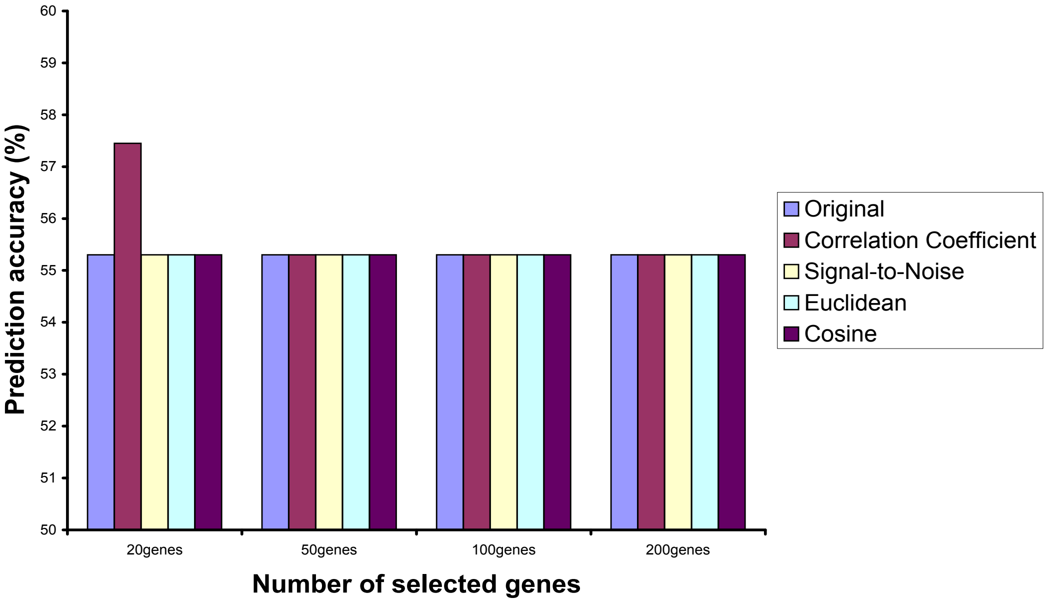


Figure 7.9: The accuracy results for SVM tested on Lymphoma data set

consistently better on Breast Cancer and Leukemia data sets with improved accuracy by up to 12.4%, but failed on the other cancer data sets. Euclidean in contrast performed worst among the compared methods, decreasing the accuracy performance on all provided cancer data sets except Breast and Prostate by up to 18.7%.

3. The experimental results show that with preprocessing, the number of genes selected has an affect on some classification methods in terms of performance accuracy. In the figures for C4.5, the highest accuracy for all cancer data sets except the prostate cancer data set are based on 50 genes; while the highest accuracy for prostate cancer data set is based on 20 genes. The overall performance is better when data sets contain 50. However, the number of genes selected has little impact on

