Neuro-Fuzzy Modeling for Microarray Cancer Gene Expression Data

Zhenyu Wang

Linacre College

Oxford University Computing Laboratory

First Year Transfer Report

PRS in Applications and Algorithms Group

October, 2005

Abstract

In recent years, DNA microarray technology has proven to be a very powerful tool for simultaneously monitoring the expression of several thousands of genes. Analysing the large amount of gene expression data from microarray chips can play a very important role in biology medicine, especially in cancer diagnosis. It also offers an opportunity and a challenge for current machine learning research. Clustering analysis and some other statistical methods have been established as primary tools for the analysis of microarray data. Some other supervised learning approaches, such as Support Vector Machines (SVM), Multi-Layer Perceptions (MLP) and Decision Trees (DT) are also widely used in practice. In this study, we attempt to use a different tool, Neuro-Fuzzy models (NF). For this aim, a NF model called Adaptive-Network-Based Fuzzy Inference System (ANFIS) has been applied for the first time to microarray data analysis. Then, we propose a Neuro-Fuzzy Ensemble model (NFE), in order to reduce the computation cost and achieve better generalization ability. Two proposed approaches were tested on three benchmark cancer gene expression data sets. The experimental results show that our NF and NFE models can be used as efficient computational tools for microarray data analysis. In addition, compared to some current most widely used approaches, NF/NFE models not only give a good classification result, but their behavior can also be explained and interpreted in human understandable terms, which can provide the researchers with a better understanding of the data. The report will also briefly review related current work and intended future research directions.

Keywords:

Machine learning, cancer classification, microarray analysis, gene selection, adaptive-network-based fuzzy inference system, neuro-fuzzy ensemble, leave one out cross validation, leukemia cancer, colon cancer, lymphoma cancer, noisy data, missing values

Publications:

1. Z. Wang, X Yao, and Y. Xu, "An Improved Constructive Neural Network Ensemble Approach to Medical Diagnoses", Proc. of the Fifth International Conference on Intelligent Data Engineering and Automated Learning (IDEAL04), Lecture Notes in Computer Science, Springer, Vol. 3177, pp.572-577, August 2004. (Published)

2. Z. Wang, V. Palade, and Y. Xu, "Neuro-Fuzzy Ensemble Approach for Microarray Cancer Gene Expression Data Analysis", Proc. of the Second International Symposium on Evolving Fuzzy System (EFS06), IEEE Computational Intelligence Society, September 2006, Lancaster, UK. (Submitted)

3. Z. Wang, X. Yao, Y. Xu, and V. Palade "A Smart Evolutionary Programming Method for Function Optimizations", Proc. of Tenth International Conference on Knowledge-Based, Intelligent Information, Engineering Systems (KES06), IEEE Computational Intelligence Society, October 2006, Bournemouth, UK. (To be Submitted)

Acknowledgements

Firstly, I would like very much to thank my supervisor, Dr. Vasile Palade, for his guidance and support throughout my first year in Oxford University. I have learned a lot from him, including how to choose and pursue an interesting research topic.

Secondly, I would like to thank Professor Xin Yao and Dr. Yong Xu, in University of Birmingham, for their great help in year 2003-2004, during my first year of staying aboard. Most importantly, many thanks for their time to review this report and give very important suggestions for both of the report and my future research.

I would also like to thank my colleagues and friends: Gaofeng Hang, Romesh Ranawana, Douglas Vieira in Applications and Algorithms Group, Computing Laboratory, Oxford University, and Lizhong Hao, Jian Chen in Wellcome Trust Centre for Human Genetics (WTCHG) in Clinical Medicine Department, Oxford University for their warmly discussions and suggestions.

Last but not least, thanks to my parents for all their support.

Contents

List of Figures

Chapter 1

Introduction

Even though the complete sequence of the human genome has been published by both international publicly funded Human Genome Project (The-Genome-International-Sequencing-Consortium (2001)) and a private corporation, Celera Genomics (Venter (2001)) on February 11th, 2001, the analytical work has just begun. Many new methods have been designed for discovering the mystery behind the gene data. One of the most revolutionary techniques is called high-density DNA microarray chips, or commonly called DNA chips or gene chips. This technique can measure the activities of thousands of genes simultaneously under different experimental environments and conditions. It allows us to have a "global" view of the cell for first time (Gregory & Pablo (2003)).

The gene expression profiles from particular microarray experiments have been recently used for cancer classification (Alon *et al.* (1999); Golub *et al.* (2002); Slonim *et al.* (2000)). This approach promises to give a better therapeutic measurement to cancer patients by diagnosing cancer types with improved accuracy (Slonim *et al.* (2000)). However, the amount of data produced by this new technology is more than one can manually analyse. Hence, the need to automatically analyse of the microarray data offers an opportunity for Machine Learning (ML) methods to have a significant impact on cancer research (Molla *et al.* (2004)). Machine learning approaches are suitable for microarray gene expression data due to the learning ability to construct classifier/hypotheses that can explain complex relationships within the data (Tan & Gilbert (2003)). Generally, there are two types of learning schemes in machine learning (Mitchell (1997)):

- Supervised learning, where the output has been given and labeled a priori, or the learner has some prior knowledge of the data;
- Unsupervised learning, where no prior information is given to the learner regarding the input data or the output.

Unsupervised methods, such as Clustering (Baumgartner *et al.* (1998)), Self-Organizing Maps (SOM)(Kohonen (1997)), and others, are originally used to analyse the relationships among different genes. Recently, supervised methods, such as Support Vector Machines (SVM) (Brown *et al.* (2000); Furey *et al.* (2000)), Multi-Layer Perceptrons (MLP or NN) (Khan *et al.* (2001); Xu *et al.* (2002)), K Nearest Neighbors (KNN)(Li *et al.* (2001)), Bayesian Network (BN) (Hwang *et al.* (2002)) and Decision Trees (DT) (Peng & Flach (2001)) have been successfully applied to classify different tissues. In this research, we mainly focus on using the supervised learning approaches to analyse microarray cancer gene expression data. Some of the most popular supervised microarray analysis methods will be discussed in details in Section 4.3.

A challenge in predicting the diagnostic categories by using microarray data is that the number of genes is usually much greater than the number of tissue samples available. Common approaches are to select a subset of most useful genes, then classify the different samples as cancer or non-cancer, according to the selected genes by using certain classifiers. The selection of relevant genes for classification process is known as gene selection. In this report, two most popular gene selection techniques, Information Gain (IG) and Signal to Noise Ratio (SNG) will be introduced and adopted. In addition, microarray data often brings in multiple missing gene expression values and noisy signals from the experiments, which usually degrade the performance of statistical and machine learning algorithms. Therefore, classifying cancer microarray gene expression data can be regarded as a high-dimensional-low-sample data set with lots of noisy/missing data. Most of current methods in microarray data analysis are generally lacking robustness with respect to noisy and missing data. Meanwhile, they can not completely bring out the hidden information in the data (Prabakaran *et al.* (2005)). Fuzzy set, as a novel way of characterizing non-probabilistic uncertainties, was firstly published by Zadeh (Zadeh (1965)). Unlike general crisp set, fuzzy set provides

us a relatively new concept for representing uncertain and imprecision in data (Ohno-Machado *et al.* (2002)). Fuzzy Inference Systems (FIS) are built based on this theory. Different from some black-box methods, fuzzy rule-based model can not only provide simple classification results, but also easily be explained and interpreted by human understandable fuzzy rules. This can provide the researchers an insight of the models. Furthermore, FISs adapt numerical data (input/output pairs) into human linguistic terms, which offer very good capabilities to deal with noise and missing data.

Unfortunately, how to define the rules and membership functions also requires a lot of prior knowledge. Sometimes it seems to be impossible to obtain them in practice, e.g., especially in the case of large amount of gene expression data. Meanwhile, FISs or some other rule-based systems have well-known limitations in dealing with high dimensional data. Although some fuzzy-based applications for microarray analysis have already been presented (Jiang & Gruenwald (2005); Ressom *et al.* (2003); Guthke *et al.* (2002)), all those FISs are small models, and only performing well on simple data sets. Because large rule-based models require huge computational cost and long time, sometimes are unacceptable in practice. To sum up, fuzzy-based methods have not attracted enough attentions from researchers and have not being effectively applied to most of cancer gene expression data problems.

But some recent developments in machine learning area provide us with some good ways to resolve these conflicts. In this research, we attempt to construct hybrid Neuro-Fuzzy models (NF) which combine the learning ability of neural systems and the interpreting ability of fuzzy systems, so that classifiers can automatically generate and adjust the membership functions and linguistic rules from the given microarray gene expression analysis data. We successfully applied a well known NF model, Adaptive-Network-based Fuzzy Inference System (ANFIS) (Jang (1992); Jang & Sun (1995); Jang & Sun (1997)) to this problem for the first time, which have not been published in any other's work, according to our knowledge. In order to improve the inherent weakness of ANFIS model, another NF model, Neuro-Fuzzy Ensemble model (NFE) has also been developed. Two proposed approaches are tested on three benchmark microarray cancer data sets, including leukemia cancer data set, colon cancer data set, and lymphoma

cancer data set. During the experiments, we found that the traditional training and test strategies can not perform well, because a small number of available data usually can not sufficiently represent the whole search space. As suggested by many other researchers (Khan *et al.* (2001); Ding & Peng (2003)), we adopt Leave One Out Cross Validation strategy (LOOCV) to evaluate models which will be introduced later in Section 5.3. Experimental results show that our NF models can be efficient computational tools for microarray data analysis, by not only giving classification results comparable to previous approaches, but also providing an interaction between the problems and solutions. More motivations about this research will be discussed in Section 1.1.

1.1 Research Motivations

1. Neuro-Fuzzy approach is necessary. There are over 200 different types of cancer, each of which has a unique set of clinical characteristics, a specific treatment regime and a different chance of being cured (Khan *et al.* (2001)). Unfortunately, it is sometimes difficult for even the experienced specialists to tell the difference among particular cancer and their subtypes. The DNA microarray technology provides a much more robust diagnosis than traditional approaches. However, the thirst for a good computational analysis tool of the microarray data is still unquenched. Current methods only provide simple classification results, which is not as reliable as to be applied by clinicians. Neuro-fuzzy approaches combine the advantages of both NNs and FISs, and have been emerged as a successful practical technology in many areas, including control field (Jang & Sun (1995)), signal and image processing (Al-Jarrah & Halawani (2001)), fault diagnosis of industrial equipments (Palade *et al.* (2002)), medical diagnosis (Carmona *et al.* (2001)) and financial forecasting (Ringhut & Kooths (2003)). In most application areas, NF systems can achieve a higher accuracy within a relatively shorter training time (comparing with NNs). Unlike other approaches, NF models are more transparent models. Their behavior can be explained in human understandable terms, such as linguistic terms and linguistic rules. This provides us with a better understanding of the data and gives the researchers and clinicians a clear explanation how the diagnose results are given . Meanwhile, NF models can also

easily incorporate prior knowledge, which helps obtaining more refined models. In addition, NF methods offer good capabilities to deal with high noisy/missing data. All these advantages show that NF models can be very strong and effective tools in microarray data analysis study. The first motivation comes from the question on how we can apply a NF model on microarray gene expression data, and whether our NF models still generate a good classification result on this kind of problems.

2. What is the optimal structure of neuro-fuzzy models for microarray gene expression data classification? The first question has been answered by our current research work. A classic NF model called ANFIS model and its ensemble structure have been successfully applied to cancer microarray analysis. The experimental results show that our models perform well on most tested data sets, see Chapter 7. However, ANFIS is not the only available solution for this problem. There are some other different neuro-fuzzy combination systems in the literature. Every type of neuro-fuzzy system has particular computational properties that make them suitable for certain kind of particular problems but not for others. In other words, their advantages and disadvantages are complementary. The second research motivation is to study different NF combination models, by combining the strong-points of different models, in order to obtain a better structure of neuro-fuzzy models for microarray gene expression data classification

3. How to select informative genes? The third motivation comes from the features of microarray data. In microarray data set, the number of samples is usually less than 100 due to the cost of the microarray experiments, (e.g., there are only 74 samples in leukemia cancer data set (Golub *et al.* (2002)), 62 samples in colon data set(Alon *et al.* (1999)), 47 samples in lymphoma cancer data (Slonim *et al.* (2000)), etc.), but the number of genes in each sample is around 10^3-10^4. Using all genes to train classifiers requires huge computational cost for training the classifiers, especially for rule based systems. Meanwhile, too many genes may cause the models easily to be over-fitted. Common approaches for such tasks are to select the smallest number of genes with most information before we train the classifiers. But for different data sets, the optimal number of selected genes varies. A good classification system is normally required to obtain a higher classification accuracy with less number of selected genes. How to adopt or construct more

effective gene selection strategy for our models is also a big challenge for our research. More details will be discussed in Chapter 3.

4. How to reduce the number of rules? During the experimental studies, we found that some data requires a large gene subset to present its property. This may cause too many rules than our model or machine can offer. The fourth research motivation is how to reduce the number of rules if the minimal number of gene is relatively large. Reducing the number of rules can also reduce the computational cost and training time, meanwhile discovery the most important knowledge from the data. Several machine learning techniques can be considered for solving this problem, e.g., ensemble learning (Hansen & Sakamon (1990)), hierarchical approaches (Berenji *et al.* (1991)), genetic algorithms (Bishop (1995), and pruning algorithms (Fürnkranz (1997); Reed (1993)).

5. Ensemble learning is useful. Targeting the above problems, we construct another NF model, Neuro-Fuzzy Ensemble (NFE) model. The advantage of this new model is obvious. By applying ensemble learning, the number of fuzzy rules in the model can be significantly reduced. It allows the model to study more features when the optimal gene subset is large. Better classification performance can be obtained according to the capability of ensemble learning at the same time.

Ensemble learning is a very popular and important current issue in current machine learning area. It can significantly improve the classification performance, especially in less sample data case. Many different types of ensemble classifiers have been published, for example, Neural Network Ensemble (NNE) (Kuncheva & Whitaker (2003)), Support Vectors Machine Ensemble (SVME) (Kim *et al.* (2002)), Bayesian Network Ensemble (BNE) (Valpola & Karhunen (2002)), etc. But NFE is a relatively new topic. Constructing good NFE models and studying the behavior of NFEs are also very attractive research topics.

6. How to deal with noisy data and missing values in cancer gene expression data set by using NF approaches, and how good is our NF models. As we previously mentioned, microarray gene expression data is highly noisy data, while missing data is very common in microarray experiments. Traditional approaches can range from simple steps, such as filling the spot of the missing data with a zero or row (gene) averaging, or using sophisticated interpolation techniques to

fill in the missing data spots. None of them can promise to perform well when the noise rate and missing rate are very high. But NF systems have strong capabilities on dealing with the noise data and missing rate, which are very suitable for microarray gene expression data. This can be looked as an important future direction.

7. How to extract knowledge from NF model? The next motivation comes from the interpretability of NF models. Firstly, a NF system is a rule based fuzzy system. It can summarize the knowledge from data, it also can be interpreted easily by using human linguistic rules. These features may help the biology researchers to understand and summarize the knowledge behind gene expression data. Combined with the learning ability of NNs, NF models can automatically generate and adjust the linguistic rules from unknown data. This seems very attractive in that it may give a brief outlook about the problem to biologists before they even start to analyse gene-expression data. Meanwhile, some basic and well-known prior knowledge from biological area may improve the learning abilities of the models.

To sum up, the aim of this project is to develop a set of neuro-fuzzy based bioinformatics tools which could be used by biomedical researchers to facilitate the analysis of microarray gene expression data. The short term (first two year) tasks include creating a simple and effective NF structure; analyse the behavior of our model; comparing our model with other approaches, especially in the highly noisy data and missing data case. The long term purpose is to construct models that can not just give good classification results, but also discover useful knowledge from the data for further cancer research. A software package of Neuro-Fuzzy approaches for cancer microarray analysis, named GeneNeFu, is also considered to be developed .

1.2 Report Outline

In Chapter 2, we shortly introduce some basic notions and important principles of NNs, FISs and NF systems. Discussing the advantages and disadvantages of NNs and FISs, and explaining why Neuro-Fuzzy combination is necessary. Some recent developments in NF area are also represented.

In Chapter 3, we will introduce some basic notions of microarray technology and microarray gene expression data. Some problems in current microarray data usage are pointed out. How to use microarray gene expression data for cancer classification and cancer prediction will also be described.

In Chapter 4, we formalize this problem as a high-dimensional-low-sample data set with lots of noisy/missing values classification problem in machine learning. We also summarize current research work in this area in order to compare with their work later. Two efficient genes selection methods, Information Gain (IG), Signal to Noise Ratio (SNR), will be introduced. Both of them will be adopted and compared in our study. Two special strategies of IG (IG2 and IG3), are discussed and given as a case study. Next, we introduce some most popular classifiers in microarray gene expression data application, including MLPs, SVMs and KNNs.

In Chapter 5, we introduce a well known NF model to cancer microarray gene expression data classification problem, called Adaptive-Network-based Fuzzy Inference System (ANFIS). The advantages and weaknesses of this approach are discussed. In order to obtain a better comparison with other's work, a training strategy, called leave one out cross validation (LOOCV), will be introduced.

Chapter 6 starts with introducing the notion of ensemble learning, and explaining why ensemble learning is better than individual models. Then we construct a novel Neuro Fuzzy Ensemble Model (NFE), followed by discussing the advantages/disadvantage of NFE by comparing with single NF model.

In Chapter 7, we test our NF and NFE models on three benchmark microarray gene expression data sets, including Colon Cancer Data Set, Leukemia Cancer Data Set and Lymphoma Cancer Data Set. Top 20 ranked genes of each data set selected by IG2 and SNR are presented. Comparison and analysis work among NF, NFE, and some other's methods are also given in this chapter.

Conclusions, future direction, and research plan are described in the last two chapters.

Appendix A gives top the 50 ranked genes by using IG method, for future usage. Appendix B summarizes the lectures, meetings, seminars I have attended from 09/2004 to 09/2005.

Chapter 2

Neuro-Fuzzy Systems

Artificial Neural Networks (ANNs/NNs) (Haykin (1994); Hertz *et al.* (1991); Bishop (1995); Nguyen *et al.* (1997)) and Fuzzy Inference Systems (FISs) (Zadeh (1965); Zadeh (1968); Zadeh (1985); Kandel (1987)) are two very important research areas in today's Artificial Intelligence (AI) studies. The success of FISs is due to the fact that fuzzy "IF-THEN" rules are well-suited for capturing the imprecise nature of human knowledge and reason processes. On the other hand, NNs tackle the same problems with a different strategy; they are equipped with a remarkable learning capability such that a desired input-output mapping can be discovered through learning by examples. Combination between artificial neural network and fuzzy system is a very important research topic, called Neuro-Fuzzy combination system (NF) (Nauck *et al.* (1997); Nauck (1997); Jang & Sun (1997)). In this chapter, we will introduce the basic notions, operations, and structures of NNs (Section 2.2), FISs (Section 2.1)and NFs (Section 2.3) in short. The pros and cons of these schemes will also be listed and compared.

2.1 Fuzzy Systems

System modeling based on conventional mathematical tools (e.g., differential equations) is not well suited for dealing with ill-defined and uncertain systems. By contrast, a fuzzy based system employing fuzzy if-then rules can model the qualitative aspects of human knowledge and reasoning processes without employing precise quantitative analysis. In this section, we will introduce some basic knowledge about fuzzy sets, fuzzy rules, and fuzzy inference systems.