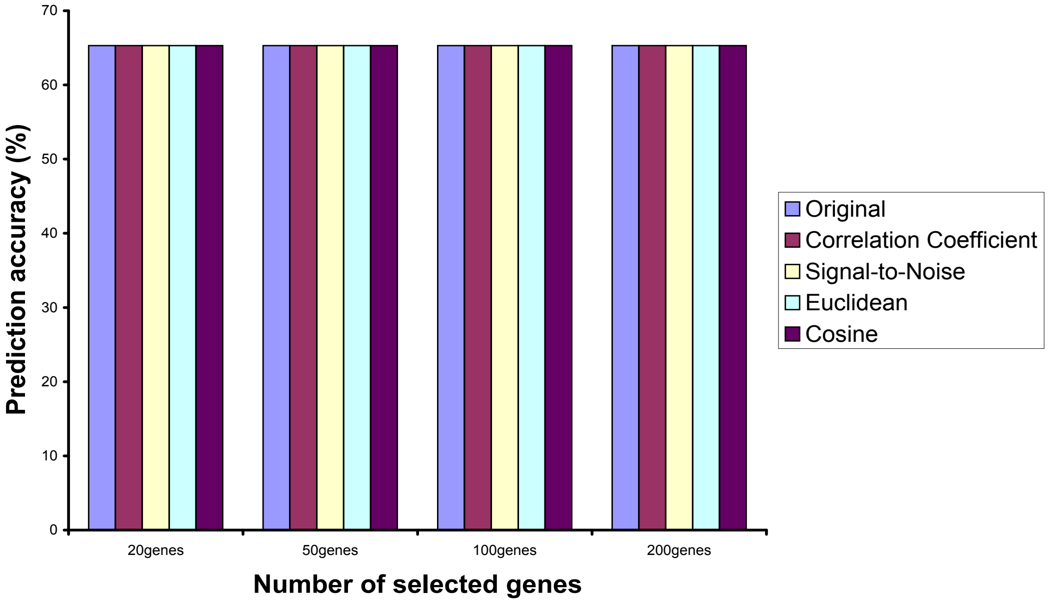


Figure 7.10: The accuracy results for SVM tested on Leukemia data set

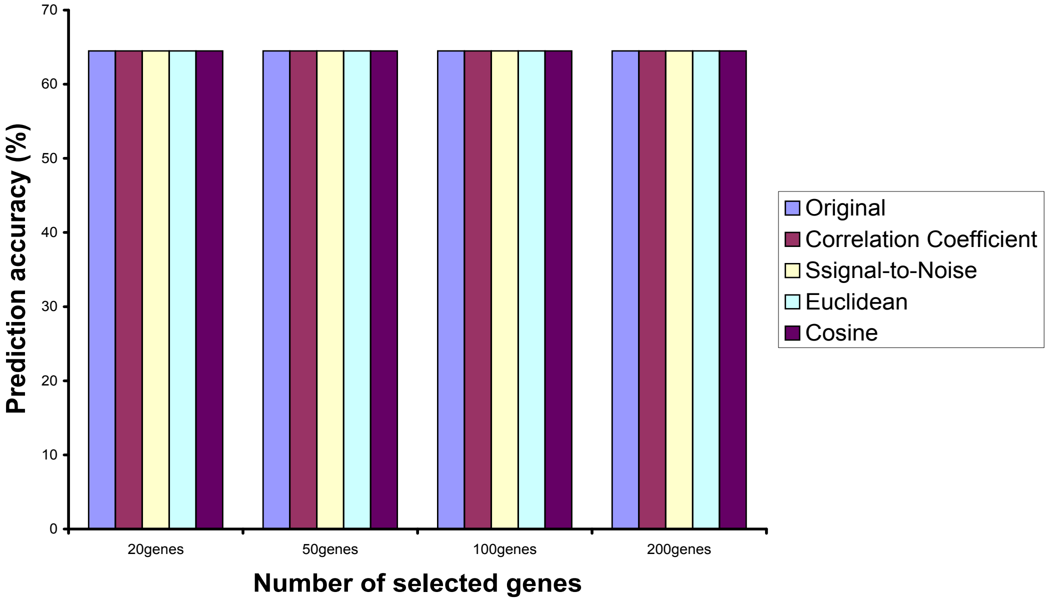


Figure 7.11: The accuracy results for SVM tested on Colon data set

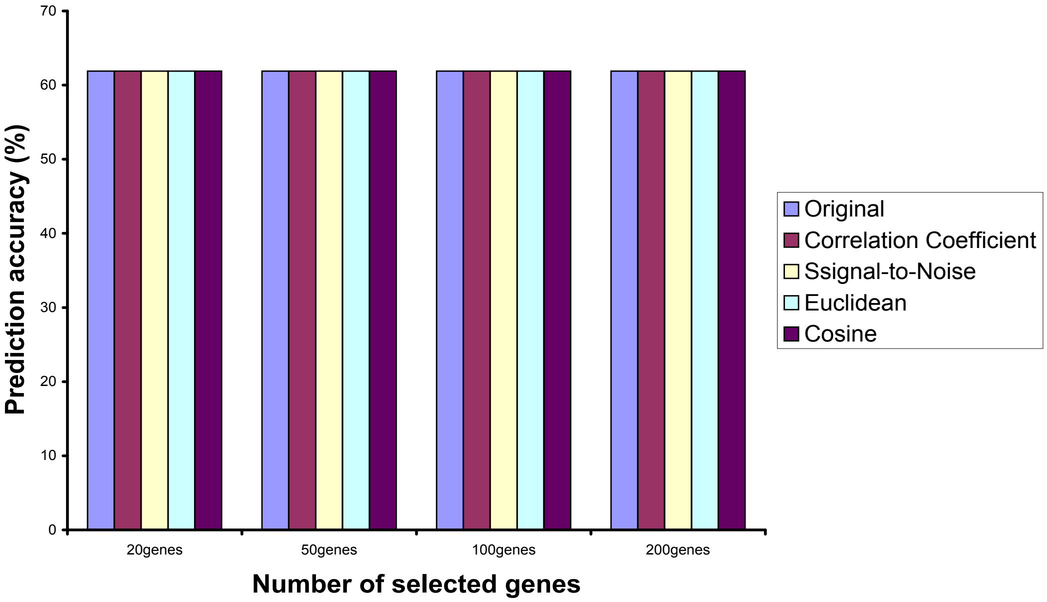


Figure 7.12: The accuracy results for SVM tested on Prostate data set

the performance of SVMs.

7.5 Discussion of experimental results

In this section, we discuss the implication of gene selection methods upon the classification methods.

(a) The results indicate that gene selection improves the performance of classification methods in general. using a suitable gene selection method with C4.5 increases the accuracy performance of C4.5 dramatically. For SVMs, its performance remained unchanged unless a very small size of genes was selected. Moreover, gene selection does not decrease the accuracy performance of SVMs. This result ensures that we can reduce the number of genes to a smaller size without hurting the accuracy performance of classification. This is very helpful for noisy Microarray data classification as most irrelevant genes in Microarray data classification would be reduced. It increases the performance of classification significantly in terms of speeding up the efficiency of Microarray data classification.

(b) These results indicate that not all gene selection methods help the performance of Microarray classification methods in terms of improving the prediction accuracy of classification. Their performance depends on which Microarray classification method they are combined with. For C4.5, with the help of some gene selection, such as the Correlation coefficient method, the accuracy performance improved significantly. The Signal-to-Noise method generated mixed results combined with C4.5; while the Euclidean method is not a suitable gene selection method for C4.5 as it failed to improved the accuracy performance of C4.5 on most data sets. So to apply gene selection to C4.5, we have to seriously consider which gene selection algorithm to use to achieve maximum improvement. With SVMs, only the Correlation coefficient method managed to improve the accuracy performance on up to two data sets.

(c) Gene selection may have little impact on some classification methods. The figures show that SVMs is insensitive to the gene selection methods used and hence data preprocessing does not increase its performance in most cases. This indicates that the SVMs classification method can initially handle noise data very well. Moreover, it would require little effort to select a gene selection method for SVMs.

(d) The observations indicate that a data set with less genes or attributes does not necessarily guarantee the highest prediction accuracy. The number of genes selected by a preprocessing method should not be too small. At this stage, the objective of gene selection is just to eliminate irrelevant and noise genes. However, less informative genes can sometimes enhance the power of classification if they are co-related with the most informative genes. If the number of genes has been eliminated too harshly, it can also decrease the performance of the classification. So during the preprocessing, we need to make sure that a reasonable number of genes are left for classification.

(e) Those results remind us that when selecting the gene selection method for data preprocessing, we must consider which classification method the gene selection is for. For example, if we select SVMs as a classification algorithm, then the Correlation coefficient or Signal-to-Noise gene selection methods are better for data preprocessing. An inappropriate choice can only harm the power of classification prediction.

7.6 Summary of the chapter

In this chapter, we introduced several gene selection methods. We discussed the advantage of applying gene selection methods for Microarray data classification. We have conducted some experiments on different existing gene selection methods for preprocessing gene expression Microarray data for classification by SVMs and C4.5, which themselves contain a wrapped method. In the next chapter, we conclude our research findings and describe the future work.

Chapter 8

Conclusions and future work

In this thesis, we described and investigated the development of a robust and accurate Microarray data classification method. The research presented in this thesis has proved practically that medical researchers or biologists can take advantage of Microarray technologies to help their medical research, in particular with cancer studies. More specifically, (1) we proposed an algorithm of diversified multiple decision trees method (DMDT), which makes use of a set of diversified trees in the decision committee; (2) then compared DMDT with some well-known ensemble methods; (3) we also investigated how gene selection affect classification accuracy.

This chapter is organized as follows. In Section 1, we summarize all the research findings and experimental results. In Section 2, we reflect on our research. In Section 3, we highlight some possible study directions for the future.

8.1 Summary of our research

8.1.1 New diversified multiple decision tree algorithm

In this research, we concentrated on a study which uses diversified multiple decision trees to classify Microarray data. DMDT can maximally diversify trees in the ensemble decision tree committee. The maximal diversity is attributed to the fact that trees in the committee share no common genes; genes in trees are not randomly selected, but are chosen by C4.5 in a covering-algorithm manner. The experimental results demonstrated that the proposed method and another existing diversified decision tree method, which diversifies trees by using distinct tree roots, are more accurate on average than the Bagging, Boosting and Random Forests methods. This study also indicates that diversity improves the classification accuracy of ensemble classification on Microarray data.

8.1.2 Robustness analysis of Microarray classification

Apart from the accuracy of Microarray data classification methods, we also explored the robustness of many ensemble decision tree methods including DMDT. Perturbed data sets with increased noise data level were used to test the robustness of the ensemble decision trees generated from C4.5, Random Forests, AdaBoostC4.5, Baggingc4.5, CS4 and DMDT. We observed that DMDT, CS4 and Random Forests tolerate noise better than Baggingc4.5 and Boostingc4.5 methods do, particularly with increasing levels of noise data. Experimental results indicate that Random Forests is comparable to DMDT regarding the robustness issue and performs better than CS4 AdaBoostC4.5 and BaggingC4.5 on noise data, while CS4 is comparable to DMDT on original data sets. However, when the noise level increases in the training and test data, DMDT performs better than CS4. Experimental results also show that ensemble decision tree methods tolerate noise values better than single tree C4.5 does.

These observations imply that the proposed DMDT with unique trees can tolerate noise values better than the ensemble methods with ensemble trees containing overlapping genes. It indicates that maximally diversifying the outputs of ensemble trees is a simple and effective way to improve the classification prediction accuracy of Microarray classification.

8.1.3 Gene selection methods for Microarray data classification

In addition to the accuracy and robustness analysis, we have looked into a technique to improve the quality of Microarray data sets using the technology of preprocessing gene expression Microarray data for classification by SVMs and C4.5, which themselves contain a wrapped method. We observed that although in general the performance of SVMs and C4.5 are improved by using the preprocessed datasets rather than original data sets in terms of accuracy and efficiency, not all gene selection methods help improve the performance of classification. The rule-of-thumb is that some gene selection methods are suitable for some specific classification algorithms. For example, if we select SVMs as the classification algorithm, then a Correlation coefficient or

Signal-to-Noise gene selection method is better for data preprocessing. On the contrary, an inappropriate choice can only harm the power of prediction. Our results also implied that with preprocessing, the number of genes selected affects the classification accuracy.

8.2 Reflection on our research

DMDT provides more accurate and robust classification than the many existing systems and the results generated are more easily evaluated and interpreted by users. It might be even better to combine filter and wrapper methods to further improve the accuracy performance of gene expression Microarray data classification. The study also reminds us that not all filter gene selection methods help improve the performance of classification for a given classification algorithm.

8.3 Future work

Although the DMDT algorithm has many advantages over many well-known ensemble decision tree methods, we certainly can further improve on our research in the future. More specifically, we may expand our existing work on the following aspects.