2.1.1 Fuzzy Sets

In classical mathematics, a classical set is a set with crisp boundary. For example, let X be a certain universe of discourse, where

$$X = \{x_1, x_2, ...x_n | x \in X\}, \tag{2.1}$$

and its elements x denote all the possible weight values (kg) of an adult male human, $x \in X$. A classical crisp set C_{Thin} of X is defined as a function Φ called characteristic function of C_{Thin}, as:

$$\Phi : X \to \{0, 1\}, \tag{2.2}$$

for any element x of universe X, the characteristic function Φ is equal to 1 if x is an element of set A, and is equal to 0 if x is not an element of A. The same as $Thin$, another two similar crisp sets $C_{Average}$ and C_{Fat} can be defined. The characteristic functions of these crisp sets are depicted in Figure 2.1(left).

$$\Phi_{C_{Thin}}(x) = \begin{cases} 1, & \text{if } less - than - 50(x), \\ 0, & \text{otherwise.} \end{cases} \tag{2.3}$$

$$\Phi_{C_{Average}}(x) = \begin{cases} 1, & \text{if } between - 50 - and - 100(x), \\ 0, & \text{otherwise.} \end{cases} \tag{2.4}$$

$$\Phi_{C_{Fat}}(x) = \begin{cases} 1, & \text{if } larger - than - 100(x), \\ 0, & \text{otherwise.} \end{cases} \tag{2.5}$$

One problem arises if we have to define a linguistic term "fat". The use of any crisp set above results in a stiff situation, when a person of $100kg$ is considered to be a "fat man", but a $99.9kg$-person is said to be "not fat". In contrast to a classical set above, a fuzzy set is a set with fuzzy boundaries. The membership function of a fuzzy set is allowed to have values between 0 and 1, and it expresses the degree in which an element belongs to a given fuzzy set. This transition makes fuzzy sets more flexible and intelligent for the interaction between human and machine. Fuzzy sets were first introduced by Zadeh in his famous paper Zadeh (1965). As Zadeh mentioned, "fuzzy sets can play a very important role in human thinking, particularly in the domains of pattern recognition, communication of

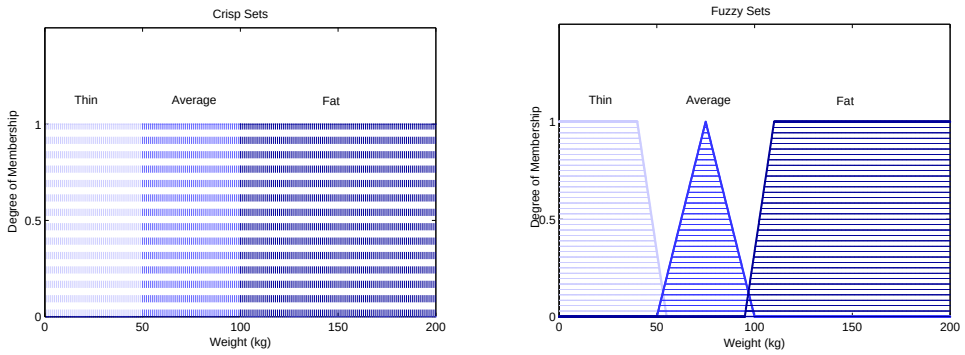

Figure 2.1: Typical crisp sets (left) and typical fuzzy sets (right) characterizing the human weight value (left), and typical fuzzy sets characterizing the human weight value (right)

information, and abstraction" (Zadeh (1965)). Using the same example as above, new fuzzy sets F_{Thin}, $F_{Average}$ and F_{Fat} of X can be defined as:

$$F_{Thin} = \{(x, \eta_{F_{Thin}}(x)), x \in X\}, \tag{2.6}$$

$$F_{Average} = \left\{\left(x, \eta_{F_{Average}}(x)\right) x \in X\right\}, \tag{2.7}$$

$$F_{Fat} = \{(x, \eta_{F_{Fat}}(x)), x \in X\}, \tag{2.8}$$

where η_F is called the membership function (MF), and it gives the degree to which x is an element of set F. This degree, a value between 0 and 1, denotes the degree of membership, also called membership value, as shown in Figure 2.1 (right). Triangular, trapezoidal, gaussian,or bell-shaped functions (See Figure 2.1 (right)) are most frequently used functions in the application areas. When the membership function takes only two values 0 and 1, F is identical to a crisp set which is defined by a characteristic function. In this instance, crisp sets can be looked as special cases of fuzzy sets. Several operators, e.g., "MAX", "MIN" and "COMPLEMENT", can be defined on fuzzy sets in a similar manner as "AND", "OR", "COMPLEMENT" in classical logic sets (Jang (1992)).

2.1.2 Fuzzy Rules

Fuzzy rules, or Fuzzy if-then rules, are defined as a conditional statement in the form "IF A THEN B". The IF-part of the rule is called the premise or antecedent, while the THEN-part of the rule is called the conclusion or consequence. There are two basic forms of fuzzy rules that have been developed to date: Mamdani's fuzzy rules (Mamdani & Assilina (1975); Jang (1992)) and Takagi-Sugeno's (TSK) fuzzy rules (Sugeno & Kang (1988); Takagi & Sugeno (1985)). The differences between these two types of fuzzy rules appear in the consequence part.

- In Mamdani fuzzy rules case, both A and B are described by linguistic variables. An example of such kind of rules is this:

  IF the distance is far, THEN the price of ticket is expensive. (2.9)

- In Takagi-Sugeno's fuzzy rules case, fuzzy sets are only used in the antecedent part, the consequence part is described by a non-fuzzy equation of the input variables. An example is shown in the followings:

  IF the distance is far, THEN the price of ticket $= \alpha_{far} \times \{distance/\beta_{far}\}^2$. (2.10)

Both types of fuzzy rules are widely used in system modeling and control areas. Fuzzy inference systems can be divided into two types according to which forms of fuzzy rules employ, called Mamdani-type FIS and Takagi-Sugeno-type FIS. We will give a more detailed description on these in Section 2.1.3.

2.1.3 Fuzzy Inference Systems

Fuzzy inference systems have been successfully applied in many fields, such as automatic control (Hegyi *et al.* (2000)), pattern classification (Berkan & Trubatch (1997)), fault diagnosis (Schmidt (1989)), etc. Fuzzy inference systems have different structures and different names associated with their application areas. They are also known as fuzzy-rule-based systems, fuzzy expert systems, fuzzy associative memory, or fuzzy controllers when it is used in control areas (Jang & Sun (1995)). But all fuzzy inference systems can be divided into four basic functional blocks, as shown in Figure 2.5:

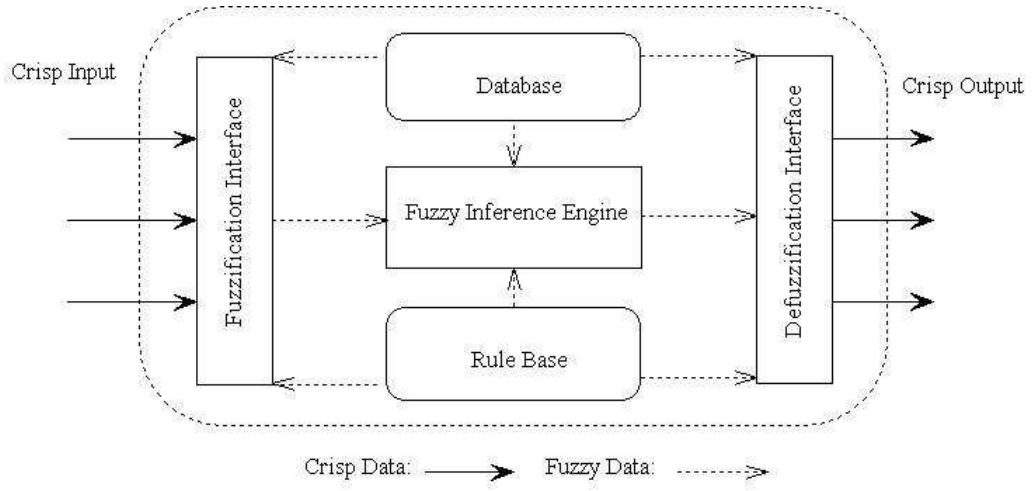

Figure 2.2: The basic structure of a fuzzy inference system.

- A fuzzification interface which transfers crisp inputs to fuzzy sets according to pre-defined membership functions;
- A database block and rule base which defines a number of membership functions and fuzzy rules;
- A fuzzy inference engine block that applies a fuzzy reasoning mechanism to obtain a fuzzy output;
- The fuzzy output is translated back to crisp value a via defuzzification interface.

Many different types of FISs have been proposed in the past years. All fuzzy inference systems perform the following four steps:

- *Step 1. Fuzzification:* Determine the degree with which crisp inputs belong to each of the defined fuzzy sets;
- *Step 2. Rule evaluation:* Generate the weight of each rule by using related operators between different membership values;

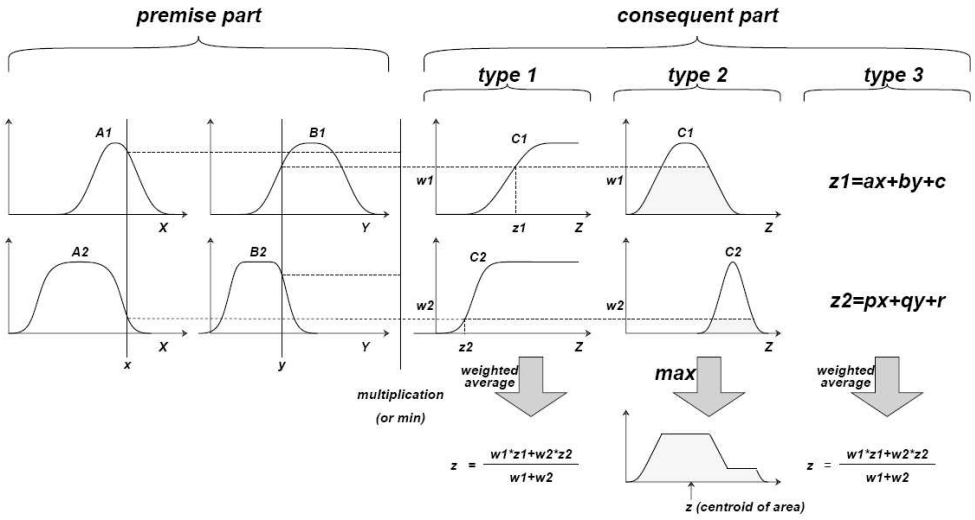

Figure 2.3: Different types of fuzzy inference systems (Jang & Sun (1995))

- *Step 3. Rule aggregation:* Combine all weights of rules into a single output. If the output is a fuzzy set, go to *Step 4*, otherwise, output the final crisp value;

- *Step 4. Defuzzification:* Translate the fuzzy output into a single crisp number, because the final output of a fuzzy system has to be a crisp number.

As we described in Section 2.1.2, most FIS can be classified into two types: Mamdani's fuzzy inference method and Takagi-Sugeno-type fuzzy inference systems. In Figure 2.3, Mamdani-types fuzzy rules are used in the first two systems, Takagi-Sugeno-type fuzzy rules are used in the last system. It is easy to see that the major difference lies in the consequence part.

Normally, Mamdani's method has more widespread acceptance, are more easily for human to understand, while the advantage of Takagi-Sugeno's method is that it has better computationally efficiency, it works well with optimization and adaptive techniques, which makes it very attractive in control areas, particularly for dynamic non-linear systems (Jang & Gulley (1997); Jang & Sun (1995)). But which system is better for a certain problem is problem-dependent. Such

phenomenon is called "No Free Lunch Theorem" (NFL) (Wolpert & Macready (1997)).

2.2 Neural Systems

In this section, we will introduce the basic structure of NN and explain how it work in practice.

2.2.1 Neural Networks

NNs have emerged as a practical technology, which have been successfully applied in many fields: control field (Newton & Xu (1993)), speech recognition (Diaz-Verdejo *et al.* (1991)), medical diagnosis (Wang *et al.* (2004)), signal and image processing (Trussell & Vrhel (1999); Teschioni *et al.* (1999); Varoglu & Hacioglu (1999)), etc. The main advantages of NNs include self-adaptivity, self-organization, real time operation, etc. This model takes us a different approach to problem solving from that of conventional computers. More details can be found in the textbooks of Taylor (1993) and Bishop (1995).

A NN is made up of a set of artificial neurons which are called nodes, and they have connections called weight between them. The simplest architecture of artificial neural networks is single-layered network, also called Perceptron, where inputs connect directly to the outputs through a single layer of weights. The most commonly used form of NN is the Multi-layer Perceptron (MLP), see Figure 2.4. NNs offer a very powerful and very general framework for representing non-linear mapping from several input variables to several output variables (Bishop (1995)). If we look to a NN as a function approximater, a three-layered MLP can approximate any function with any accuracy (Hagan *et al.* (1996)). A more detailed introduction of MLPs will be given in Section 4.3.1

In a NN, each of these inputs are multiplied by a connection weight, these weights are represented by W_n. In the common case, these products are simply summed, (see Eq 2.11), and then fed through a transfer activation function to generate a result.

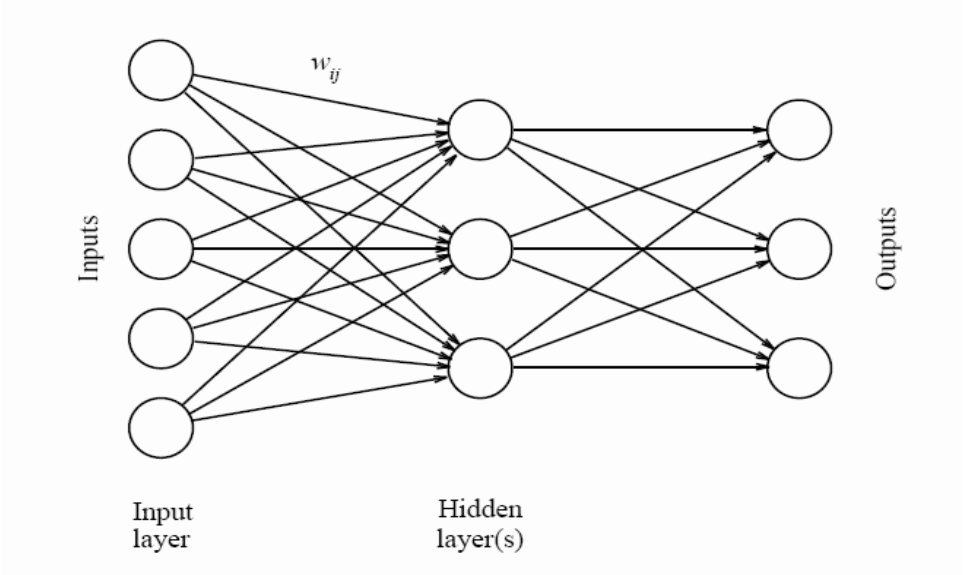

Figure 2.4: A standard three layers feed-forward NN.

$$Y_{output} = f(\sum_{n=0}^{N} X_n W_n). \tag{2.11}$$

The most frequently used activation function is logistic-sigmoid function, shown in Eq 2.12:

$$g(a) = \frac{1}{1 + exp(-a)}, \tag{2.12}$$

where the function generates outputs between 0 and 1 as the neuron's input goes from negative to positive infinity.

To obtain a trained network, we must adjust the weights of each unit in such a way that the error between the desired output and the actual output is reduced. There are many popular training algorithms in the literature, such as Genetic Algorithms (GA) (Bishop (1995); Hagan *et al.* (1996); Jain *et al.* (1996); Kitano (1990); David (1994)), Evolutionary Programming (EP) (Xin (1995); Yao & Liu (1997); Curran & O'Riordan (2002)), Particle Swarm Optimization (PSO) (Settles & Rylander (2002); Bergh & Engelbrecht (2000)), and their hybrids (Manfred & Yee (1998), etc.. The classic backpropagation algorithm (BP) is

the most popular supervised learning algorithms in practical application (Bishop (1995); Ripley (1998)). In general, BP algorithm is a gradient descent technique, the weight changes are made in proportion to the gradient of the error function. The error function E and weight-update rule are defined as shown:

$$E = \frac{1}{2}\sum_{o}^{n}(z_o - t_o)^2, \tag{2.13}$$

$$W_{ij}(t+1) = W_{ij}(t) - \eta\frac{\partial E}{\partial W_{ij}} + \beta(W_{oj}(t) - W_{ij}(t-1)) \tag{2.14}$$

where z_o is the real output from nodes of the final layer, t_o is the expected output, $W_{ij}(t)$ is the weight form node i to node j during the tth iteration, η and ∂ are two constants called the learning rate and the momentum rate.

But there are still some most common problems for BP algorithms: the NN must be trained in a fixed structure and the training time is relatively long. For this reason, we will apply a hybrid Backpropagation-Least Squares Estimate approach (BP-LSE) to speed up the training process of our NF model, which will be introduced in Chapter 5.

2.3 Hybrid Neuro-Fuzzy Systems

The basic idea behind neuro-fuzzy combination is to design a system that uses a fuzzy system to represent knowledge in an interpretable manner and have the learning ability derived from a NN to adjust its membership functions and parameters in order to enhance the system performance. The main drawbacks of both individual systems could be avoided, i.e., the black box behavior of NNs, and the problem of selecting suitable membership values for FISs. More details about this issue will be discussed in this section, together with the current research work in this area.

2.3.1 Neural Systems vs. Fuzzy Systems

Neural networks are trained automatically by certain learning algorithms until a satisfying output is acquired. Because the data of input and output are connected with the combination weights in the network, the concept or knowledge can not

Table 2.1: Comparison between neural networks and fuzzy inference systems

Neural Networks	Fuzzy Inference Systems
Prior knowledge can not be used	Prior knowledge can be incorporated
Learning from data	No learning algorithm
Black box	Interpretable
Slow	Quick
Difficult to extract knowledge	Easy to extract Knowledge

be clearly expressed. It has been difficult to explain how they reach their results. This is called "black box" phenomenon of NNs. Meanwhile, the learning process is relatively slow. Neither is it possible to extract knowledge (rules) from the trained NN, nor can we integrate some prior knowledge about the problem (normally in the form of "IF... THEN..." rules) into the NN in order to simplify the learning procedure.

In contrast, FISs are more favorable, in that their behaviors can be explained using fuzzy rules. FISs can easily be structured to include prior knowledge. Also FISs can easily be interpreted in human understandable terms during work. The structural knowledge can be expressed in linguistic terms and hence provide an understanding about the properties of the problem. But for building a FIS, we have to define the membership functions, fuzzy operators and the knowledge base. The problem of finding appropriate membership functions and fuzzy rules is often a tiring process of trial and error. It requires users to understand the data before training, which is usually difficult to achieve when the database is relatively large. A simple comparison between NNs and FISs is shown in Table 2.1.

To overcome these problems, a hybrid of NN and FIS can combine the advantages of two systems and avoid their disadvantages. This combination can constitute an interpretable model that is capable of learning, as NNs, and reasoning, as FISs (Kosko (1992)). On one hand, using this technique, it is possible to adjust the membership functions automatically from data by using NN learning algorithms. The trained membership functions can provide a better understanding about the properties of the unknown database. On the other hand, the new system can take advantage of the ability to embed a priori knowledge and constraints

into the system. This can have the effect of shorter learning times and better system performance. Therefore, hybrid systems of NNs and FISs are especially suitable for applications, where user interaction in model design or interpretation is desired.