1. Based on the experimental design in this thesis, in the future we will conduct extended robustness experiments under the following conditions.

(a) Training data contains a higher level of noise than test data. Under this condition, the test data are original data with 0 level of noise. We increase the certain level of noise in training data only for each experiment. This condition is tested for the situation when the historical data has lower quality than future incoming data.

(b) Test data contains a higher level of noise data. Under this condition, the training data are original historical data with 0 level of noise. We increase the certain level of noise in test data only for each experiment. this condition is tested for the situation when the future incoming data contains more noise data than the historical data.

2. A strong diversified committee should include trees that are independent of each other. It seems that we should not use two correlated genes in different trees. To simplify the explanation, let us consider that each tree only contains one gene. If all genes in trees are independent, it is not easy to achieve a consensus . What we really want is that all trees are correlated for corrections and uncorrelated for errors. In this case, no simple remedy works. This is one direction to further improve DMDT and CS4.

3. In all of our experimental study, we found that the deviations among the constructed trees are big. The question arises whether we need to select the most accurate trees from the constructed trees in order to further improve the performance of the maximally diversified multiple

decision tree algorithm. In future, we will explore a best tree selection method and combine it into this maximally diversified multiple decision tree algorithm.

Bibliography

[1] http://www.troy.k12.ny.us/thsbiology/skinny/skinny_genetics.html.

[2] http://faculty.uca.edu/~johnc/rnaprot1440.htm.

[3] http://www.cs.tau.ac.il/~rshamir/algmb/00/scribe00/html/lec11/node11.html.

[4] Normalization for ccdna microarray data: a robust composite mehtod addressing single and multiple slide systematic variation. *Nucleic Acids Research*, 30:4e15, 2002.

[5] M. Aksela and J. Laaksonen. Using diversity of errors for selecting members of a committee classifier. *Pattern Recognition*, 39(4):608–623, 2006.

[6] A. Alizadeh, M. Eishen, E. Davis, and C. M. et. al. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, 2000.

[7] A. Amin. Recognition of printed arabic text based on global features and decision tree learning techniques. *Pattern Recognition*, 33(8):1309–1323, 2000.

[8] H. Amir-Kroll, A. Sadot, I. R. Cohen, and D. Harel. Gemcell: A generic platform for modeling multi-cellular biological systems. *Theor. Comput. Sci*, 391(3):276–290, 2008.

[9] R. Andrews, S. Bajcar, J. W. Grzymala-Busse, Z. S. Hippe, and C. Whiteley. Optimization of the ABCD formula for melanoma diagnosis using C4.5, a data mining system. In S. Tsumoto, R. Slowinski, H. J. Komorowski, and J. W. Grzymala-Busse, editors, *Rough Sets and Current Trends in Computing*, volume 3066 of *Lecture Notes in Computer Science*, pages 630–636. Springer, 2004.

[10] P. Arena, L. Fortuna, and L. Occhipinti. DNA chip image processing via cellular neural networks. In *ISCAS (3)*, pages 345–348. IEEE, 2001.

[11] M. Baglioni, B. Furletti, and F. Turini. DrC4.5: Improving C4.5 by means of prior knowledge. In H. Haddad, L. M. Liebrock, A. Omicini, and R. L. Wainwright, editors, *SAC*, pages 474–481. ACM, 2005.

[12] E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36(1-2):105–139, 1999.

[13] E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36(1-2):105–139, 1999.

[14] B. Bergeron. *Bioinformatics Computing.* Prentice Hall,New Jersey, 2003.

[15] R. Blanco, P. Larrañaga, I. Inza, and B. Sierra. Gene selection for cancer classification using wrapper approaches. *IJPRAI*, 18(8):1373–1390, 2004.

[16] A.-L. Boulesteix, C. Porzelius, and M. Daumer. Microarray-based classification and clinical predictors: on combined classifiers and additional predictive value. *Bioinformatics*, 24(15):1698–1706, 2008.

[17] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[18] L. Breiman. Random forests–random features. Technical Report 567, University of California, Berkley, 1999.

[19] L. Breiman and P. Spector. Submodel selection and evaluation in regression: The x-random case. *International Statistical Review*, 60:291–319, 1992.

[20] L. J. Breiman, R. A. Olshen, and C. J. Stone. *classification and regression trees.* Chapman and Hall, New York, 1984.

[21] M. Brown, W. Grundy, D. Lin, N. Cristianini, C. Sugnet, T. Furey, M. Jr, and D. Haussler. Knowledge-based analysis of microarray gene expression data by using suport vector machines. In *Proc. Natl. Acad. Sci.*, volume 97, pages 262–267, 2000.

[22] M. Brown, W. N. Grundy, D. Lin, N. Cristianini, C. Sugnet, M. Ares, and D. Haussler. Support vector machine classification of microarray gene expression data. Technical Report UCSC-CRL-99-09, University of California, Santa Cruz, Santa Cruz, CA 95065, 1999.

[23] E. by Philip E. Bourne and H. Weissig. *Structural Bioinformatics*. Wiley-Liss, Hoboken, New Jersey, 2003.

[24] O. Çamoglu, T. Can, A. K. Singh, and Y.-F. Wang. Decision tree based information integration for automated protein classification. *J. Bioinformatics and Computational Biology*, 3(3):717–742, 2005.