2.3.2 Neuro-Fuzzy Systems

There are many different combinations between neural network system and fuzzy system. All the combinations can be divided into two types (Nauck (1997)):

- NNs equipped with fuzzy capabilities. NN is basic structure and FIS is the second. It is possible to fuzzify a NN by the extension principle in order to be able to process fuzzy inputs. Or fuzzy techniques are adopted to speed up the learning process. This system is usually called Fuzzy Neural Network (FNN), and either the network inputs/outputs or the weights are fuzzy sets.

- FISs augmented by neural networks. FIS is the basic structure and NN is the second. The NN is used to provide inputs for a fuzzy system, or to change the output of a fuzzy system. They can be seen as extensions of FISs by NNs, and are usually called Neuro-Fuzzy Systems (NF).

In this study, we mainly considered the second case, NF systems. Many different integrated neuro-fuzzy models have been published in literature. Some of the major works in this area are GARIC (Zhang & Kandel (1993)), FALCON (Lin & Lee (1991)), NEFCON (Nauck & Kruse (1997)), FUN (Sulzberger *et al.* (1993)), SONFIN (Feng & Teng (1998)), FINEST (Tano *et al.* (1996)), EFuNN (Kasabov & Song (1999)), EvoNF (Sulzberger *et al.* (1998)), and many others. Some of these architectures are shown in Figure 2.5. A neuro-fuzzy system can be viewed as a special three-layered feed-forward neural network. Some neuro-fuzzy models use more than three layers (Jang & Gulley (1997)). A well known five layered NF architectures, called ANFIS will be used in this work.

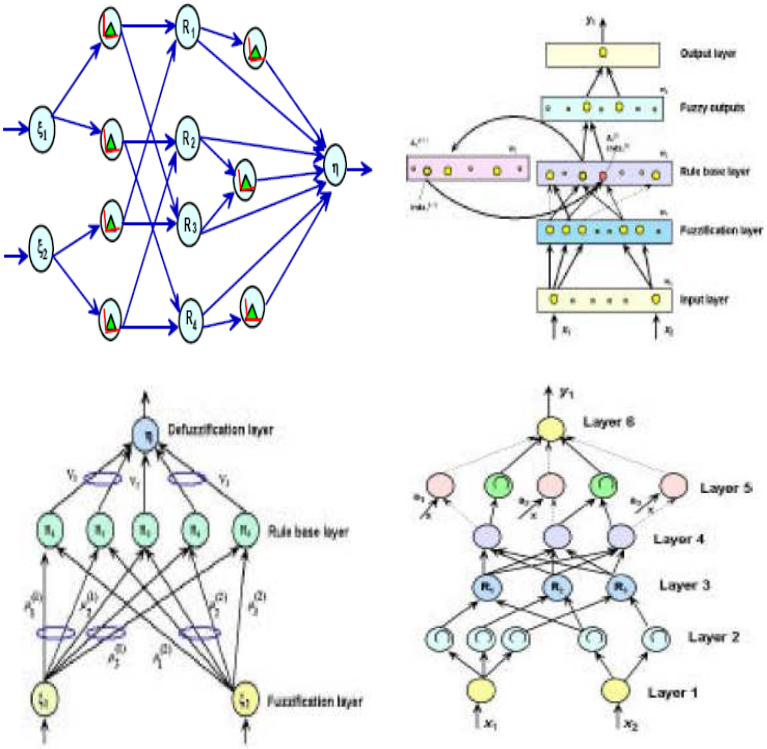

Figure 2.5: The basic structure of GARIC (top left), EFuNN (top right), NEFCON (bottom left), SONFIN (bottom right).

Chapter 3

Microarray Analysis

We begin this chapter by presenting a brief overview of DNA microarray gene expression technology. A short introduction of how to apply microarray technology for cancer classification is given in Section 3.2. Some current problems in microarray usage are discussed in Section 3.3.

3.1 Microarray Technology

Cells are the most basic units of all organisms on earth, except for viruses, e.g., yeast has only one cell, while any of mammals, including humans beings, have tons of cells. Inside a cell, there is a nucleus, and inside a nucleus, there are several separated long segments called chromosomes[1] which organized by deoxyribonucleic acid (DNA). The basic units of DNA are nucleotides which consist of sugar phosphate backbone and one of the four bases adenine(A), cytosine(C), guanine (G), and thymine (T), see Brändle *et al.* (2003). A pairs with T, while C pairs with G. DNA codes the hereditary information through particular order of these base pairs on a double-stranded helix for making future organisms (see Figure 3.1).

DNA has coding and non-coding segments, and the coding segments are called genes. The way from genes to proteins includes two steps: First, DNA is transcribed into messenger ribonucleic acid, via mRNA or RNA for shorter. Second, the mRNA is translated into proteins. Practically all cells in the same organism

[1]There are 46 chromosomes, viz 23 pairs, in a human cell, each parent contributes 23 chromosomes for their children.

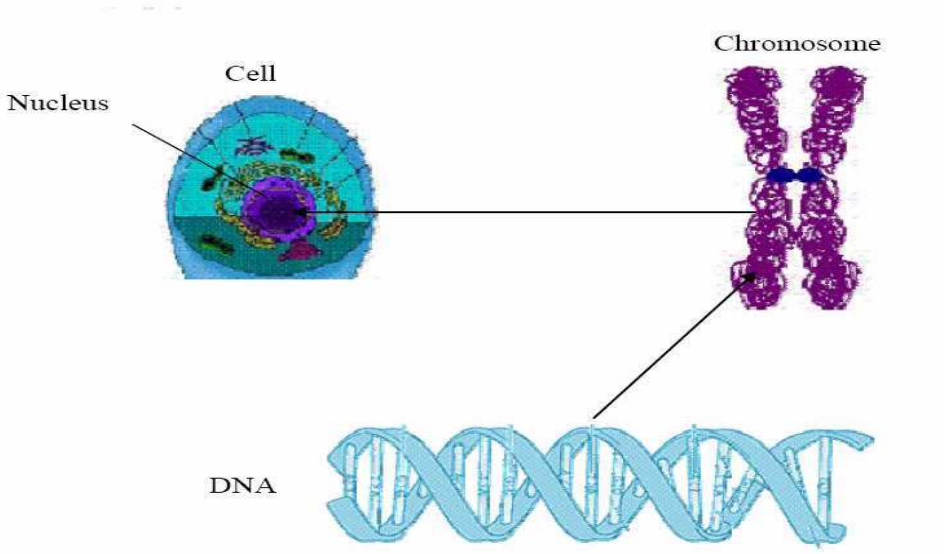

Figure 3.1: An overview of the relationship among cell, nucleus, chromosome, gene, and DNA.

have the same genes, but these genes can be expressed differently at different times and under different conditions. Most of the molecular biology research is focused on mRNA level due to the fact that all major differences in cell state or type are correlated with changes in the mRNA levels of many genes (Dettling (2004)). More information can be found in other molecular biology textbooks (Alberts *et al.* (2002), Dubitzky *et al.* (2002)) and papers (Gregory & Pablo (2003), Prabakaran *et al.* (2005)), etc.

Applying microarray experiments to study biological problems was firstly introduced by a Stanford University research team in 1995. Nowadays, it has become one of the strongest tools for biomedical research. Even though there are some other technologies in use, the microarray technology seems to be more promising than others (Gregory & Pablo (2003)). For example, Northern Hybridization is labor-intensive and it only aims at a single gene in each experiment (Sambrook *et al.* (1989)). Microarray techniques make it possible to simultaneously measure the expression of thousands of genes under different experimental environments and conditions. It enables us to analyze the gene information very rapidly and precisely by managing them at one time (Cho & Won (2003)).

Figure 3.2: Affymetrix Gene Chip (from GeneticLab Co. Ltd, Japan)

Affymetrix Gene Chips (see Figure 3.2) are currently the most widely used microarray products. There are two main types of microarray: oligonucleotide microarray, using representative gene segments, and cDNA microarray, using entire transcripts. Our study mainly focused on the gene expression data from the latter category. The applications of microarray analysis research include: identification of differently expressed genes, cellular classification, disease subtype identification, and inference of gene regulatory interactions (Prabakaran *et al.* (2005)).

The whole process of a microarray experiment is shown in Figure 3.3. First, design the microarray experiments according to a biological problem we want to study. Generally, microarray experiments can be divided into two types. One focuses on time series data which contains the gene expression data of various genes under various course of time of an experiment (Prabakaran *et al.* (2005)). Another type of microarray experiment consists of gene expression data of various genes taken from various tissue samples or under different experimental conditions, such as nutrition, temperature, or chemical environment (Dubitzky *et al.* (2002)). Different conditions can be used to answer the question "which genes are changed under this condition" (Paul & Kumar (2004)), while different tissues used in similar conditions are helpful in the classification of different types of

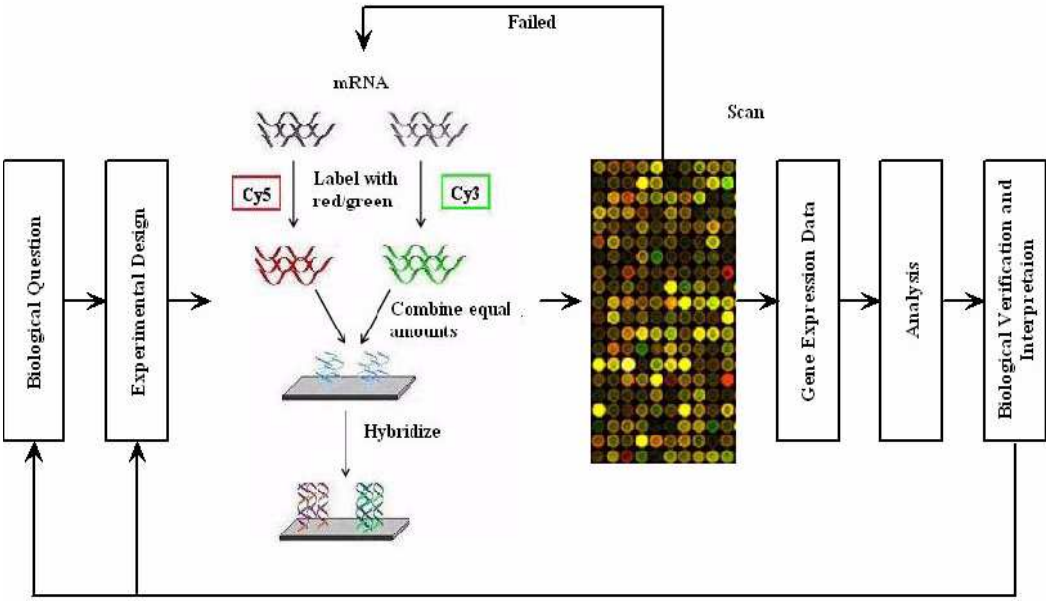

Figure 3.3: General process of acquiring the gene expression data from a typical DNA microarray experiment.

tissues.

3.2 Cancer Gene Expression Classification

The determination of cancer type and stage is often very important to the assignment of the appropriate treatment. It is known that mutations in genes can lead to cancer. Normal cells can evolve into malignant cancer cells through a series of mutations in genes that control the cell cycle, apoptosis, and genome integrity, to name only a few (Gregory & Pablo (2003)). These mutations are absent in normal cells, and we would expect the expression levels of these genes, and genes regulated by these genes, to be different in normal and cancerous cells. By learning these differences, it is now possible to classify cells as cancerous or normal by measuring the expression levels of various genes present in the cells.

For example, acute myeloid leukemia (AML) and acute lymphoblastoid leukemia (ALL) cells look very similar, but they respond to different therapies. A correct diagnosis is therefore essential for successful treatment. In the case of AML,

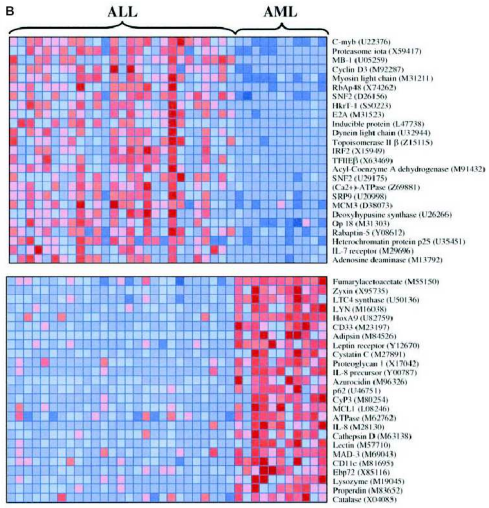

Figure 3.4: The top 50 ranked genes with the ALL-AML class distinction are shown. Each row corresponds to a gene, with the columns corresponding to expression levels in different samples. The top panel shows genes highly expressed in ALL, the bottom one shows genes more highly expressed in AML (Golub *et al.* (2002)).

drugs such as daunorubicin and cytarabine are favored, in the case of ALL, patients respond better to drugs such as vincristine and methotrexate (Gregory & Pablo (2003)). Traditionally, the correct diagnosis of leukemia subtypes (class predication) has relied on a combination of techniques. Smears or biopsies are studied by skilled clinicians to look for subtle differences in the cell shape (Golub *et al.* (2002)). However, none of these tests is 100 percent accurate.

A central goal of the analysis of gene expression data is the identification of sets of genes than can serve, via expression profiling assays, as classification diagnostic platforms. In the case of AML and ALL it was found that an accurate classification is possible by analyzing the expression of 50 genes on an array representing nearly 7000 genes in total. In Golub *et al.* (2002), 36 out of 38 patients were classified correctly using this single test. In only two cases was the diagnosis uncertain, see Figure 3.4.

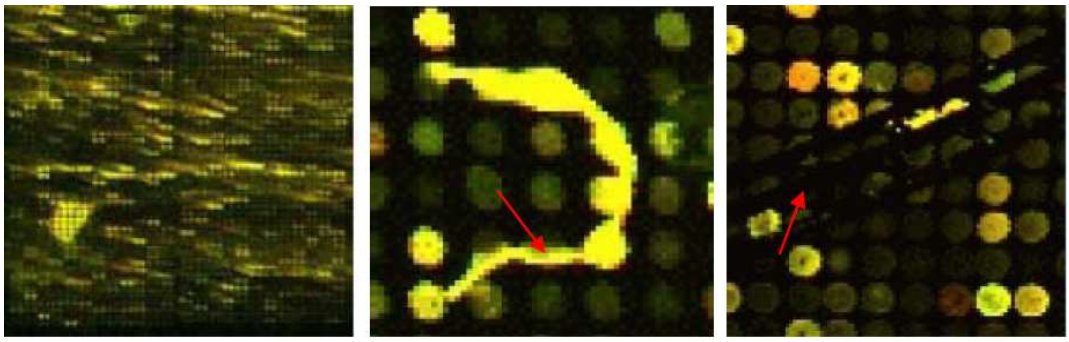

Figure 3.5: Experimental noisy and missing data are very common in microarray experiments. The left figure is an overview of a gene chip with high noise. The middle figure and right figure show how the experimental noisy and missing data happened in the experiments (marked by red narrows).

3.3 Some Problems in Microarray Data Usage

Widely using microarray techniques as a clinical diagnosis is still a big challenge for biologists and computer researchers. There are some practical problems which limits its usage:

1. Missing values in the spots - sometimes not all spots have values because there is no expression of the gene at that place, or because of improper scanning, see Figure 3.5 (right).

2. Error and noise is brought in, due to various accessories such as scanners, see Figure 3.5 (left and middle).

3. Microarray needs prior requirement to prepare samples by using other methods like Gel Electrophoresis. Quality of these samples has to be standardized before using them in microarray, since different samples of the same type may vary in quality (Gregory & Pablo (2003)).

4. A standard has to be developed and used to make the data obtained from various experiments comparable.

5. The data is only obtained at transcription level. For higher level application, such as inferring gene regulatory network, additional information resources should be integrated (Prabakaran *et al.* (2005)).

Chapter 4

Machine Learning and Microarray Analysis

In this chapter, we formalize cancer microarray gene expression classification problem into a high-dimensional-low-sample data set with a lot of noisy/missing values classification problem in machine learning. Some current work for this problem is summarized. Two efficient gene selection methods and three most widely used classifiers are introduced in Section 4.2 and Section 4.3.

4.1 Machine Learning in Microarray Analysis

In microarray experiments, different DNA samples are fixed to a glass microscope slide, each at a pre-defined position in the array, known as "gene chip". mRNAs isolated from different tissue or under different conditions are labeled with two different fluorochromes (generally the green Cy3 and the red Cy5), then hybridize them with the arrayed DNA probes on the slide (Step 3 in Figure 3.3). Using a fluorescent microscope and image analysis, the gene expression data (denoted as G) is measured by computing the log ratio between the two intensities of each dye.

$$G = log_2 \frac{Int(Cy5)}{Int(Cy3)} \tag{4.1}$$

where $Int(Cy5)$ is the intensity of red color, while $Int(Cy3)$ is the intensity of green color.

Table 4.1: A typical gene expression matrix ($m \times n$), where rows represent samples obtained under different experimental conditions and columns represent genes

	Gene 1	Gene 2	...	Gene m-1	Gene m	Class
Sample 1	165.1	276.4	...	636.6	784.9	1
Sample 2	653.6	1735.1	...	524.1	104.5	-1
...	...	...	...	...	...	...
Sample n-1	675.0	45.1	...	841.9	782.8	-1
Sample n	78.2	893.8	...	467.9	330.1	1

The data from a series of n such experiments can be represented as a $m \times n$ gene expression matrix, see Table 4.1. Each row represents a sample that consists of m genes from one experiment. Each sample belongs to a certain class (normal or tumor). In each data set, the researchers repeated the same experiment from n different volunteers, each line in this data set representing a volunteer.

From Table 4.1, we can see that classifying microarray gene expression data can be looked as a high-dimensional-low-sample problem. Common approaches are to select a subset of the most useful features, then classify the different samples as cancer or non-cancer, according to the selected features, by using certain classifiers. This can be summarized as follows. Let the given gene expression data set as:

$$\tilde{D} = \{(g_1, t_1,), ..., (g_n, t_n,)\}, \tag{4.2}$$

where an input vector $g_i = (g_1, ..., g_m)$ denotes a gene expression pattern, m is the number of genes in this pattern, t_i represents which class the pattern belongs to (see Section 4.2), and n denotes the number of patterns in the data set. Then, choose $\tilde{m}$ genes out of m according to certain algorithms. Select $\tilde{n}$ patterns with $\tilde{m}$ genes to train the classifier, and leave $n - \tilde{n}$ patterns (with $\tilde{m}$ genes) to test the performance of trained model. Figure 4.1 gives a typical cancer classification system.

Unsupervised methods do not use any tissue annotation (e.g., tumor vs normal) in the partitioning step. In contrast, supervised methods attempt to predict the classification of new tissues based on their gene expression profiles after training on examples that have been classified by an external "supervisor". In this

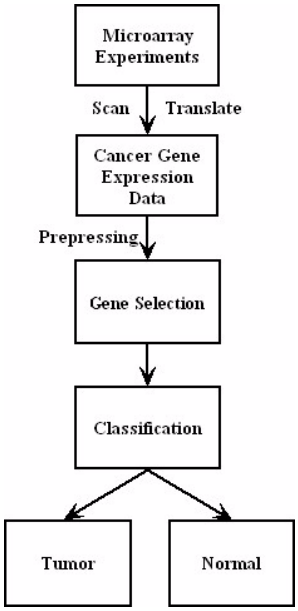

Figure 4.1: A typical cancer classification system.

research, we are interested in using supervised learning technologies to cancer diagnosis and cancer discovery. Related published models are shown in Table 4.2, and some of them will be discussed later in details (see Section 4.3).