[25] A. Cavallo and A. C. R. Martin. Mapping SNPs to protein sequence and structure data. *Bioinformatics*, 21(8):1443–1450, 2005.

[26] J. Cendrowska. Prism: An algorithm for inducing modular rules. *International Journal of Man-Machine Studies*, 27(4):249–370, 1987.

[27] N. V. Chawla, T. E. Moore, K. W. Bowyer, L. O. Hall, C. Springer, and W. P. Kegelmeyer. Investigation of bagging-like effects and decision trees versus neural nets in protein secondary structure prediction. In *BIOKDD*, pages 50–59, 2001.

[28] S.-C. Chen, M.-L. Shyu, M. Chen, and C. Zhang. A decision tree-based multimodal data mining framework for soccer goal detection. In *ICME*, pages 265–268. IEEE, 2004.

[29] S.-B. Cho and H.-H. Won. Machine learning in dna microarray analysis for cancer classification. In *CRPITS '19: Proceedings of the First Asia-*

Pacific bioinformatics conference on Bioinformatics 2003, pages 189–198, Darlinghurst, Australia, Australia, 2003. Australian Computer Society, Inc.

[30] S.-B. Cho and H.-H. Won. Cancer classification using ensemble of neural networks with multiple significant gene subsets. *Appl. Intell*, 26(3):243–250, 2007.

[31] F. Chu and L. Wang. Applications of support vector machines to cancer classification with microarray data. *Int. J. Neural Syst*, 15(6):475–484, 2005.

[32] J. Cohen. Bioinformatics—an introduction for computer scientists. *ACM Comput. Surv.*, 36(2):122–158, 2004.

[33] I. H. G. S. Consortium. Finishing the euchromatic sequence of the human genome. *Nature*, 431:931–945, 2004.

[34] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[35] M. Dash and H. Liu. Feature selection for classification. *Intell. Data Anal*, 1(1-4):131–156, 1997.

[36] D. DeGroot. Modeling and simulating the brain as a system. In *PADS*, page 3. IEEE Computer Society, 2005.

[37] L. Dematté, C. Priami, and A. Romanel. BetaWB: modelling and simulating biological processes. In G. A. Wainer, editor, *SCSC*, pages 777–784. Simulation Councils, Inc, 2007.

[38] M. Dettling. Bagboosting for tumor classification with gene expression data. *Bioinformatics*, 20(18):3583–3593, 2004.

[39] L. Devroye. *Non-Uniform Random Variate Generation.* Springer-Verlag, New York, 1986.

[40] T. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging,boosting, and randomization. *Machine Learning*, 40(2):139–158, 1998.

[41] C. H. Q. Ding. Unsupervised feature selection via two-way ordering in gene expression analysis. *Bioinformatics*, 19(10):1259–1266, 2003.

[42] C. H. Q. Ding and H. Peng. Minimum redundancy feature selection from microarray gene expression data. *J. Bioinformatics and Computational Biology*, 3(2):185–206, 2005.

[43] E. Domany. Analysis of DNA-chip and antigen-chip data: studies of cancer, stem cells and autoimmune diseases. *Computer Physics Communications*, 169(1-3):183–187, 2005.

[44] P. Domingos and M. J. Pazzani. Beyond independence: Conditions for the optimality of the simple bayesian classifier. In *Proceedings of IEEE International Conference on Machine Learning*, pages 105–112, 1996.

[45] B. Efron and R. Tibshirani. Cross-validation and the bootstrap: Estimating the error rate of a prediction rule. Technical Report 176, Stanford University, May 1995.

[46] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. In *Proceedings of the National Academy of Sciences*, volume 95, pages 14863–14868, 1998.

[47] B. Enfron and R. Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall, New York, 1993.

[48] D. S. et al. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 1:203–209, 2002.

[49] U. A. et al. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *PNAS*, 96:6745–6750, 1999.

[50] D. Faller, H. U. Voss, J. Timmer, and U. Hobohm. Normalization of DNA-microarray data by nonlinear correlation maximization. *Journal of Computational Biology*, 10(5):751–762, 2003.

[51] K. A. Frenkel. The human genome project and informatics. *Commun. ACM*, 34(11):40–51, 1991.

[52] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, pages 148–156, 1996.

[53] T. S. Furey, N. Christianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Hauessler. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10):906–914, 2000.

[54] C. Furlanello, M. Serafini, S. Merler, and G. Jurman. Entropy-based gene ranking without selection bias for the predictive classification of microarray data. *BMC Bioinformatics*, 4:54, 2003.

[55] M. Gal-Or, J. H. May, and W. E. Spangler. Using decision tree models and diversity measures in the selection of ensemble classification models. In N. C. Oza, R. Polikar, J. Kittler, and F. Roli, editors, *Multiple Classifier Systems*, volume 3541 of *Lecture Notes in Computer Science*, pages 186–195. Springer, 2005.

[56] A. Ghosh and B. Parai. Protein secondary structure prediction using distance based classifiers. *Int. J. Approx. Reasoning*, 47(1):37–44, 2008.

[57] G. Gordon, R. Jensen, L.-L. Hsiao, and S. G. et. al. Translation of microarray data into clinically relevant cancer diagnostic tests using gege expression ratios in lung cancer and mesothelioma. *Cancer Research*, 62:4963–4967, 2002.

[58] J. Gray and P. J. Shenoy. Rules of thumb in data engineering. In *Proceedings of IEEE International Conference on Data Engineering*, pages 3–12, 2000.

[59] 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.

[60] 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.

[61] I. B. Hallgrímsdóttir and B. Sturmfels. Resultants in genetic linkage analysis. *J. Symb. Comput*, 41(2):125–137, 2006.

[62] S.-Y. Ho, C.-C. Lee, H.-M. Chen, and H.-L. Huang. Efficient gene selection for classification of microarray data. In *IEEE Congress on Evolutionary Computation*, pages 1753–1760. IEEE, 2005.

[63] T. K. Ho. Random decision forests. In *ICDAR*, page 278, 1995.

[64] T. K. Ho. The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Mach. Intell*, 20(8):832–844, 1998.

[65] H. Hu and J. Li. Using association rules to make rule-based classifiers robust. In H. E. Williams and G. Dobbie, editors, *Sixteenth Australasian Database Conference (ADC2005)*, volume 39 of *CRPIT*, pages 47–54, Newcastle, Australia, 2005. ACS.

[66] X. Huang, W. Pan, S. J. Park, X. Han, L. W. Miller, and J. Hall. Modeling the relationship between LVAD support time and gene expression changes in the human heart by penalized partial least squares. *Bioinformatics*, 20(6):888–894, 2004.

[67] I. Hussain, O. Ormandjieva, and L. Kosseim. Automatic quality assessment of SRS text by means of a decision-tree-based text classifier. In *QSIC*, pages 209–218. IEEE Computer Society, 2007.

[68] T. Ideker, V. Thorsson, J. Ranish, R. Christmas, and et.al. Integrated genomic and proteomic analyses of a systematically perturbed metabolic network. *Science*, 292:929–933, 2001.

[69] E. P. III, A. Ardekani, B. Hitt, and P. L. et. al. Use of proteomic patterns in serum to identify ovarian cancer. *The lancet*, 359:572–577, 2002.

[70] T. Joachims. Text categorization with support vector machines: learning with many relevant features. In C. Nédellec and C. Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, number 1398, pages 137–142. Springer Verlag, Heidelberg, DE, 1998.

[71] N. C. Jones and P. A. Pevzner. Comparative genomics reveals unusually long motifs in mammalian genomes. In *ISMB (Supplement of Bioinformatics)*, pages 236–242, 2006.

[72] Y. S. Kim, W. N. Street, and F. Menczer. Feature Selection in Unsupervised Learning via Evolutionary Search. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2000.

[73] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1137–1143, 1995.

[74] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[75] D. Koller and M. Sahami. Toward optimal feature selection. In *International Conference on Machine Learning*, pages 284–292, 1996.

[76] D. Koller and M. Sahami. Toward optimal feature selection. In *ICML*, pages 284–292, 1996.

[77] L. I. Kuncheva and C. J. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51(2):181–207, 2003.

[78] M. Lajoie, D. Bertrand, and N. El-Mabrouk. Evolution of tandemly arrayed genes in multiple species. In G. Tesler and D. Durand, editors, *RECOMB-CG*, volume 4751 of *Lecture Notes in Computer Science*, pages 96–109. Springer, 2007.

[79] H.-H. Lee, N.-T. Nguyen, and J.-M. Kwon. Bearing diagnosis using time-domain features and decision tree. In D.-S. Huang, L. Heutte, and M. Loog, editors, *ICIC (2)*, volume 4682 of *Lecture Notes in Computer Science*, pages 952–960. Springer, 2007.

[80] Z.-J. Lee. An integrated algorithm for gene selection and classification applied to microarray data of ovarian cancer. *Artificial Intelligence in Medicine*, 42(1):81–93, 2008.

[81] G. Leshem and Y. Ritov. Traffic flow prediction using adaboost algorithm with random forests as a weak learner. *Transactions on engineering, computing and technology*, January 2007.

[82] J. Li and H. Liu. Kent ridge bio-medical data set repository. http://sdmc.lit.org.sg/gedatasets/datasets.html, 2002.

[83] J. Li and H. Liu. Ensembles of cascading trees. In *ICDM*, pages 585–588, 2003.

[84] J. Li, H. Liu, S.-K. Ng, and L. Wong. Discovery of significant rules for classifying cancer diagnosis data. In *ECCB*, pages 93–102, 2003.

[85] S. Li, X. Wu, and X. Hu. Gene selection using genetic algorithm and support vectors machines. *Soft Comput*, 12(7):693–698, 2008.

[86] W. Li, J. Han, and J. Pei. CMAR: Accurate and efficient classification based on multiple class-association rules. In *Proceedings of IEEE International Conference on Data Mining*, pages 369–376, 2001.

[87] A. Lindlöf, M. Bräutigam, A. Chawade, B. Olsson, and O. Olsson. Identification of cold-induced genes in cereal crops and arabidopsis through comparative analysis of multiple EST sets. In S. Hochreiter and R. Wagner, editors, *BIRD*, volume 4414 of *Lecture Notes in Computer Science*, pages 48–65. Springer, 2007.

[88] B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *Proceedings of ACM International Conference on Knowledge Discovery and Data Mining*, pages 80–86, 1998.

[89] H. Liu and R. Setiono. Chi2: Feature selection and discretization of numeric attributes, 1995.

[90] X. Liu, A. Krishnan, and A. Mondry. An entropy-based gene selection method for cancer classification using microarray data. *BMC Bioinformatics*, 6:76, 2005.