4.2 Gene Selection

As we previously mentioned, the number of features (usually in the range of 2000-30000) is much bigger than the number of samples (usually in the range of 40-200). When such data is presented, many standard data analysis and machine learning techniques are either inappropriate or become computational infeasible (Rogers (2004)). Not all of the thousands of genes whose expression levels measured are needed for classification (Paul & Iba (2004)). Most genes are not related to the performance of the classification. Taking such genes into account during classification increases the dimension of the classification problem, poses computational difficulties, and introduces unnecessary noise in the process. A major goal for diagnostic research is to develop diagnostic procedures based on inexpensive microarrays that have enough probes to detect diseases. Thus, it is crucial

Table 4.2: Current work on cancer classification by using microarray gene expression data

Authors	Year	Dataset	Gene Selection	Classifiers
Ben-Dor *et al*	1999	Leukemia Cancer	TNoM score	KNN
		Colon Cancer	All genes	SVM
Furey *et al*	2000	Leukemia Cancer	SNR	SVM
		Colon Cancer		
Li *et al*	2001	Leukemia Cancer	GA	KNN
		Colon Cancer	GA	KNN
Khan *et al*	2001	Lymphoma Cancer	PCA	SVM
Yu *et al*	2004	Leukemia Cancer	RBF/all gene	DT
		Colon Cancer	RBF/all gene	DT
		Lung Cancer	RBF /all gene	DT
		Breast Cancer	RBF/all gene	DT

to recognize whether a small number of genes can suffice for good classification. This requires selection of some genes that are highly related to particular classes for classification, which are called informative genes. This process is called gene selection, or feature selection in machine learning in general

Some recent research has shown that a small number of genes is sufficient for accurate diagnosis of most cancers, even though the number of genes vary greatly between different diseases (Xiong *et al.* (2001); Krishnapuram & et al. (2004)). Indeed, a large set of gene expression level may decrease the classification accuracy due to the phenomenon known as curse of dimensionality, in which the risk of over-fitting increases as the number of selected genes grows. It means each cancer data set has an optimal subset of genes for classification, and the range of this set should be small. More importantly, by using a small subset of genes, we can not only get a better diagnostic accuracy, but also get an opportunity to further analyse the nature of the disease and the genetic mechanisms responsible for it. However, the best subset of genes is usually unknown (Yeung *et al.* (2005)). Therefore, the microarray cancer classification problem can be classified as the combinational optimization problem with two main objectives: minimizing the number of selected genes and maximizing the classification accuracy.

The problem of feature selection received a thorough treatment in pattern recognition and machine learning. The gene expression data sets are problematic in that they contain a large number of genes (features) and, thus, methods that search over subsets of features can be prohibitively expensive. Moreover, these data sets contain only a small number of samples, so the detection of irrelevant genes can suffer from statistical instabilities. Two basic approaches for feature selection are used in machine learning and information theory literature (Inza *et al.* (2004); Xiong *et al.* (2001)): the filter and wrapper method.

- Filter methods calculate the goodness of the proposed feature subset based on the relation of each single gene with the class label, by using some simple statistics approaches. The most common way is to rank all features in terms of the values of an univariate scoring metric. The top ranked features are selected for classification.

- In wapper methods, a search is conducted in the space of genes (Xiong *et al.* (2001)). Evaluate the goodness of each found gene subset by the estimation of the accuracy percentage of the specific classifier to be used, then train the classifier only with the found genes.

Filter procedures are used in most of the works in the area of microarray analysis. The wrapper approach, which is popular in many machine learning applications, is not extensively used in DNA microarray tasks. Accordingly, some most widely used filter methods are introduced and adopted in our work. More detailed discussions about these two approaches was described in Inza *et al.* (2004).

Many gene selection strategies have been proposed in the literature. The performance of feature selection methods seems to be problem-dependent. It means if a strategy work well, often depends on the nature of the data and expected results of the concerned microarray data. Generally, feature selection is done by ranking genes on the basis of scores, correlation co-efficient, mutual information and sensitivity analysis. Some special approaches like use of Signal to Noise Data (SND) (Golub *et al.* (2002); Slonim *et al.* (2000)), Thershold Number of Misclassification (TNoM) (Ben-Dor *et al.* (2000)), Genetic Algorithm (GA)(Li *et al.* (2001)), voting technique and perception method are also in use (Prabakaran

et al. (2005)). Two most widely used gene selection methods will be introduced in Section 4.2.1 and Section 4.2.2.

4.2.1 Information Gain

Information Gain (IG) technique uses the concept of Shannon entropy (Xing *et al.* (2001)). Given entropy E as a measure of impurity in a set of training samples, it is possible to define a measure of the effectiveness of a feature/gene in classifying the training data. This measure is simply the expected reduction in entropy caused by partitioning the data according to this feature, so-called Information Gain (Mitchell (1997); Yu & Liu (2004)). Assume a given set of microarray gene expression data M, the information gain of a gene i is defined as:

$$IG(M, i) = E(M) - \sum_{v \in V(i)} \frac{M_v}{M} E(M_v), \tag{4.3}$$

where $V(i)$ is the set of all possible values of feature i, M_v is the subset of M for which feature I has value v, $E(M)$ is the entropy of the entire set, and $E(M_v)$ is the entropy of the subset M_v. The entropy function E is defined by:

$$E = \sum_{j=1}^{c} -\frac{|C_j|}{|\sum C|} log_2 \frac{|C_j|}{|\sum C|} \tag{4.4}$$

where $|C_j|$ is the number of samples in class C_j. The entropy is supposed to give the information required in bits, and is traditionally used to deal with boolean valued features (hot/cold, true/false, etc). Fortunately, this method can be extended to handle the data with continuous valued features, for example, microarray gene expression data. A case study of applying this method to select best subset of leukemia cancer data will be given in Section 4.2.3

4.2.2 Signal to Noise Ratio

Signal to Noise Ratio (SNR) is essentially uesd in (Golub *et al.* (2002)) and (Slonim *et al.* (2000)). We also adopted this gene selection method in our work to find out the set of the most informative genes for training and testing. Slonim

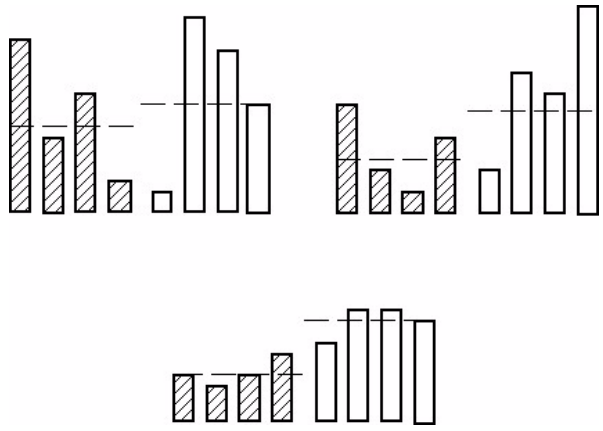

Figure 4.2: Select three genes from one data set. Each of them has eight samples. Samples 1-4 belong in one class, samples 5-8 belong in another class. The gene on the top left is unlikely to predict well because the mean of the class is quite close, this gene can not give us enough power to distinguish between classes. The mean of top right one and bottom are the same, but the bottom one has less variation around those means and so is likely to be a better classifiable gene.

et al. (2000) reported that the best performance was obtained with the relative class separation metric defined by:

$$SNR(g_i, c) = \frac{\mu_1 - \mu_2}{\sigma_1 + \sigma_2}, \tag{4.5}$$

where c is the class vector, g is the gene expression vector, μ_1 and μ_2 denote the mean expression level of g_i for the samples in class 1 and class 2, σ_1 and σ_2 is the standard deviation of expression for the samples in class 1 and class 2, respectively. Then we take the gene with the highest scores as our top features for the next classification task.

SNR hints that the separation between two classes of expression data is proportional to distance between their mean (Golub *et al.* (2002)). Further more, this distance is normalized by the standard deviation of the classes. A large standard deviation value implies that we find points in the group far away from the mean value and that the separation would not be strong. For example, Figure 4.2 shows the selected gene on the top left is unlikely to predict well because the mean of the class is quite close, this gene can not give us enough power to

distinguish between classes. The mean of top right one and bottom are the same, but the bottom one has less variation around those means and so is likely to be a better classifiable gene. Unfortunately, this method is only expected to work well when the data is normally distributed in each class of samples (Golub *et al.* (2002)).

4.2.3 A Case Study

Unlike boolean attributes, every numeric attribute, like microarray gene expression data, has many possible split points. Targeting this problem, we sorted instances by the values of the numeric attribute, the place split points halfway between values to divide instances into several subsets , but the potential optimal split point is usually hard to find, and the breakpoints between values of the same class can not be optimal, in addition, the same class on both sides can not be optimal point. In order to find good breakpoints, we design two experiments.

- Study One: Sort instances according to the values of the gene. Then divide the attribute into two subsets at the value most close to the average value (without the same class on both sides), denoted as IG2.
- Study Two: Sort instances according to the values of the gene. Then divide the attribute into three subsets at the points most close to the 1/3 and 2/3 of the attribute (without the same class on both sides), denoted as IG3.

IG2 has been often widely used in many others's work. In this report, we adopt IG2 in order to compare the classification performance with other classifiers. Because our NF models use three membership functions to label the input as high/medium/low, IG3 seems more suitable for future analysis purposes. Top 20 ranked leukemia genes selected by IG2 and IG3 are shown in Table 4.3 and Table 4.4.

4.3 Classifiers

Several different supervised classifiers from machine learning area have been perviously used in classifying cancer-based gene expression data. In this section, we

Table 4.3: Top 20 ranked leukemia genes selected by IG2

Rank	ID	Gene name	Description
1	4050	X03934	GB DEF = T-cell antigen receptor gene T3-delta
2	6510	U23852	GB DEF = T-lymphocyte specific protein tyrosine kinase p56lck (lck) abberant mRNA
3	4342	X59871	TCF7 Transcription factor 7 (T-cell specific)
4	4055	X04145	CD3G CD3G antigen, gamma polypeptide (TiT3 complex)
5	5542	M37271	T-CELL ANTIGEN CD7 PRECURSOR
6	5543	M37271	T-CELL ANTIGEN CD7 PRECURSOR
7	5466	X58072	GATA3 GATA-binding protein 3
8	6606	X00437	TCRB T-cell receptor, beta cluster
9	1694	M12886	TCRB T-cell receptor, beta cluster
10	6696	X76223	GB DEF = MAL gene exon 4
11	1893	M28826	CD1B CD1b antigen (thymocyte antigen)
12	2833	U16954	(AF1q) mRNA
13	4357	X60992	T-CELL DIFFERENTIATION ANTIGEN CD6 PRECURSOR
14	**4847**	**X95735**	Zyxin
15	1106	J04132	CD3Z CD3Z antigen, zeta polypeptide (TiT3 complex)
16	3332	U50743	Na,K-ATPase gamma subunit mRNA
17	**6236**	**U83239**	CC chemokine STCP-1 mRNA
18	4484	X69398	CD47 CD47 antigen (Rh-related antigen, integrin-associated signal transducer)
19	4291	X56468	14-3-3 PROTEIN TAU
20	2454	S65738	Actin depolymerizing factor [human, fetal brain, mRNA, 1452 nt]

Table 4.4: Top 20 ranked leukemia genes selected by IG3

Rank	ID	Gene name	Description
1	4050	X03934	GB DEF = T-cell antigen receptor gene T3-delta
2	6510	U23852	GB DEF = T-lymphocyte specific protein tyrosine kinase p56lck (lck) abberant mRNA
3	4342	X59871	TCF7 Transcription factor 7 (T-cell specific)
4	4055	X04145	CD3G CD3G antigen, gamma polypeptide (TiT3 complex)
5	5542	M37271	T-CELL ANTIGEN CD7 PRECURSOR
6	5543	M37271	T-CELL ANTIGEN CD7 PRECURSOR
7	5466	X58072	GATA3 GATA-binding protein 3
8	6606	X00437	TCRB T-cell receptor, beta cluster
9	1694	M12886	TCRB T-cell receptor, beta cluster
10	6696	X76223	GB DEF = MAL gene exon 4
11	1893	M28826	CD1B CD1b antigen (thymocyte antigen)
12	2833	U16954	(AF1q) mRNA
13	4357	X60992	T-CELL DIFFERENTIATION ANTIGEN CD6 PRECURSOR
14	**4847**	**X95735**	Zyxin
15	1106	J04132	CD3Z CD3Z antigen, zeta polypeptide (TiT3 complex)
16	3332	U50743	Na,K-ATPase gamma subunit mRNA
17	**6236**	**U83239**	CC chemokine STCP-1 mRNA
18	4484	X69398	CD47 CD47 antigen (Rh-related antigen, integrin-associated signal transducer)
19	4291	X56468	14-3-3 PROTEIN TAU
20	2454	S65738	Actin depolymerizing factor [human, fetal brain, mRNA, 1452 nt]

introduce three most widely used models, including MLP (Xu *et al.* (2002) and Khan *et al.* (2001)), KNN (Li *et al.* (2001); Jirapech-Umpai & Aitken (2005)), and SVM (Brown *et al.* (2000); Shi & Chen (2005)).

4.3.1 Multi-Layer Perceptron

Multi-Layer Perceptrons (MLP) are also known as three layered forward neural networks. A feed-forward MLP consists of one input layer of nodes, one output layer of nodes, and one or more layers of hidden nodes. A node is only connected with the units in its next neighboring layers, it only allows signals to travel one way, from input to output. There is no feedback, the output of any layer does not affect that same layer (see Figure 2.4). MLPs are back-box systems training by certain learning algorithms. The most popular one is Backpropagation (BP). The weight between different neurons is adjusted by BP according to the error value between the real output and expected output. More details can be found in Section 2.2.

4.3.2 Support Vector Machine

Support Vector Machines (SVMs) were developed in early 1990s (Boser *et al.* (1992)), and they have become one of the standard tools for machine learning and data mining along with neural networks. The successful applications include hand-written character recognition, text categorisation, image classification, etc. They have also been used for the classification of microarray gene expression data by (Brown *et al.* (2000)). Excellent descriptions of SVMs can be found in the books of Vapnik (1995); Vapnik (1999). Here we just provide an short introduction of using SVMs for classification of gene expression data.

Let M be the given gene expression vectors, where a kernel function is defined as Equation 4.6, and is often chosen as a polynomial of degree d.

$$K(M, m_i) = (m^T m_i + 1)^d, \tag{4.6}$$

The discriminant function is defined by:

$$L(M) = \sum_{i=1}^{T} \alpha_i c_i K(M, m_i), \tag{4.7}$$

where $m_{i\,i=1}^{T}$ is a set of gene vectors and $c_{i\,i=1}^{T}$ is the corresponding class, α_i is the weight of training sample y_i. It denotes the strength with which that sample is embedded in the final decision function. Only a part of the training vectors will be associated with a non-zero α_i. These vectors are so-called support vectors. The training process is to update the weights α_i to maximize the distance between the samples from two classes, viz., maximize the following object function:

$$F(a) = \sum_{i=1}^{T} \alpha_i (2 - \beta_i L(m_i)) = 2 \sum_{i=1}^{T} \alpha_i - \sum_{i=1}^{T} \sum_{j=1}^{T} \alpha_i \alpha_j \beta_i beta_i K(y_i, y_j) \tag{4.8}$$

where $\alpha_i \geq 0$. The disadvantages of SVMs are that it can only be used for two-class classification problems. If one wants to use it for multi-class classification problems, it has to be treated as a series of dichotomous classification problems (Bennett & Campbell (2000)).

4.3.3 K Nearest Neighbour

K Nearest Neighbours (KNN) is one of the simplest classifiers in use. It was first introduced in 1951 by (Bressan & Vitri; (2003)), and since then there have been many extensions and variations proposed, e.g., the probabilistic nearest neighbour model (Holmes (2002)). KNN methods do not have a training phase and the class of a new sample is simply predicted to be the most common class among the k nearest neighbours. In our case, the decision function is defined by:

$$L(M) = \sum_{i=1}^{T} \alpha_i c_i K(M, m_i), \tag{4.9}$$

where K is the set of neighbours closest to the new point, x. The time complexity of this method is $O(N)$, where N is the number of training samples. Compared to SVMs, this method can easily be extended to multi-class classification with the class of a new point determined by the consensus of its neighbours. The set of nearest neighbours is determined by a distance metric, e.g., Euclidean Distance (ED), see Equation 4.10, or the Cosine Coefficient (CC) between the two gene expression vectors, see Equation 4.11:

$$ED = \sqrt{\sum (m_i - m_j)^2}, \tag{4.10}$$

$$CC = \frac{\sum m_i m_j}{\sqrt{\sum m_i^2 \sum m_j^2}}, \tag{4.11}$$

where m_i and m_j represent two gene expression vectors. Generally it is best to scale the influence of each neighbour depending on the distance from the new point. This is easily accomplished by multiplying the class labels of each neighbour by a weighting term.

Chapter 5

Neuro-Fuzzy Modeling

Adaptive-Network-based Fuzzy Inference System (ANFIS) was first proposed by Jang (Jang (1992); Jang & Sun (1995); Jang & Sun (1997)). ANFIS can be easily implemented for a given input/ouput task, and hence is attractive for many application purposes. It has been successfully applied in many different areas (Garzon *et al.* (2002); Belal *et al.* (2002); Virant-Klun & Virant (1999)). In this chapter, we will give a more detailed introduction (Section 5.1) and apply it to microarray cancer data (Section 5.2). The analytical work about this method will also be given.

5.1 Adaptive-Network-based Fuzzy Inference System

Adaptive-Network-based Fuzzy Inference System (ANFIS) is a Sugeno-like fuzzy system in a five-layered network structure (see Figure 5.1). Back-propagation strategy is used to train the membership functions, while the least mean squares algorithm determines the coefficients of the linear combinations in the consequent part of the model. Takagi and Sugeno type fuzzy if-then rules (TSK) (Sugeno & Kang (1988); Takagi & Sugeno (1985)) are used in ANFIS model, for example:

$$\text{If (x is } A_1\text{) and (y is } B_1\text{), then} f_1 = p_1 x + q_1 y + r_1, \tag{5.1}$$

$$\text{If (x is } A_2\text{) and (y is } B_2\text{), then} f_2 = p_2 x + q_2 y + r_2, \tag{5.2}$$

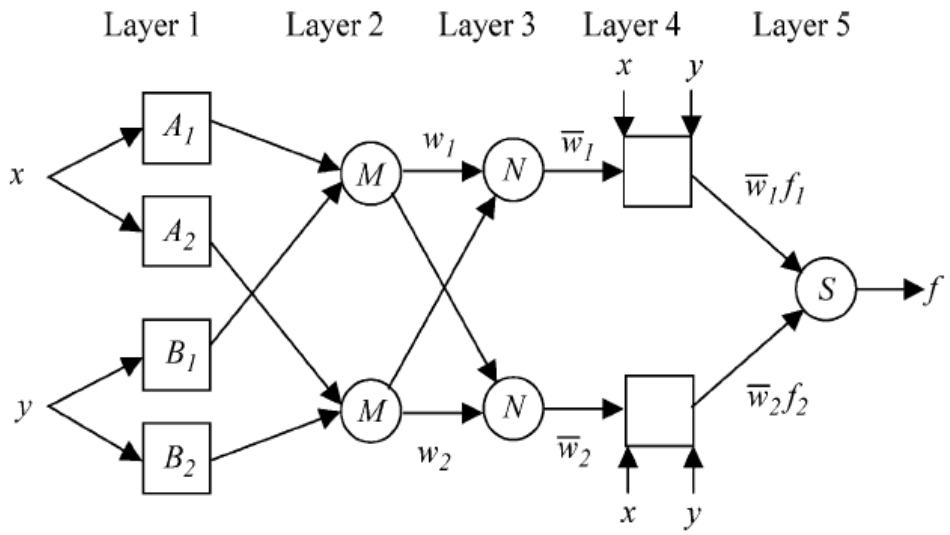

Figure 5.1: ANFIS architecture Jang & Sun (1997)

where x and y are the inputs, f_i is the output, p_i, q_i and r_i are the design parameters that are determined by the users during the training process. A_i and B_i are the fuzzy sets according to pre-defined membership function. An ANFIS model with two inputs and two fuzzy rules are implemented in Figure 5.1.