[91] D. Lockhart, H. Dong, and M. B. et al. Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nature Biotechnology*, 14:1675–1680, 1996.

[92] K. Lu, A. P. Patterson, L. Wang, R. Marquez, and E. Atkinson. Selection of potential markers for epithelial ovarian cancer with gene expression arrays and recursive descent partition analysis. *Clin Cancer Res*, 10:291–300, 2004.

[93] R. Maclin and D. W. Opitz. An empirical evaluation of bagging and boosting. In *AAAI/IAAI*, pages 546–551, 1997.

[94] O. Maron and T. Lozano-Pérez. A framework for multiple-instance learning. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10, pages 570–576. The MIT Press, 1998.

[95] R. S. Michalski. A theory and methodology of inductive learning. *Artificial Intelligence*, 20:111–116, 1983.

[96] M. Mramor, G. Leban, J. Demsar, and B. Zupan. Visualization-based cancer microarray data classification analysis. *Bioinformatics*, 23(16):2147–2154, 2007.

[97] S. Mukkamala, Q. Liu, R. Veeraghattam, and A. H. Sung. Feature selection and ranking of key genes for tumor classification: Using microarray gene expression data. In L. Rutkowski, R. Tadeusiewicz, L. A. Zadeh, and J. M. Zurada, editors, *ICAISC*, volume 4029 of *Lecture Notes in Computer Science*, pages 951–961. Springer, 2006.

[98] K. Muralidhar and R. Sarathy. Security of random data perturbation methods. *ACM Trans. Database Syst.*, 24(4):487–493, 1999.

[99] E. Osuna, R. Freund, and F. Girosi. Training support vector machines:an application to face detection. In *In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997.

[100] T. Park, S.-G. Yi, S.-H. Kang, S. Y. Lee, Y.-S. Lee, and R. Simon. Evaluation of normalization methods for microarray data. *BMC Bioinformatics*, 4:33, 2003.

[101] D. Partridge and W. Krzanowski. Distinct failure diversity in multiversion software. (personal communication. Technical report, Dept. Computer Science, University of Exeter, sec@dcs.exeter.ac.uk, 1999.

[102] P. Pavlidis, C. Tang, and W. S. Noble. Classification of genes using probabilistic models of microarray expression profiles. In *BIOKDD*, pages 15–21, 2001.

[103] S. D. Peddada, E. K. Lobenhofer, L. Li, C. A. Afshari, C. R. Weinberg, and D. M. Umbach. Gene selection and clustering for time-course and dose-response microarray experiments using order-restricted inference. *Bioinformatics*, 19(7):834–841, 2003.

[104] M. J. Pérez-Jiménez and F. J. Romero-Campero. P systems, a new computational modelling tool for systems biology. 4220:176–197, 2006.

[105] A. Petrov and S. Shams. Microarray image processing and quality control. *VLSI Signal Processing*, 38(3):211–226, 2004.

[106] K. Polat and S. Günes. Classification of epileptiform EEG using a hybrid system based on decision tree classifier and fast fourier transform. *Applied Mathematics and Computation*, 187(2):1017–1026, 2007.

[107] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1,1:81–106, 1986.

[108] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, California, 1993.

[109] R. J. Quinlan. Bagging, boosting, and c4.5. In *AAAI/IAAI, Vol. 1*, pages 725–730, 1996.

[110] J. Reumers, S. Maurer-Stroh, J. Schymkowitz, and F. Rousseau. SNPeffect v2.0: a new step in investigating the molecular phenotypic effects of human non-synonymous SNPs. *Bioinformatics*, 22(17):2183–2185, 2006.

[111] S. Ruggieri. Efficient C4.5. *IEEE Trans. Knowl. Data Eng*, 14(2):438–444, 2002.

[112] D. E. Rumelhart, G. E. Hinton, and J. R. Williams. Learning internal representations by error propagation. In D. E. Rumelhart and J. L. McClelland, editors, *Proceedings of Parallel Distributed Processing*, pages 318–362, Cambridge, MA, 1986. MIT Press.

[113] Y. Saeys, I. Inza, and P. Larrañaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, 2007.

[114] A. Santin, F. Zhan, S. Bellone, and M. Palmieri. Gene expression profiles in primary ovarian serous papillary tumors and normal ovarian epithelium: idnetification of candidate molecular markers for ovarian cancer diagnosis and therapy. *International Journal of Cancer*, 112:14–25, 2004.

[115] D. R. B. P. Schena M, Shalon D. Wuantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, 270:467–470, 1995.

[116] J. Shao. Linear model selection by cross-validation. *American Statistical Association*, 88(422):486–494, 1993.

[117] A. Sharma and K. K. Paliwal. Cancer classification by gradient LDA technique using microarray gene expression data. *Data Knowl. Eng*, 66(2):338–347, 2008.

[118] D. K. Slonim, P. Tamayo, J. P. Mesirov, T. R. Golub, and E. S. Lander. Class prediction and discovery using gene expression data. In *In Proceedings of the Forth Annual Conference on Research in Computational Molecular Biology*, pages 263–272, 2000.

[119] M. Song and S. Rajasekaran. A greedy correlation-incorporated SVM-based algorithm for gene selection. In *AINA Workshops (1)*, pages 657–661. IEEE Computer Society, 2007.