The first hidden layer is for fuzzification of the input variables. The outputs of layer 1 are fuzzy membership grade of the inputs, which are given by:

$$O_i^1 = \alpha_{A_i}(x), i = 1, 2 \tag{5.3}$$

$$O_i^1 = \alpha_{B_{i-2}}(y), i = 3, 4 \tag{5.4}$$

where $\alpha_{A_i}(x)$, $\alpha_{B_{i-2}}(y)$ are membership function. If the bell-shaped membership function is employed, $\alpha_{A_i}(x)$ is defined by:

$$\alpha_{A_i}(x) = \frac{1}{1 + \left\{ \left(\frac{x - c_i}{a_i} \right)^2 \right\}^{b_i}} \tag{5.5}$$

where a_i,b_i and c_i are the parameters of bell-shaped function, and can be used to define the region of the fuzzy sets.

There are fixed number of nodes in the second layer, labeled with M. The outputs of the second layer can be defined as:

$$O_i^2 = w_i = \alpha_{A_i}(x)\alpha_{B_i}(y), i = 1, 2 \tag{5.6}$$

where w_is are so-called firing strength of the rules.

In the third layer, the number of nodes is also fixed, labeled with N. It normalizes the rule strengths from the second layer. The output of this layer can be defined as:

$$O_i^3 = \bar{w}_i = \frac{w_i}{w_1 + w_2}, i = 1, 2 \tag{5.7}$$

which are the so-called normalized firing strengths.

The consequent parameters of the rule are determined in the fourth layer. The output of each node in this layer is the product of the normalized firing strength and the polynomial defined in fuzzy rule, shown as:

$$O_i^4 = \bar{w}_i f_i = \bar{w}_i(p_i x + q_i y + r_i), i = 1, 2 \tag{5.8}$$

The fifth layer computes the overall output as the summation of all incoming signals. There is only one node in this layer, labeled with S. Hence, the output of this layer can be presented as:

$$O_i^5 = \sum_{i=1}^{2} \bar{w}_i f_i = \frac{\left(\sum_{i=1}^{2} w_i f_i\right)}{w_1 + w_2} \tag{5.9}$$

There are two adaptive layers in the ANFIS architecture, namely the first layer and the fourth layer. There are three modifiable parameters in the first layer a_i, b_i, c_i, so-called premise parameters, which are related to the shape of the membership function. In the fourth layer, there are also three modifiable parameters p_i, q_i, r_i, so-called consequent parameters, which are related to the output of the first order polynomial.

A training process of ANFIS is to tune all these six parameters, so that the model can give a satisfying output. If we apply the standard backpropagation method to adjust the parameters, the method is generally slow and likely to become trapped in local minima. Suggested by Jang (1992), we combined the

gradient method and the least squares estimate (LSE) to identify the parameters of the network.

When the premise parameters are fixed, the output of the ANFIS can be written as:

$$f = \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_2 = \bar{w}_1 f_1 + \bar{w}_2 f_1 \tag{5.10}$$

Substituting the Equation 5.8 into the Equation 5.10:

$$f = \bar{w}_1(p_1 x + q_1 y + r_1) + \bar{w}_2(p_2 x + q_2 y + r_2) \tag{5.11}$$

Rearrange the Equation 5.12, we get:

$$f = (\bar{w}_1 x)p_1 + (\bar{w}_1 y)q_1 + (\bar{w}_1)r_1 + \bar{w}_2 x)p_2 + (\bar{w}_2 y)q_2 + (\bar{w}_2)r_2 \tag{5.12}$$

which is a linear combination of the modifiable consequent parameters p_1, q_1, r_1, p_2, q_2 and r_2. The least squares method can be used to identify the optimal values of these parameters easily. In each epoch, the LSE method is used to optimize the consequent parameters, while the premise parameters are fixed. The output of the ANFIS is calculated by employing the consequent parameters found in the forward pass. Once the optimal consequent parameters are found, the BP method will immediately start to adjust the premise parameters corresponding to the fuzzy sets in the input domain immediately, according to the output error. It has been proven that this hybrid algorithm is highly efficient comparing with a standard gradient method in training the ANFIS (Nauck (1997)).

5.2 Apply ANFIS to Microarray Analysis

We apply a single NF model, ANFIS to microarray cancer classification, and test it on three benchmark microarray cancer data sets. It can be observed that there are several advantages of applying ANFIS to microarray gene expression data.

- Firstly, ANFIS is relatively fast to convergence, due to its hybrid learning strategy, and it is easy to interpret. Users can adjust the output results by adding or deleting rules.

- Secondly, a Neuro-Fuzzy system can always (i.e. before, during, and after learning) be interpreted as a system of fuzzy rules. It is possible to create the system out of training data from scratch, and it is possible to initialize it by prior knowledge in the form of fuzzy rules.

- Thirdly, the trained membership function and rules can help researchers to better understand unknown data.

As we described in Chapter 3, microarray gene expression data analysis is a high-dimensional-low-sample problem. The collection of well-distributed, sufficient, and accurately measured input data is the basic requirement to obtain an accurate model (Belal *et al.* (2002)). Selection of the ANFIS inputs is the most important task of designing the classifier, since even the best classifier will perform poorly if the inputs are not selected well enough. It is difficult for ANFIS to handle high dimensional problems, as this leads to a large number of input nodes, rules, and, hence, consequent parameters. Some important research questions appear:

- How many inputs can a single ANFIS model deal with when the model is trained on a regular personal computer?

- Is it enough to represent a given microarray gene expression data set?

- If not, how to fit the data better and get better classification results?

The number of selected genes is the same as the number of ANFIS inputs. If the number of input is N, the number of membership function for each input variable is K, then the number of the fuzzy rules R is given by:

$$R = K^N \tag{5.13}$$

The number of the adaptive parameters P is given by:

$$P = K^{N+1} + K^N \times N \tag{5.14}$$

where K^{N+1} is called non-linear adaptive parameters, $K^N \times N$ is called linear adaptive parameters. When the number of membership functions for each input

Number of inputs	Number of rules	Number of parameters
2	9	45
3	27	108
4	81	360
5	243	774
6	729	2484
7	2187	7008
...	...	...

Table 5.1: The relationship among the number of input features, the number of fuzzy rules, and the number of parameters need to be updated in each epoch.

is set to three, the relationship between the inputs and rules are shown in Table 5.1. From Equation 5.13, Equation 5.14 and Table 5.1, we can see that the computation cost increases very quickly while the number of inputs grows. We simulated the models on an IBM R51 laptop (CPU: PIV-1.5G, Memory: 1G). The computer is out of memory when the number of input is larger than 6. So for the first question, it's very inconvenient for a common user to use ANFIS model to analysis microarray cancer gene expression data, when the number of selected gene is larger than 6 the result will be given in Chapter 7. The same as current fuzzy based models, our ANFIS model is a small model, and only can deal with small data sets. For the last question, we need to design some strategies to enable the model to accept more inputs with less computation cost. Common approaches include (Nauck (1997)):

1. Evaluate and select rules: if the system creates too many rules, it is possible to evaluate them and to keep only the best rules. If the learning process starts with a large number of rules, and many of them have a poor performance measure (Nauck *et al.* (1997)), these can be deleted from the rule base. This strategy usually is very useful. ANFIS model is easy to interpret. The users can define the number of useful rules as k, several machine learning techniques can be applied here: start the models with 1 or 2 rules, then increase the number of rules by adding the useful rules one by one until k, so-called constructive algorithms (Burgess (1994); Wang *et al.* (2004)); or start the models with a large number of rules, decrease the number of

rules by deleting the useless rules one by one until k, so-called pruning algorithms (Fürnkranz (1997); Reed (1993)); or use evolutionary algorithms or some other optimization algorithms to select the best k rules (Yao & Liu (1997); (Xin (1995))). More details will be discussed in Chapter 8.

2. Delete antecedents: for each rule, some "IF" conditions in the antecedents part are repeatedly defined or have negative performance for the final decision The user can delete these conditions from the rule. But this strategy leads to an inconsistent rule base, therefore the rule has to be repaired before being used.

3. Delete fuzzy sets: if a variable is partitioned by more than two fuzzy sets, sometimes the support for one or more of them can become quite large during learning. This can be seen as evidence that such a fuzzy set is superfluous. The user can specify a percentage value, and all fuzzy sets that cover more percent of the domain can be deleted, which leads to a reduction of variables in the antecedents. This procedure can also lead to an inconsistent rule base, which will be noticed by the rule editor.

Besides the approaches above, we design a novel model, called Neuro-Fuzzy Ensemble (NFE), which uses several ANFIS models to learn different parts of the data, in order to let the whole system to have the ability to accept higher input dimensions. The next model allows the model to fit the data better if small number of genes can not well present the data, and it can also obtain an extra classification accuracy by applying the nature of ensemble learning. More details about ensemble learning and NFE will be introduced in Chapter 6. Both ANFIS model and NFE model will be tested, and the experimental results will be given in Chapter 7.

5.3 Training and Testing Strategy

We train our model in order to obtain a good generalization ability, but using the testing data to do this would clearly be cheating (Schaffer (1993)). Usually, users divide the database into three sub-databases. One of them is used for the

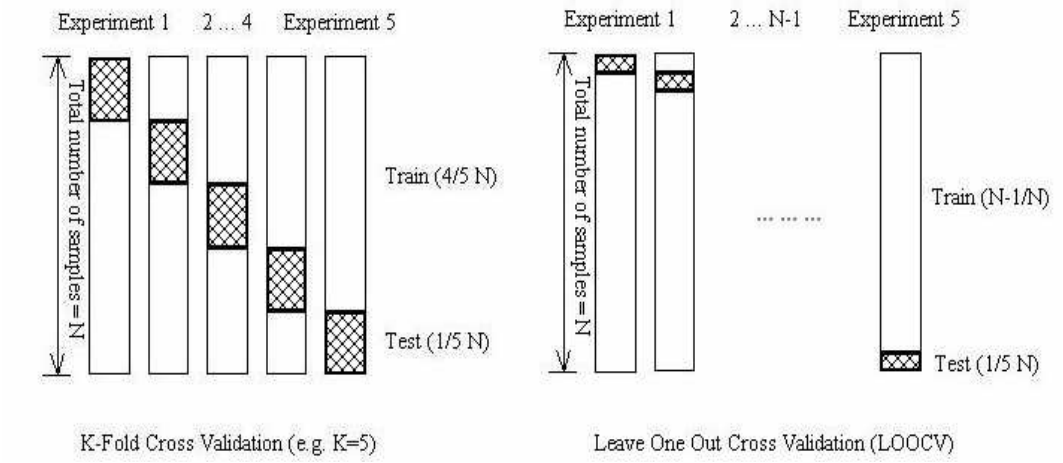

Figure 5.2: K-fold Cross Validation (left) and LOOCV (right)

training the networks, another one is used for testing the training performance, and last one is used for withhold over-fitting, called validation set. For large data sets, this strategy works quite well.

For microarray gene expression data sets, it is difficult to show which model is significantly better than another with such a small number of samples. We abandon the traditional training strategies from machine learning (three subsets method) for two reasons: firstly, one incorrect classification may cause the accuracy decrease greatly. The variety of accuracy can not represent the true difference between two methods, since small number of incorrect classification always happen even we use the same model by different experiments. Secondly, due to the fact that the number of samples is too small, each subset can not fully represent the space, it is not meaningful to train on one space, but test on another very different one. In order to train on as many examples as possible (Ding & Peng (2003)), another strategy has been considered, leave one out cross validation (LOOCV or jackknife strategy). We divide all samples at random into K distinct subset, where K equals to the number of samples (see Figure 5.2). Train the model using $K-1$ subsets, and test the training performance on the Kth sample. The LOOCV accuracy is obtained by:

$$LOOCV accuracy = \frac{Acs}{K} \tag{5.15}$$

where Acs is the number of correctly classified samples in K experiments. LOOCV accuracy is strongly suggested to be used as an evaluation of microarray data classification by many other researchers.

Chapter 6

Neuro-Fuzzy Ensemble Modeling

In recent years, ensemble learning attracts the attention of more and more researchers in machine learning, several ensemble techniques have been proposed, as it has been shown that they can significantly enhance the accuracy of classification tasks. Here we construct a Neuro-Fuzzy Ensemble model by combining several single NF models to learn the same data with different subset of genes. This approach not just allow the models to study more genes when small gene subset can not well represent the whole data set, but also obtain a better classification performance due to the good generalization ability of the ensemble model itself. Some basic notions of ensemble learning and the reasons of why ensemble learning can perform well is explained from several different views.

6.1 Ensemble Learning

Classifier Ensemble (CE) is a collection of finite numbers of individual classifiers that are trained for the same tasks (Hansen & Sakamon (1990)), see Figure 6.2. CE is expected to fuse a better overall decision from its constituents or experts, and the overall decision should be better than the result given by any individual expert. Such kind of knowledge appeared in the classic machine learning area is Committee Machine. Committee machines can be classified into two major categories (Haykin (1996)):

- First, dynamic structures: The input signal is directly involved in actuating the mechanism that integrates or combines the constituent outputs. The

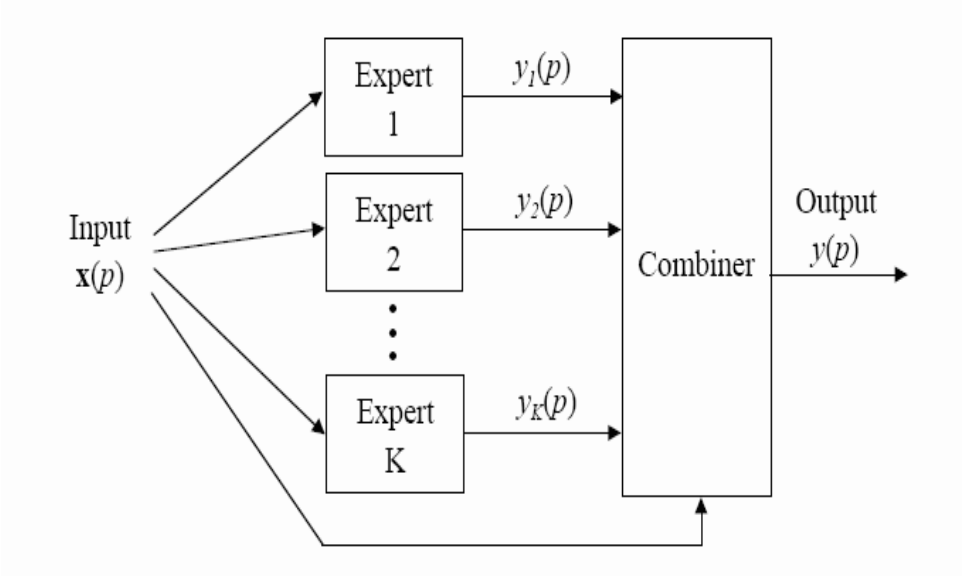

Figure 6.1: The structure of a mixture of experts.

main example is a mixture of experts, see Figure 6.1.

- Second, static structures: where the output of several expert network are combined by a mechanism that does not involve the input signal, for example, classifier ensemble, see Figure 6.2.

Some recent work shows that the ability of a neural network ensemble is determined by the accuracy and diversity of the individual networks in the ensemble (Kuncheva & Whitaker (2003); Liu *et al.* (2003); Thomas (2000)). Further details about "why ensemble is better than individual?" is given in the next section.

6.1.1 Ensemble vs. Individual

In this section, we will further explain why an ensemble of classifiers is better than one individual classifier. Consider the standard supervised learning problem. A learning algorithm is given training patterns(examples) of the form $(x_1, t_1), (x_1, t_1), ..., (x_q, t_q)$ to learn some unknown function $t = f(x)$. The t values are typically approximated to a discrete set of classes in case of classification problems or from the real line in the case of regression problems (Thomas (2000)).

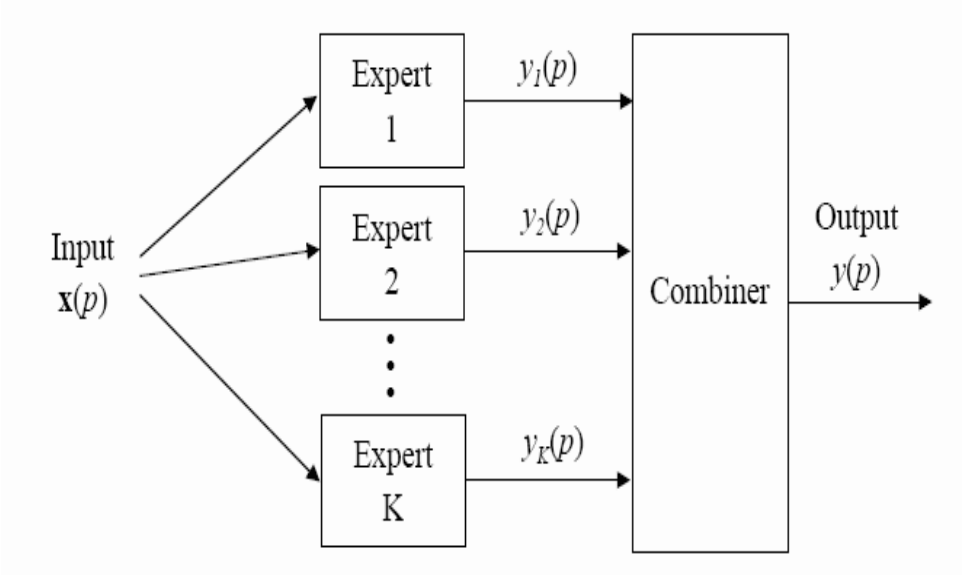

Figure 6.2: The structure of an ensemble classifier.

(In this stage, we only consider classification problem, i.e., medical diagnosis problems). The x_i values are typically vectors of the form $x_{k1}, x_{i2}, ..., x_{km}$, whose components are discrete- or real-valued variables such as height, weight, color, age, and so on, the so called features of x_i.

Now given a set of S training patterns, a learning algorithm generates a classifier (using the training set). The classifier is a hypothesis about the true function f. Given new x values (test set), it predicts the corresponding y values. We can denote classifiers by h_1, h_2, ..., h_l. With this in mind, an ensemble of classifiers can be redefined as a set of classifiers whose individual decisions are combined in some way to classify new examples.

There are three fundamental reasons (Thomas (2000)) from different points of view for the question "why ensemble may be better than individual?"

The first reason is from a statistical point of view. A learning algorithm can be seen as searching a space H of hypotheses to identify the best hypothesis (nearest to true hypotheses f) in the search space. When the training example available is not sufficient, the ensemble can find many different hypotheses in H that all give the same accuracy on the training data. Then different hypotheses

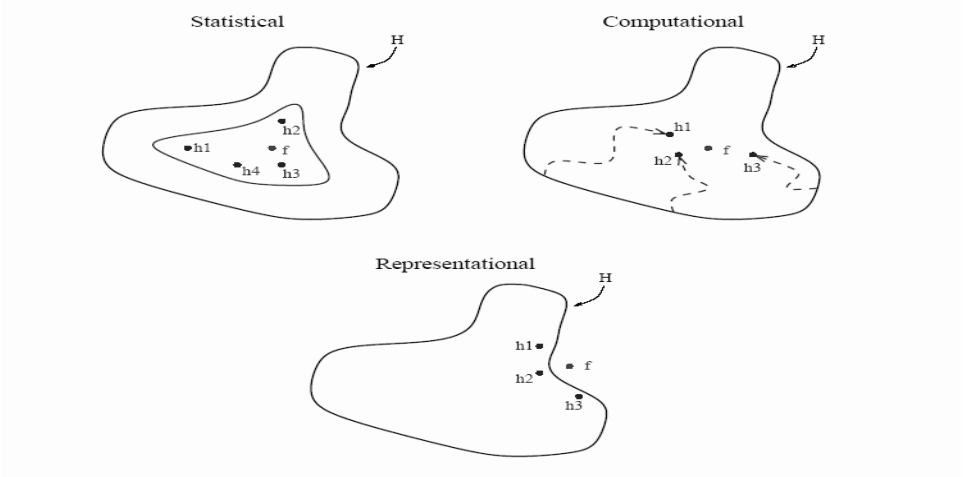

Figure 6.3: Three fundamental reasons for the Question: "why an ensemble may work better than an individual classifier?" The outer curve denotes the hypothesis space H; The inner cure denotes the set of hypotheses that all give good accuracy (< 0.5) on the training data. The point labeled f is the true hypothesis, i.e., global optimum (Thomas (2000)).

"average" their votes and reduce the risk of choosing the wrong classifier or give the worst hypothesis. From Figure 6.3 (top left), we can see that by averaging the accurate hypotheses, we can find a good approximation to f.

The second reason is a computational reason. The ensemble randomly initializes weights of individual classifier. This means all the individual classifiers run local search from different starting points. Combine such work helps avoid being tricked by local optima, and might be able to provide a better approximation to the true function f than any of the individual classifiers (see Figure 6.3 (top right)).

Representational reason serves as the third reason. In the most application of machine learning, the true function f can not be exactly represented by any of the hypotheses in H. The combination of a set of classifiers can expand the space of representational limitation, see Figure 6.3 (bottom).

6.1.2 Output of Ensemble

How to determine the output of the ensemble? There are three main strategies. The first one is Simple Averaging (SA), where the output of ensemble is formed by a simple averaging of output of individual NNs in the ensemble. The second strategy is Majority Voting (MV), where the output of the greatest number of individual NNs will be the output of the ensemble. If there is a tie, the output of the ensemble is rejected. The last one is winner-takes-all (WTA). For each pattern of the testing set, the output of the ensemble is only decided by the individual NN whose output has the highest activation. In this report, we adopt MV as the output strategy of our NFE model. Future study about how to combine this result is also very necessary, see Chapter 8.

6.2 Neuro-Fuzzy Ensemble

The main problem of our single NF models is that they hardly cope with a large number of genes, because of the high computation cost. In order to deal with this problem, we propose a new NF models, called Neuro-Fuzzy Ensemble models, which consists of several different single ANFIS models. Each model learns different subsets of genes, so that the whole model can work with a relatively large number of genes. Meanwhile, extra good performance can be obtained by the nature of the ensemble learning itself. We assume the maximum number of inputs for an individual in NFE is 4. Because we believe an individual NF model can perform very well on small number of genes, there is no need to construct an ensemble structure, when the number of selected genes is smaller than 5. The output strategy of our NFE model is MV. The main structure of NFE is shown in Figure 6.4. The NFE model is tested on three benchmark microarray data sets by using LOOCV training strategy.

The advantages of our NFE models can be summarized as follows:

- It allows the model to learn more features when the optimal subset of genes is relatively large.
- Normally, several classifiers have bigger computational cost than the individual classifier. Our NFE model seems to have much less computational

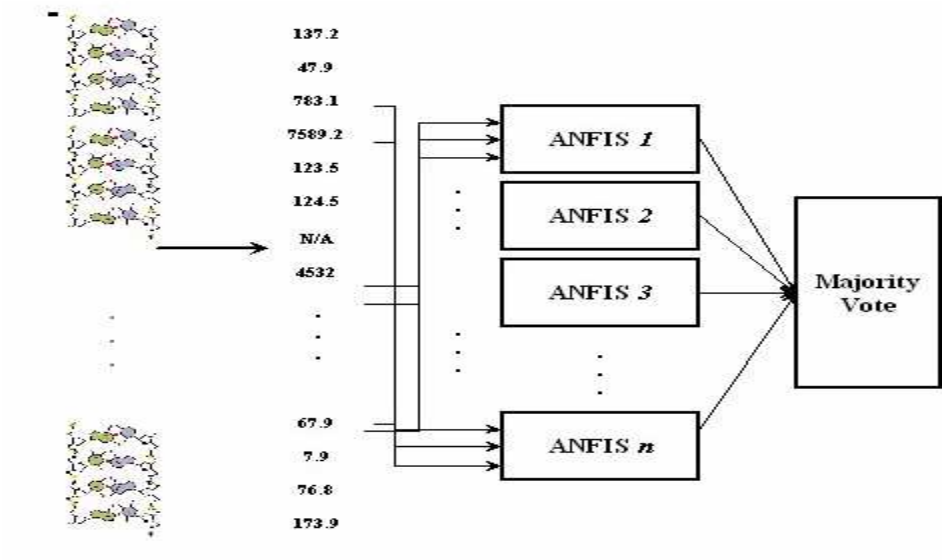

Figure 6.4: The main structure of NFE. N single ANFIS classifier in the ensemble, each single model has R inputs, so the whole model can use $R * N$ genes. The output of the ensemble is taken by simply majority voting (MV).

cost than the individual NF model when the necessary number of inputs is relatively large. Comparisons of computational cost between individual NF model and NFE are shown in Table6.1.

- NFE can significantly improve the generalization ability (classification performance) compared to single NF model, and they can also help address three classic machine learning problems: lack of data, local optima, and representational limitations. Lack of data is one of the main problems of microarray analysis.

- NFE can relieve the trial-and-error process by tuning architectures. To obtain a good model, the architecture of the model and the parameters of the model must be finely tuned, which is usually very difficult in practice. Use of a collection of models without finely tuning their architectures may still produce good performance.

But there are still some disadvantages: Firstly, bigger ensemble will increase the storage cost for generating and storing more models. Some worse individual

Table 6.1: Comparisons of computational cost between individual NF and NFE models. We compare the number of rules and parameters of individual NF and NFE models by using the same number of genes. We use two individual NF models with 3 inputs when the number of selected genes equals to 6. NoG denotes the number of selected genes, NoR denotes the number of rules, NoP denotes the number of parameters needed to be updated in each epoch.

NoG	NF		NFE	
	NoR	NoP	NoR	NoP
2	9	45	N/A	N/A
3	27	108	N/A	N/A
4	81	360	N/A	N/A
6	729	2484	54	216
8	6.6×10^4	7.9×10^5	162	720
12	1.7×10^7	2.7×10^8	243	1080
16	4.3×10^9	8.6×10^{10}	324	1440
20	1.1×10^{12}	2.6×10^{13}	405	1800
...	...	...	...	...

not only waste the system resource, but also decrease the whole ability of the ensemble. It's important to find the necessary number of individual NF models and select the most useful individuals. Secondly, the performance of NFE can be further enhanced by using other ensemble training techniques, i.e., bagging and boosting at the next stage, which will increase the comprehensibility of the learning process. Thirdly, the model becomes more complex than a single NF, and therefore more difficult for analysis.

Chapter 7

Experimental Results

In this chapter, we test NF model and NFE model on three benchmark gene expression microarray data sets: leukemia cancer data set, colon cancer data set, and lymphoma cancer data set. A short introduction about the data sets can be found in Section 7.1. The classification results of NF and NFE models are compared with some other popular classifiers in Section 7.3. Top 20 ranked genes selected by IG2 and SNR are listed in Section 7.2. The performances of these two gene selection methods are very similar, due to the fact that many overlapped genes are selected, especially in colon cancer data set and lymphoma cancer data set. Different from some approaches, our models not only just give the classification result, but also extract knowledge from the data, for example, the adjustment of membership function and fuzzy rules. Related analytical work is also given in Section 7.3.3. Here we adopt three important criteria to empirically evaluate the performance of our models:

- Number of selected genes
- Predictive accuracy on selected genes.
- Extracted knowledge from the trained model.

Before the experiments, we linearly scale all data in a small range [0, 1]. If y is a gene expression value of a gene g, the scaled value would be

$$g(a') = \frac{y - min(g)}{max(g) - min(g)} \tag{7.1}$$

where $min(g)$ and $max(g)$ is the minimum and maximum values of gene expressions in the database.

7.1 Cancer Data sets

There are many different benchmark microarray data sets, reported in published cancer gene expression studies, including leukemia cancer data set, colon cancer data set, lymphoma data set, breast cancer data set, NC160 data set, and ovarian cancer data set. In this study, the proposed models are tested on three data sets: leukemia cancer data set, colon cancer data set, lymphoma cancer data set. The aim of testing on several different data sets is not only to show that our models are better or worse, but also to find out when our models performs better (or worse) and why, what are the reasons causing the unsatisfying results, and how to solve the problems. Meanwhile, these data sets have been studied in many papers, so a comparison work can be made.

7.1.1 Colon Cancer Data Set

The data set we used here was firstly reported in Cho & Won (2003). The "cancer" biopsies were collected from tumors, and the "normal" biopsies were collected from healthy parts of the colons of the same patients Cho & Won (2003). This data set contains 62 samples. There are 40 tumor samples, and 22 normal samples. About 6000 genes represented in each sample in original data set, only 2000 genes were selected. The data is available at http://sdmc.i2r.a-star.edu.sg/rp/ColonTumor/ColonTumor.html.

7.1.2 Leukemia Cancer Data Set

The data set we used here was reported in Golub *et al.* (2002). The gene expression measurements were taken from 63 bone marrow samples and 9 peripheral blood samples Golub *et al.* (2002). This data set contains 72 samples. All samples can be divided into two subtypes: 25 samples of acute myeloid leukemia (AML) and 47 samples of acute lymphoblastic leukemia (ALL). The expression

levels of 7129 genes were reported. The data is available at http://sdmc.i2r.a-star.edu.sg/rp/Leukemia/ALLAML.html.

7.1.3 Lymphoma Cancer Data Set

The data set we used here was reported in Lossos *et al.* (2000). This data set contains 47 samples. B cell diffuse large cell lymphoma (B-DLCL) data set includes two subtypes: germinal center B cell-like DLCL and active B cell-like DLCL. The expression levels of 4026 genes were reported. 24 samples are germinal center B-like DLCL and 23 samples are active B cell-like DLCL. The data is available at http://www.genome.wi.mit.edu/MPR.

7.2 Gene Selection Results

In this section, we list the top 20 ranked genes selected by using IG2 and SNR methods. Top 4 ranked genes are selected for classification by using the single NF model. Top 20 ranked genes are used for classification by using the NFE model. The boldface data in Table 8.1, Table 4.3, Table 7.2, and Table 8.1 are overlapped genes by both IG2 and SNR methods.

7.2.1 Colon Cancer Data Set

In colon cancer data set, 20 genes with highest scores obtained by IG2 and SNR method were selected for classification. 13 in 20 genes are overlapped by using IG2 and SNR methods, see Table 8.1.

7.2.2 Leukemia Cancer Data Set

In leukemia cancer data set, 20 genes with highest scores obtained by SNR method were selected for classification. Gene selection results of using IG2 and IG3 are listed in Section 4.2.3. Different from other tested data sets, only 2 genes are overlapped by using IG2 and SNR methods, see Table 7.2.

Table 7.1: Top 20 ranked colon genes selected by IG2 and SNR. The boldface data are overlapped by the two methods.

	IG		SNR	
Rank	ID	Gene Name	ID	Gene Name
1	**249**	**M63391**	**249**	**M63391**
2	**493**	**R87126**	**765**	**M76378**
3	**1042**	**R36977**	**1772**	**H08393**
4	**1423**	**J02854**	**493**	**R87126**
5	**267**	**M76378**	**1423**	**J02854**
6	**245**	**M76378**	**245**	**M76378**
7	**399**	**U30825**	**1582**	**X63629**
8	**1772**	**H08393**	**267**	**M76378**
9	**765**	**M76378**	513	M22382
10	467	H40560	780	H40095
11	**822**	**T92451**	1771	J05032
12	258	M16937	**377**	**Z50753**
13	66	T71025	515	T56604
14	1067	T70062	**822**	**T92451**
15	**1325**	**T47377**	138	M26697
16	62	T48804	**1325**	**T47377**
17	**377**	**Z50753**	**1042**	**R36977**
18	1892	U25138	625	X12671
19	1808	U21090	1060	U09564
20	**1582**	**X63629**	**399**	**U30825**

Table 7.2: Top 20 ranked leukemia genes selected by SNR

Rank	ID	Gene	Description
1	2642	U05259	MB-1 gene
2	2335	M89957	IGB Immunoglobulin-associated beta (B29)
3	6225	M84371	CD19 gene
4	758	D88270	GB DEF = (lambda) DNA for immunoglobin light chain
5	4680	X82240	TCL1 gene (T cell leukemia) extracted from H.sapiens mRNA for Tcell leukemia/lymphoma 1
6	1685	M11722	Terminal transferase mRNA
7	5171	Z49194	OBF-1 mRNA for octamer binding factor 1
8	1078	J03473	ADPRT ADP-ribosyltransferase (NAD+; poly (ADP-ribose) polymerase)
9	6855	M31523	TCF3 Transcription factor 3 (E2A immunoglobulin enhancer binding factors E12/E47)
10	4318	X58529	IGHM Immunoglobulin mu
11	6974	M28170	CD19 CD19 antigen
12	**6236**	**U83239**	CC chemokine STCP-1 mRNA
13	3469	U59878	Low-Mr GTP-binding protein (RAB32) mRNA, partial cds
14	**4847**	**X95735**	Zyxin
15	5552	L06797	PROBABLE G PROTEIN-COUPLED RECEPTOR LCR1 HOMOLOG
16	2288	L33930	DF D component of complement (adipsin)
17	1882	M27891	CST3 Cystatin C (amyloid angiopathy and cerebral hemorrhage)
18	2010	M38690	CD9 CD9 antigen
19	5300	L08895	MEF2C MADS box transcription enhancer factor 2, polypeptide C (myocyte enhancer factor 2C)
20	1962	M33680	26-kDa cell surface protein TAPA-1 mRNA

Table 7.3: Top 20 ranked lymphoma genes selected by IG2 and SNR, the boldface data are overlapped by the two methods

	IG		SNR	
Rank	ID	Gene Name	ID	Gene Name
1	**1279**	**GENE3330X**	**1277**	**GENE3328X**
2	**1281**	**GENE3332X**	**1279**	**GENE3330X**
3	**1277**	**GENE3328X**	**1281**	**GENE3332X**
4	**1247**	**GENE3355X**	**1291**	**GENE3314X**
5	**1291**	**GENE3314X**	**1316**	**GENE3256X**
6	**2244**	**GENE1252X**	**2439**	**GENE3968X**
7	1206	GENE3228X	**2417**	**GENE3985X**
8	**1316**	**GENE3256X**	**2244**	**GENE1252X**
9	1287	GENE3338X	2438	GENE3967X
10	**3861**	**GENE1720X**	1312	GENE3258X
11	**2439**	**GENE3968X**	**3861**	**GENE1720X**
12	2263	GENE1296X	1274	GENE3325X
13	**2243**	**GENE1251X**	3860	GENE1719X
14	1634	GENE2704X	**1247**	**GENE3355X**
15	1616	GENE2662X	3020	GENE3608X
16	2415	GENE3987X	2205	GENE1212X
17	2200	GENE1207X	**2243**	**GENE1251X**
18	34	GENE4006X	2845	GENE740X
19	3137	GENE392X	3420	GENE3821X
20	**2417**	**GENE3985X**	2437	GENE3966X

7.2.3 Lymphoma Cancer Data Set

In lymphoma cancer data set, 20 genes with highest scores obtained by IG2 and SNR method were selected for classification. 11 in 20 genes are overlapped by using IG2 and SNR methods, see Table 8.1.

7.3 Results and Comparisons

We trained and tested NF and NFE models on three cancer gene expression data sets by using LOOCV strategy. Each variable has three membership functions in both NF models and NFE models, the initial membership function is bell-

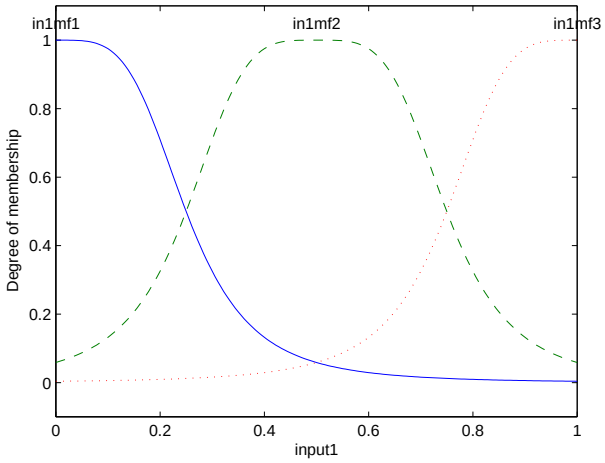

Figure 7.1: Initial membership function (bell-shaped function)

shaped function, see Figure 7.1. The proposed models were trained with the backpropagation method in combination with the least squares method, and bell-shaped membership functions were used during the training process. The cost function is the mean least-squares error (MSE), which measures the difference between the outputs and the targets. The number of selected genes and LOOCV accuracy are used to evaluate the effectiveness of the models. The final trained model is obtained by averaging all parameters of correctly classified models.

7.3.1 Results of ANFIS

We test ANFIS model on three data sets by using 2 genes and 4 genes, respectively. The classification results are shown in Table 7.4. The models with 4 genes performed better than the models with 2 genes on all three data sets, though the models with 2 genes concluded quite good results on colon and lymphoma data sets.

7.3.2 Results of NFE

Top 20 ranked genes are selected for classification by using the NFE model. There are 5 individual NF models in our NFE model, each NF model has 4 inputs. It

Table 7.4: Classification results of ANFIS model by using 2 and 4 genes selected respectively on colon cancer data set, leukemia cancer data set and lymphoma cancer data set. IG and SNR denote two different gene selection methods, information gain method and signal to noise ratio method. NoG denotes number of selected genes. LOOCV accuracies are used to evaluate the classification performance.