[120] S. Sundaresh, D. L. Doolan, S. Hirst, Y. Mu, B. Unal, D. H. Davies, P. L. Felgner, and P. Baldi. Identification of humoral immune responses

in protein microarrays using DNA microarray data analysis techniques. *Bioinformatics*, 22(14):1760–1766, 2006.

[121] A. C. Tan and D. Gibert. Ensemble machine learning on gene expression data for cancer classification. *Applied Bioinformatics*, 2(3):s75–s83, 2003.

[122] C. Toft and M. A. Fares. GRAST: a new way of genome reduction analysis using comparative genomics. *Bioinformatics*, 22(13):1551–1561, 2006.

[123] T.R.Golub, D.K.Slonim, P. Tamayo, and et.al. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

[124] L. V. Veer, H. Dai, M. V. de Vijver, and et.al. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415:530–536, 2002.

[125] M. J. V. D. Vijver, Y. D. He, L. J. V. T. Veer, and et. al. A gene-expression signature as a predictor of survival in breast cancer. *The New England Journal of Medicine*, 347:1999–2009, 2002.

[126] H. Wang and H. Huang. SED, a normalization free method for DNA microarray data analysis. *BMC Bioinformatics*, 5:121, 2004.

[127] H. Wang, W. Wang, J. Yang, and P. S. Yu. Clustering by pattern similarity in large data sets. In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, pages 394–405, 2002.

[128] Y. Wang, J. Lu, R. Lee, Z. Gu, and R. Clarke. Iterative normalization of cDNA microarray data. *IEEE Transactions on Information Technology in Biomedicine*, 6(1):29–37, 2002.

[129] Y. Wang, R. Zheng, and Y.-J. Qiao. Modeling, learning and simulating biological cells with entity grammar. In Y. Shi, G. D. van Albada, J. Dongarra, and P. M. A. Sloot, editors, *ICCS (4)*, volume 4490 of *Lecture Notes in Computer Science*, pages 138–141. Springer, 2007.

[130] Z. Wang and V. Palade. A comprehensive fuzzy-based framework for cancer microarray data gene expression analysis. In *BIBE*, pages 1003–1010. IEEE, 2007.

[131] G. I. Webb and Z. Zheng. Multistrategy ensemble learning: Reducing error by combining ensemble learning techniques. *IEEE Trans. Knowl. Data Eng*, 16(8):980–991, 2004.

[132] L. Wei, Y. Liu, I. Dubchak, J. Shon, and J. Park. Comparative genomics approaches to study organism similarities and differences. *Journal of Biomedical Informatics*, 35(2):142–150, 2002.

[133] X. wen Chen. Gene selection for cancer classification using bootstrapped genetic algorithms and support vector machines. In *CSB*, pages 504–505. IEEE Computer Society, 2003.

[134] M. West and C. B. et. al. Predicting the clinical status of human breast cancer by using gene expression profiles. *Proc Natl Acad Sci U S A*, 98:11462–11467, 2001.

[135] I. H. Witten and E. Frank. *Data Mining - Practical machine learning tools and techniques with java implementations.* Morgan Kaufmann, San Mateo, California, 2000.

[136] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg. Top 10 algorithms in data mining. *Knowl. Inf. Syst*, 14(1):1–37, 2008.

[137] Y. Xu, D. Xu, and E. C. Uberbacher. A new method for modeling and solving the protein fold recognition problem (extended abstract). In *RECOMB*, pages 285–292, 1998.

[138] C. Yeang, S. Ramaswamy, P. Tamayo, and et.al. Molecular classification of multiple tumor types. *Bioinformatics*, 17(Suppl 1):316–322, 2001.

[139] K. Yendrapalli, R. B. Basnet, S. Mukkamala, and A. H. Sung. Gene selection for tumor classification using microarray gene expression data. In S. I. Ao, L. Gelman, D. W. L. Hukins, A. Hunter, and A. M. Korsunsky, editors, *World Congress on Engineering*, Lecture Notes in Engineering and Computer Science, pages 290–295. Newswood Limited, 2007.

[140] X. Yin and J. Han. CPAR: Classification based on predictive association rules. In *Proceedings of IEEE International Conference on Data Mining*, pages 369–376, 2001.

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

[142] H. Zhang, C.-Y. Yu, and B. Singer. Cell and tumor classification using gene expression data: Construction of forests. *Proceeding of the National Academy of Sciences*, 100(7):4168–4172, April 1 2003.

[143] J. Zhang, M. Aizawa, S. Amari, Y. Iwasawa, T. Nakano, and K. Nakata. Development of kibank, a database supporting structure-based drug design. *Computational Biology and Chemistry*, 28(5-6):401–407, 2004.

[144] W. Zhang, R. Rekaya, and K. Bertrand. A method for predicting disease subtypes in presence of misclassification among training samples using gene expression: application to human breast cancer. *Bioinformatics*, 22(3):317–325, 2006.

[145] Z.-H. Zhou and Y. Jiang. Medical diagnosis with C4.5 rule preceded by artificial neural network ensemble. *IEEE Transactions on Information Technology in Biomedicine*, 7(1):37–42, 2003.

[146] Z. Zhu, Y.-S. Ong, and M. Dash. Markov blanket-embedded genetic algorithm for gene selection. *Pattern Recognition*, 40(11):3236–3248, 2007.