	NoG	IG	SNR
		LOOCV accuracy	LOOCV accuracy
Colon	2	80.65	82.56
Leukemia	2	73.61	69.44
Lymphoma	2	76.09	78.26
Colon	4	93.55	90.32
Leukemia	4	87.5	83.33
Lymphoma	4	87.23	89.13

totally allows the NFE to learn the data by using $5 * 4$ genes. The output of ensemble is obtained by using Majority Voting (MV). Other experimental sets, like initial membership function and cost function are the same as single NF model in Section 7.3.1. In colon cancer data set, all samples are accurately classified. In leukemia cancer data set, only 2 samples are inaccurately classified when the gene selection method is IG, and only 3 sample is inaccurately classified when the gene selection method is SNR. In lymphoma cancer data set, NFE accurately classified 45 samples out of a total of 46 by using either the IG or SNR method. The classification results also show that the IG method performed better than the SNR method. But there is no significant difference between IG2 and SNR methods, the number of accurate classified samples is usually only distinguished between 1 and 3. The reason for this phenomena can be explained by taking a look at the Table 8.1 and Table 7.3. The top 20 genes with higher mark of colon cancer and lymphoma cancer selected by IG and SNR are very similar, 23 overlapped genes in colon cancer data sets and 11 overlapped genes in lymphoma cancer data. It is very common to get similar results when we use very similar data sets.

Table 7.5: Classification results of NFE model by using 20 genes, on colon cancer data set, leukemia cancer data set and lymphoma cancer data set. NoG denotes number of selected genes. ATE denotes average training error. LOOCV accuracies are used to evaluate the classification performance.

	NoSG	IG	SNR
Colon	20	100.0	100.00
Leukemia	20	95.83	93.06
Lymphoma	20	95.65	95.65

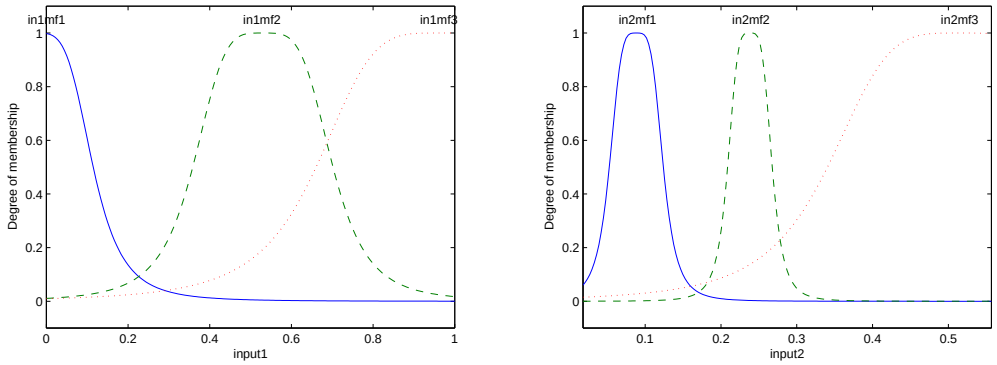

Figure 7.2: Adjusted membership function of ANFIS model in Colon Cancer Data Set. Number of selected genes = 2, gene selection method = IG.

7.3.3 Knowledge Extraction

Furthermore, different from other black-box approaches, NF models can extract some useful knowledge from the data, for example, adjusted membership function (see Figure 7.5, Figure 7.2, Figure 7.3, Figure 7.4, Figure 7.6, Figure 7.7 and Figure 7.8), and trained fuzzy rules (see Table 7.6, Table 7.7 and Table 7.8). All these knowledge is presented in human understandable form. This seems very attractive for the researcher to understand the data or explain how the result were obtained.

When the gene selection strategy is IG, and the number of selected genes is 2, the trained membership functions of ANFIS models in three tested data sets are shown in Figure 7.2, Figure 7.3 and Figure 7.4.

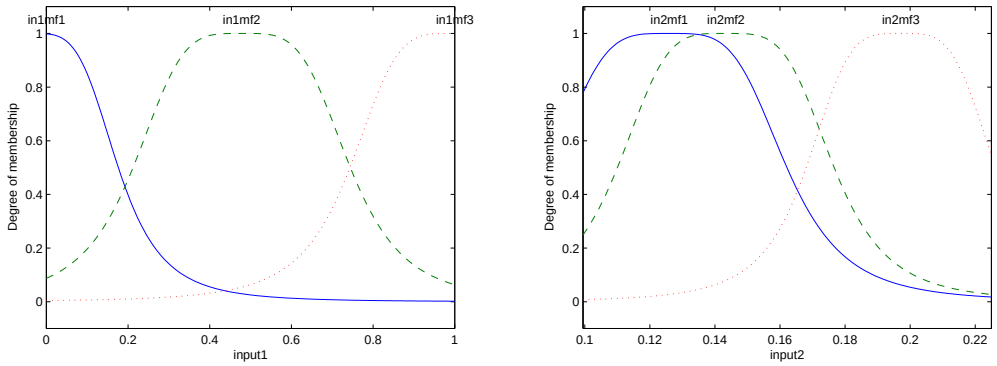

Figure 7.3: Adjusted membership function of ANFIS model in Leukemia Cancer Data Set. Number of selected genes = 2, gene selection method = IG.

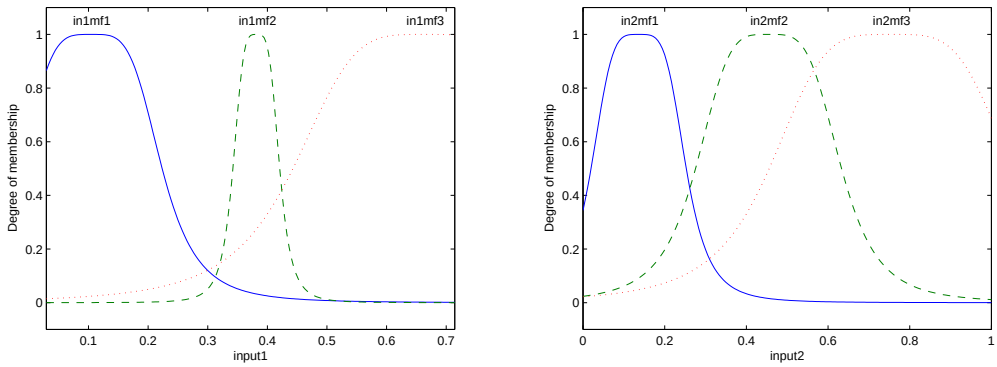

Figure 7.4: Adjusted membership function of ANFIS model in Lymphoma Cancer Data Set. Number of selected genes = 2, gene selection method = IG.

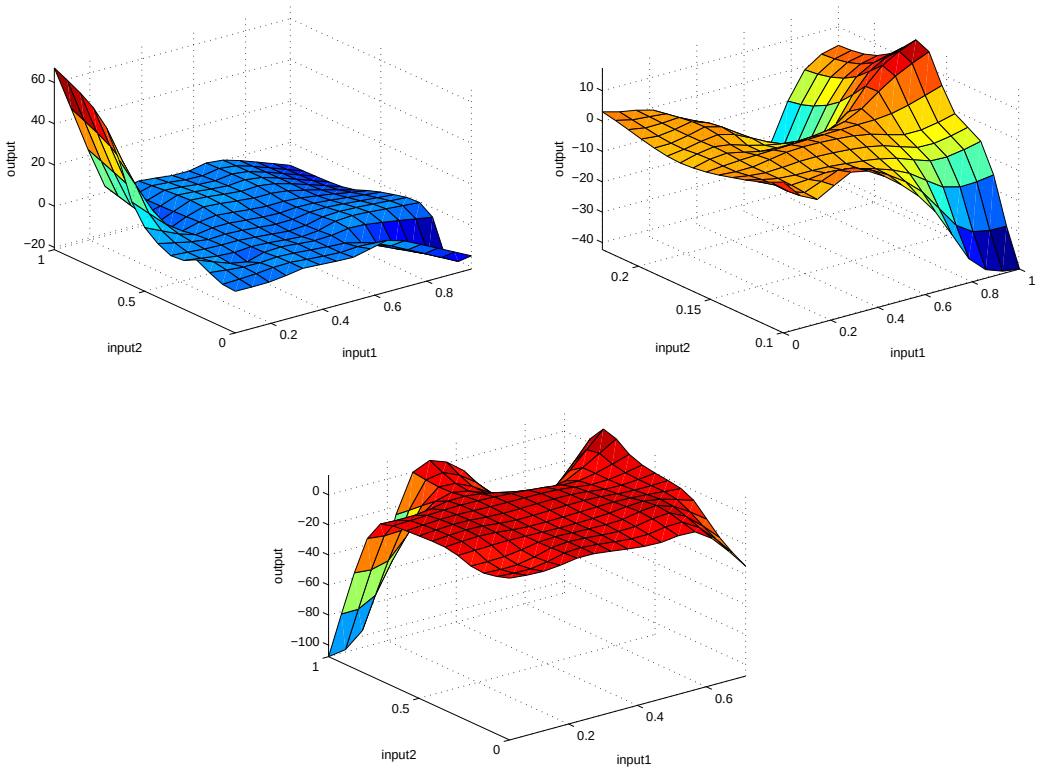

Figure 7.5: The fuzzy surface of train models when the number of selected genes = 2, gene selection method = IG.

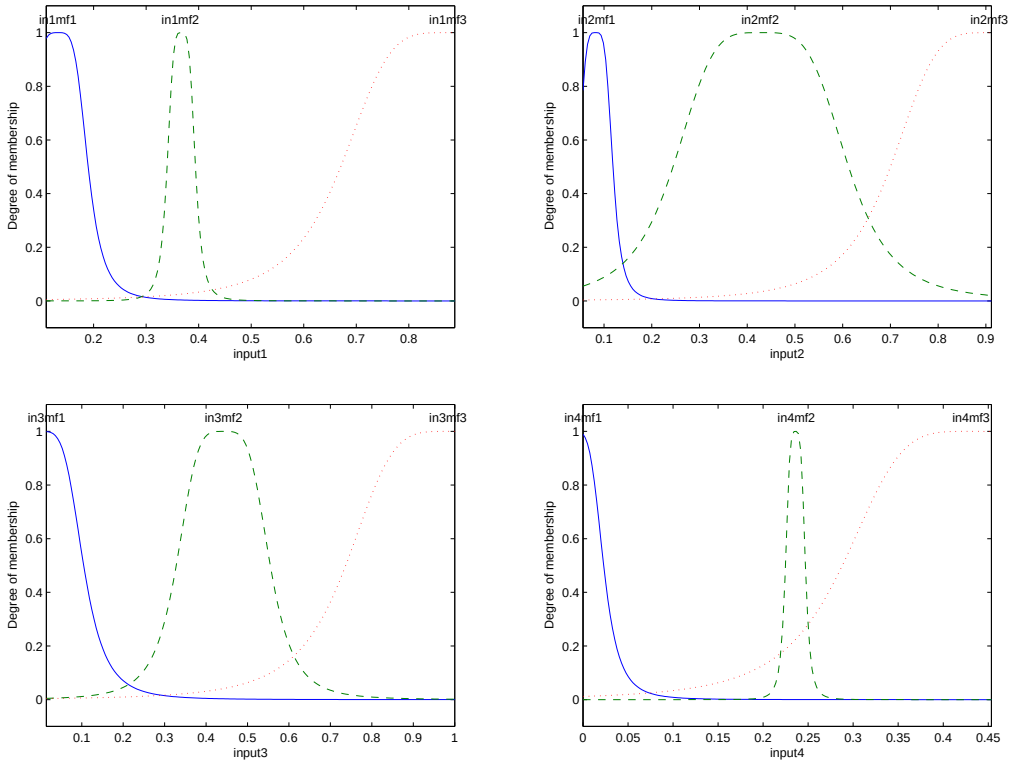

Figure 7.6: Adjusted membership functions of ANFIS model in Colon Cancer Data Set. Number of selected genes = 4, gene selection method = IG.

When the gene selection strategy is IG, and the number of selected genes is 4, the trained membership functions of ANFIS models in three tested data sets are shown in Figure 7.6, Figure 7.7 and Figure 7.8.

The classification results can be changed by modifying the rules. Here we select 5 rules from each trained models in order to give an insight of how the classification results are given.

7.3.4 Comparisons

The performance of our NF model and NFE model are compared with some previous studies, see Table 7.9. Our models obtained better results on Colon cancer data set, and similar results on Leukemia and Lymphoma data set, but both ANFIS and ANFIS ensemble models use less number of genes comparing

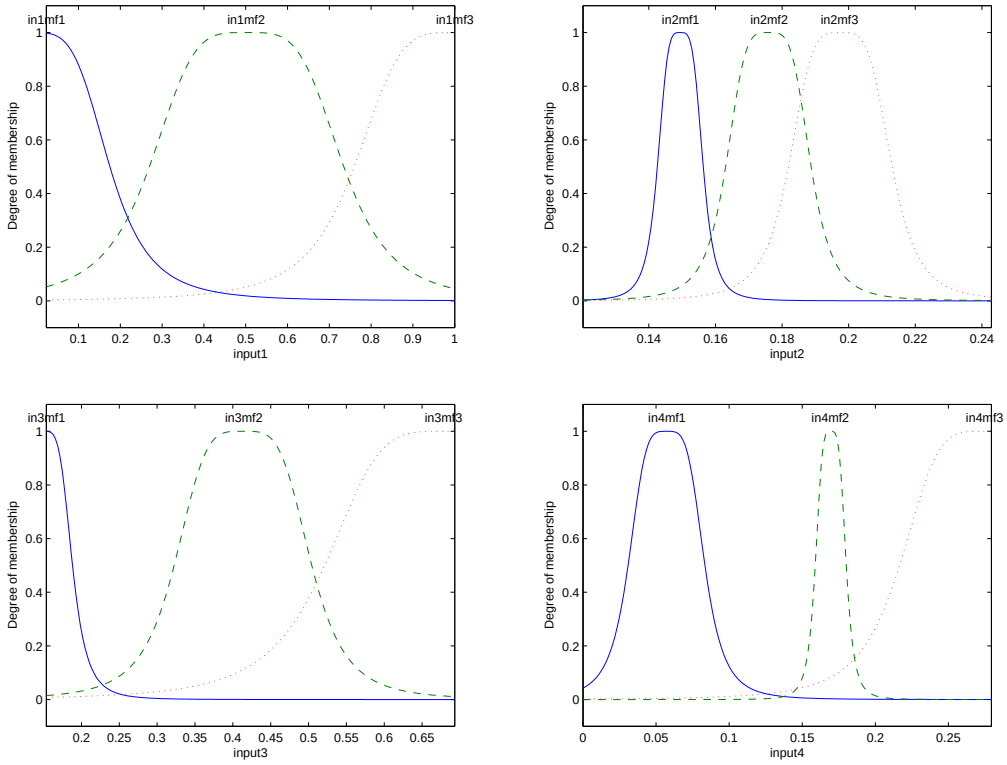

Figure 7.7: Adjusted membership function of ANFIS model in Leukemia Cancer Data Set. Number of selected genes = 4, gene selection method = IG.

Table 7.6: Five rules selected from colon data set by using an individual NF model, when the number of selected genes is 2. If the output value is larger than 0, we consider the output result as cancer.

Rank	Descriptions of Rules
1	If (M63391 is small) and (R87126 is small) then (output is Cancer)
2	If (M63391 is small) and (R87126 is medium) then (output is Cancer)
3	If (M63391 is small) and (R87126 is large) then (output is Normal)
7	If (M63391 is large) and (R87126 is small) then (output is Cancer)
9	If (M63391 is large) and (R87126 is large) then (output is Cancer)

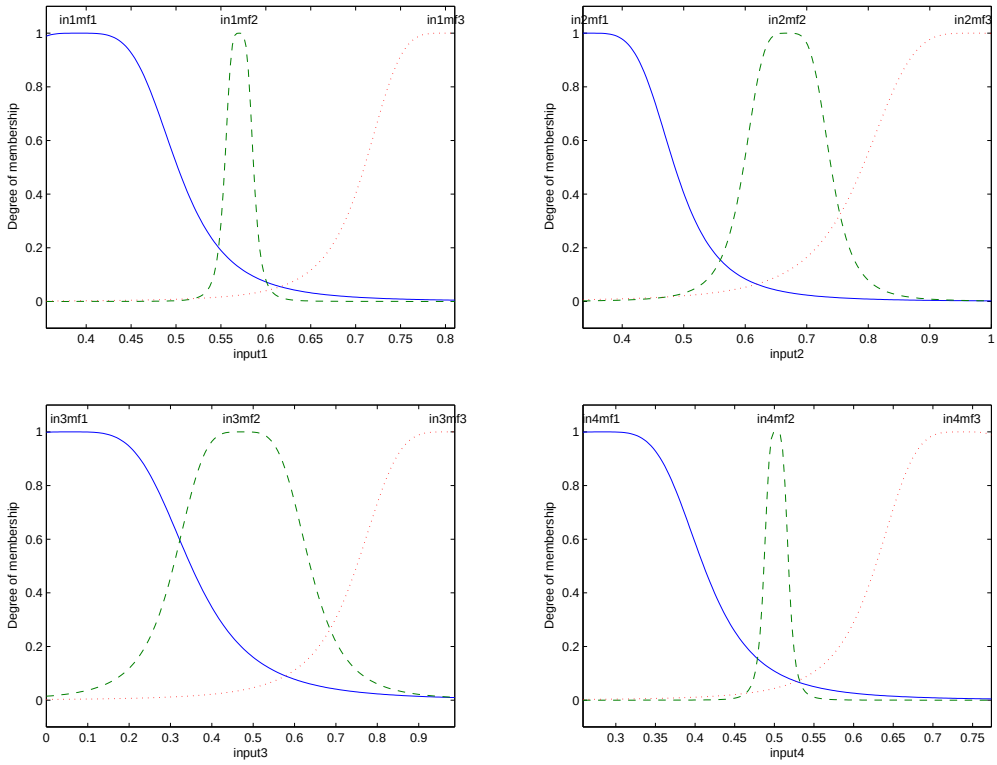

Figure 7.8: Adjusted membership function of ANFIS model in Lymphoma Cancer Data Set, Number of selected genes = 4, gene selection method = IG.

Table 7.7: Five rules selected from leukemia data set by using an individual NF model, when the number of selected genes is 2.

	Descriptions of Rules
1	If (U05259 is small) and (M89957 is small) then (output is Normal)
2	If (U05259 is small) and (M89957 is medium) then (output is Cancer)
3	If (U05259 is small) and (M89957 is large) then (output is Normal)
7	If (U05259 is large) and (M89957 is small) then (output is Cancer)
9	If (U05259 is large) and (M89957 is large) then (output is Normal)

Table 7.8: Five rules selected from Lymphoma data set by using an individual NF model, when the number of selected genes is 2.

	Descriptions of Rules
1	If (GENE3330X is small) and (GENE3332X is small) then (output is Normal)
2	If (GENE3330X is small) and (GENE3332X is medium) then (output is Normal)
3	If (GENE3330X is small) and (GENE3332X is large) then (output is Cancer)
7	If (GENE3330X is large) and (GENE3332X is small) then (output is Cancer)
9	If (GENE3330X is large) and (GENE3332X is large) then (output is Normal)

Table 7.9: Compare the performance of different classifiers in Leukemia data set The results are averaged over 30 runs.

	GSMethod	NoSG	Colon	Leukemia	Lymphomia
Single NF	IG	4	93.55	87.5	87.23
NFE	IG	20	100	95.85	95.65
SVM (Furey *et al*)	IG	50	90.30	94.10	N/A
SVM (Chao *et al*)	SNR	50	65.0	59.0	76.0
	LLE	50	85.0	95.0	91.0
KNN (Jirapech *et al*)	EA	50	N/A	72.64	N/A
C4.5 (Yu *et al*)	ReliefF	4-60	85.48	81.94	N/A

with other approaches. NFE model performance is much better than that of a single NF model on three cancer data sets (see Table 7.9).

NFE model allows us to study a large range of genes. By combining the advantage of ensemble learning, the classification performance is significantly improved. Meanwhile, comparing with a single NF model, less computational cost is required by using NFE model if we implement the model to bigger data sets (the optimal number of selected genes is large). But compared with single NF models, NFE models are more difficult to extract knowledge.

To sum up, the classification performance obtained by our NF and NFE models is very competitive. Different from other approaches, NFE models are more transparent. Their behavior can be explained in human understandable terms, such as linguistic terms and linguistic rules. This provides us with a better under-

standing of the data and gives the researchers and clinicians a clearer explanation on how the diagnose results are given. All these experimental results show that NF and NFE models can be very effective tools for microarray gene expression data classification problems.

Chapter 8

Future Work

This chapter describes the challenge and intended future research directions.

8.1 Future Directions

In this report, we constructed two NF models for cancer microarray data analysis, and tested them on three benchmark data sets. We compared and analysed the models with respect to classification accuracy, number of selected genes, and knowledge extraction. Although the simulation studies show that our models can be good tools for this application, there is still large space to improve. In future work, we will mainly focus on how to use NF-based approaches to overcome three inherent problems of microarray gene expression data, i.e., a lack of available training samples, high dimensional input space, and large amount of noisy and missing values. Theoretical analysis of our models is also necessary.

1. A better single NF model.

Firstly, we can extend our single NF model with respect to the following aspects:

- *A better structure of single NF model.* ANFIS model is just one type of NF combinations. Many other different combination methods can be found in the literature, like GARIC, EFuNN, SONFIN, etc (see Section 2.3.2). Analyzing the behavior of other different NF models is necessary. By combining their strong-points, we hope to build more feasible and efficient models for this application.

Usually, for a certain data set, the optimal structure and specified parameters of the classifier are unknown. By using some optimization techniques, we can dynamically determine the optimal structures and parameters of NF models during the training process, which allows models to better fit the data. Therefore, an Evolutionary Neuro-Fuzzy (ENF) model for cancer microarray analysis is considered for this purpose. Similar work can be found in Evolutionary Neural Networks (ENN) (Manfred & Yee (1998); Xin (1995)), Evolutionary Fuzzy System (Hoffmann (2001)).

- *A more efficient training algorithm.* ANFIS models use a Hybrid BP-LSE algorithm to speed up the training process. But some recent developments in this area promise to provide a better train performance and also a shorter convergence time. Some advanced training algorithms can be introduced to our model, e.g., Optimal BP Algorithm, Conjugate Gradient Algorithm, and Levenberg-Marquardt Algorithm.

- *Multi-class classification problems.* By now we only apply ANFIS to two-class classification problem. But multiple cancer co-existing in one data is very common. How to apply NF models for multi-class classification problems may also require us to design a new structure and training algorithms of NF models.

2. A better multi-NF model.

Multi-classifier techniques have been widely used in microarray analysis, because of their strong abilities in dealing with the problem of a small number of samples and high dimensional input space. We mainly focus on two multi-classifier techniques in our future research, i.e., ensemble methods, and hierarchical methods.

- *Neuro-Fuzzy Ensemble model.* Ensemble learning is a hot topic in current machine learning research. Our current NFE model is a relatively simple ensemble model so far, although it can obtain a good classification performance. We can further improve our NFE with respect to the following aspects:

  - Good generalization ability lies in good model structure. Therefore, structure optimization of NFE can improve the generalization performance further. It includes optimization of the whole ensemble, optimization of the individuals, and optimization of the parameters. Several optimization techniques can be applied here, e.g., genetic algorithms (GA) (Jain *et al.* (1996); Kitano (1990); David (1994)), evolutionary programming (EP) (Xin (1995); Yao & Liu (1997); Curran & O'Riordan (2002)), and particle swarm optimization (PSO) (Settles & Rylander (2002); Bergh & Engelbrecht (2000)).
  - A more complex ensemble training strategy can be used, e.g., Boosting (Hansen & Sakamon (1990)), Negative Correlation Learning(Liu *et al.* (2003)), etc.
  - A more efficient output strategy can be also be considered.
  - As we mentioned previously, a necessary condition for an ensemble of classifiers to be more accurate than any of its individual members is: diversity among individual classifiers. From this point of view, further analysis of NFE model is possible and necessary. more details can be found in (Thomas (2000); Kuncheva & Whitaker (2003); Liu *et al.* (2003)).

- *Hierarchical Neuro-Fuzzy model.* One of biggest problem in the NF modeling for microarray data analysis is the conflict between computational cost and classification accuracy. Some data sets require a large number of genes to present their properties. This number may cause too many rules, which may result in unfeasible computational cost in practice, (see Section 5.2). How to reduce the number of rules is an important issue of our model. Targeting at this problem, we created a NFE model which allows the system to study more informative genes, (see Chapter 6). But some other approaches also can be used, for example, pruning algorithms (Fürnkranz (1997); Reed (1993)), construction algorithms (Wang *et al.* (2004)), hierarchical approaches (Berenji *et al.* (1991)), etc. Hierarchical Neuro-Fuzzy model (HNF) can overcome two main weakneses of current NF models:

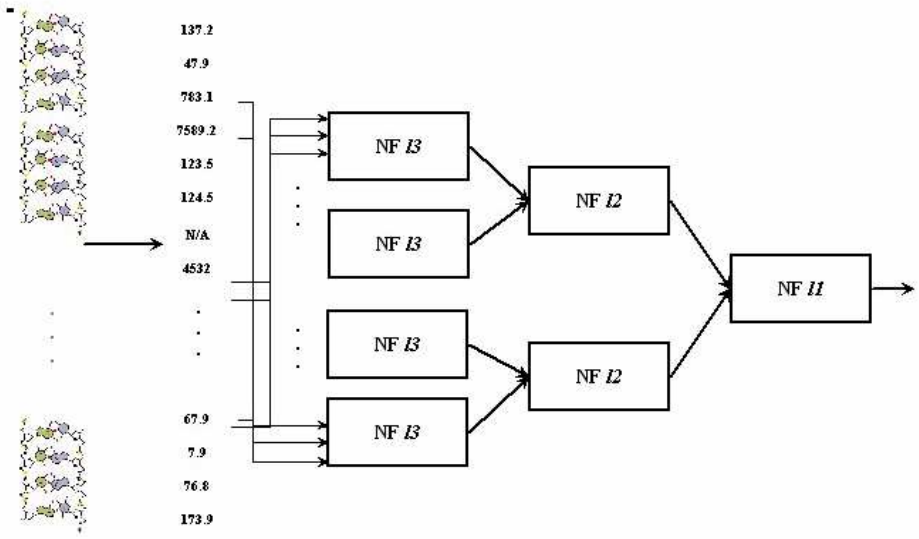

Figure 8.1: The main structure of a possible hierarchical neuro-fuzzy model.

fixed structure and limited number of inputs. It has been proven that a rule-based system using a hierarchical structure leads to a linear growth in the number of rules (Souza *et al.* (2002)). A possible structure of HNF is shown in Figure 8.1. Simulation work will be given in the future, see Timetable.

NFE methods seem to be better at dealing with the problem of a small number of samples available, while HNF methods seem to be better at dealing with the problems of high dimensional space. We can define a simple criterion in analysising cancer gene expression data in later work. If the number of the samples is very small, we prefer to use NFE-Priority methods; If the number of the samples is relatively large, but the number of input is still high, we prefer to use HNF-Priority methods. Or we can combine these two methods.

3. Noisy and missing data

Missing data problem is one of the major problems which restrict the usage of microarray techniques. Microarray experiments often generate expression data arrays with some missing values. Many techniques, such as SVM and MLP, require a complete data matrix to do the analysis. It is important to recognize

the fact that such methods require a complementary pre-processing algorithm to fill in an estimate for the missing data. Various interpolation algorithms can be used for this purpose with varying degrees of success. NF models can deal with this problem easily. We can design some experiments to test our NF model on the data sets with high noise and lots of missing values, and then compare the classification performance with other methods.

4. Fuzzy-based pre-processing techniques

Develop some fuzzy-based pre-processing techniques, for example:

- Fuzzy based gene selection methods. For example, fuzzy C mean method can be used to select the informative genes, similar works can be found in Basak *et al.* (1998), Chakraborty & Pal (2004);

- Fuzzy based gene expression measure. As we mentioned previously in Section 3.1, the gene expression data (G) is measured by computing the log ratio between the two intensities of each dye, shown as follows:

$$G = log_2 \frac{Int(Cy5)}{Int(Cy3)} \tag{8.1}$$

  where $Int(Cy5)$ is the intensity of red color, while $Int(Cy3)$ is the intensity of green color. The collected gene expression data set for analysis is a numerical matrix, which is very difficult for the users to read and understand. If we can ignore this step, and allow NF model to study the data using linguistic red/yellow/green directly, the outcome of the trained model will be in the following form:

  IF the Gene a is red and Gene b is green, THEN output = cancer. (8.2)

  This technique may allow the users to read and apply some trained knowledge directly on the gene image at the early biological research stage. A possible model is presented in Figure 8.2. Each selected gene is classified as red/ yellow/ green by a certain color estimation system. Then a NF model is trained to perform classification task by using these new inputs (red/ yellow/ green). The color estimation system can be a FIS system.

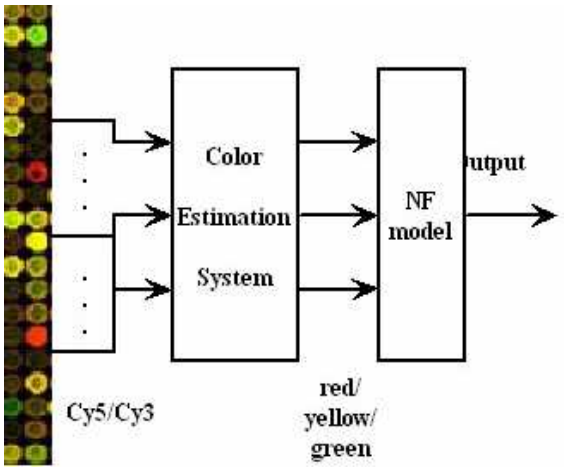

Figure 8.2: A possible NF model by using linguistic red/ yellow/ green as inputs

5. Study the interpretation ability of Neuro-Fuzzy model.

Related biological background knowledge is necessary for this propose. Design some experiments to answer the following two research questions:

- Can we generalize some useful rules for biological researchers, or find the satisfying result to prove their ideas?
- Can we add some knowledge from biological research to our system to see if it can improve the performance?

Possible experiment designs can be:

- 1. Apply NF models for some well-studied small microarray gene expression data sets;
- 2. Add some well-known biological knowledge in form of "IF... THEN..." rules to one model;
- 3. Compare the performance of the model with prior knowledge and the model without prior knowledge.

6. Computational complexity

Theoretical analysis of above models is necessary. Using NF method for microarray cancer gene data classification can be looked as a "NP-complete" problem (Ezziane (2006); Garey *et al.* (1974)). By studying the computational complexity of our model, we hope to found out the efficiency of our systems, and the inherent "difficulty" of problems which we are trying to resolve.

7. GeneNeFu

We plan to develop a software package, called GeneNeFu, which can be used for cancer gene expression data classification. The outputs of the systems include both classification results and some trained rules in human linguistic terms. Prior knowledge can also be easily embedded into the model in order to improve the classification performance further. And it also can help biologists to test-proof their idea. The package will include six modules:

- Gene Selection Module;
- Single Neuro-Fuzzy Module;
- Evolutionary Neuro-Fuzzy Module;
- Neuro-Fuzzy Ensemble Module;
- Hierarchical Neuro-Fuzzy Module;
- Hierarchical Neuro-Fuzzy Ensemble Module.

Different modules target different user requirements from the users. Gene selection module is used to select informative genes. Single neuro-fuzzy module and evolutionary neuro-fuzzy module target the users who are interested to know some useful knowledge behind the data more than the classification results. Evolutionary neuro-fuzzy module may give a better output than single neuro-fuzzy module, but it may take more computational time. Neuro-fuzzy ensemble module and hierarchical neuro-fuzzy module target the users who are more interested to know the classification results rather than hidden knowledge. While neuro-fuzzy ensemble module is suggested to be used when the number of available samples is small, hierarchical neuro-fuzzy module is suggested to be used when the number of available samples is relatively large. Hierarchical neuro-fuzzy ensemble module

is a combination of neuro-fuzzy ensemble module and Hierarchical neuro-fuzzy module. A friendly users' interface is also required.

8.2 Timetable

Table 8.1: Timetable for Year 2006 - Year 2007

Time	Research Problems	Outcomes
Hilary 2006	A. Summarize current work B. Optimize the structure of single NF models	a. Paper Submitted
Easter 2006	A. Develop new Gene Selection methods (GS)	a. Simulation work (GS module done)
Trinity 2006	A. Develop advanced learning algorithms for single NF models. B.Develop an Evolutionary NF model (ENF)	a1. Simulation work (NF module done) a2. Submit paper b1. Simulation work (ENF Module done) b2. Submit paper
Summer 2006	A. Optimize a new NF Ensemble model (NFE)	a. Simulation work (NFE Module done)
Michaelmas 2007	A. Develop a Hierarchical NF model (HNF)	a1. Simulation work (HNF Module done) a2. Submit paper
Hilary 2007	A. Develop a Hierarchical NF Ensemble Module (HNFE) B. Study at interpretation ability of models	a1. Simulation work (HNFE Module done) a2. Submit paper B1. Simulation work
Easter 2007	A. Develop users' interface of GeneNeFu B. Develop users' guide document of GeneNeFu	a. GeneNeFu
Trinity 2007-Summer 2007	A. Thesis written up	a. Final Phd Thesis

Chapter 9

Conclusions

In this report, we have constructed two NF models for the classification and analysis of highly dimensional cancer microarray data. In this chapter we recap the work described in the above chapters and draw some conclusion.

Microarray data analysis is a novel and hot topic in current biological and medical research, especially when using this technology for cancer diagnoise. But it is also a big challenge for today's machine learning research. Firstly, we introduced some basic notions of microarray technology, microarray gene expression data, and described how to use microarray gene expression data for cancer classification. We formalized this problem as a high-dimension-low-sample data set with lots of noisy/missing data classification problem in machine learning.

According with our knowledge, we applied the ANFIS model for cancer microarray data classification problem for the first time. Then we tested this model on three benchmark cancer microarray data set, including colon cancer data set, leukemia cancer data set, and lymphoma cancer data set. The simulation work shows that the ANFIS model can obtain very good classification performance as other widely used models. But ANFIS only allows the system to use a small number of genes due to it's high computational cost. Targeting at this problem, we proposed a new NF model, called Neuro-Fuzzy Ensemble model (NFE), which uses a group of ANFIS models to learn different parts of the data, and then combine the classification results from different individual classifiers. The experimental results showed that NFE allowed the classifier to cope a larger data set, and better accuracy was obtained on most tested data sets. The reason of

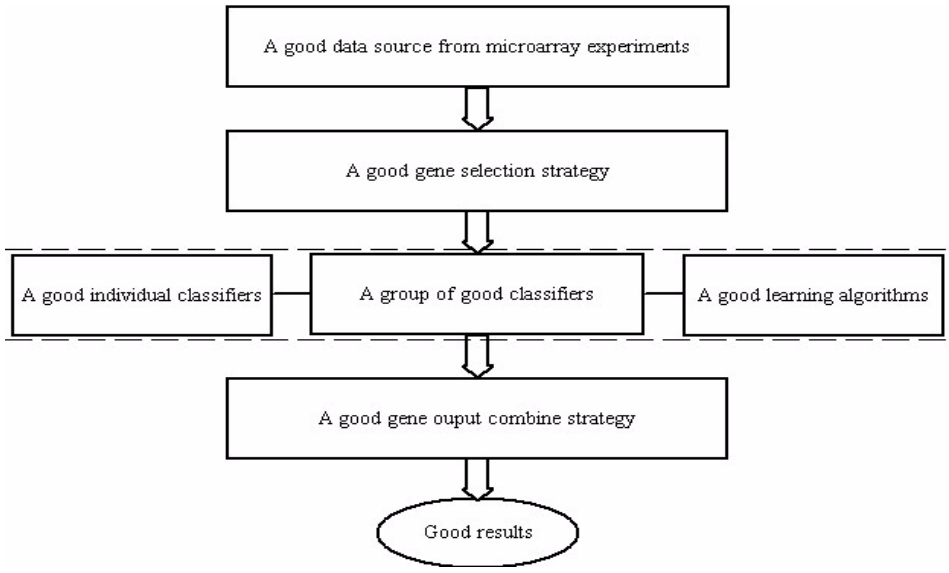

Figure 9.1: The structure of a good NF model for microarray gene expression data classification

why NFE performed better and more efficient with respect to training time is also given.

We also introduced some important related notions and techniques about microarray data analysis. Some most popular models, such as SVM, KNN, and MLP, were introduced and compared with our NF models. Two efficient genes selection methods were adopted in our experimental study, i.e., Information Gain (IG) and signal to Noise Ratio (SNR). LOOCV accuracy were used to evaluate the effectiveness of the classifiers, because traditional strategies could not well distinguish the ability of different classifiers by using such a small number of training patterns.

From previous work and our experimental results, we can conclude that: A good model for microarray gene expression data classification should contain the following six parts: A good gene expression data source from biological experiments; an efficient gene selection strategy; an efficient individual classifier, a reasonable training algorithm, a good classifier combination, and a good strategy to process the outputs of individual classifiers, shown in Figure 9.1. Future work

will focus on improving our models with respect to those aspects (the data source part is excepted).

Appendix A

Top Selected Genes

Top 50 ranked genes from colon cancer data set, leukemia cancer data set, and lymphoma cancer data set, are selected for future usage.

Table A.1: Top 50 ranked genes from colon cancer data set, leukemia cancer data set, and lymphoma cancer data set, are selected by IG method

	Colon		Leukemia		Lymphoma	
Rank	ID	Gene name	ID	Gene name	ID	Gene name
1	249	M63391	4050	X03934	1279	GENE3330X
2	493	R87126	6510	U23852	1281	GENE3332X
3	1042	R36977	4342	X59871	1277	GENE3328X
4	1423	J02854	4055	X04145	1247	GENE3355X
5	267	M76378	5542	M37271	1291	GENE3314X
6	245	M76378	5543	M37271	2244	GENE1252X
7	399	U30825	5466	X58072	1206	GENE3228X
8	1772	H08393	6606	X00437	1316	GENE3256X
9	765	M76378	1694	M12886	1287	GENE3338X
10	467	H40560	6696	X76223	3861	GENE1720X
11	822	T92451	1893	M28826	2439	GENE3968X
12	258	M16937	2833	U16954	2263	GENE1296X
13	66	T71025	4357	X60992	2243	GENE1251X
14	1067	T70062	4847	X95735	1634	GENE2704X
15	1325	T47377	1106	J04132	1616	GENE2662X
16	62	T48804	3332	U50743	2415	GENE3987X
17	377	Z50753	6236	U83239	2200	GENE1207X
18	1892	U25138	4484	X69398	34	GENE4006X
19	1808	U21090	4291	X56468	3137	GENE392X
20	1582	X63629	2454	S65738	2417	GENE3985X
21	652	R10066	6228	M26692	3860	GENE1719X
22	47	T58861	6575	U49835	3421	GENE3820X

	Colon		Leukemia		Lymphoma	
Rank	ID	Gene name	ID	Gene name	ID	Gene name
23	1002	R08183	6180	M16336	1312	GENE3258X
24	1843	H06524	2433	S34389	2207	GENE1214X
25	1247	X74295	1630	L47738	1282	GENE3333X
26	415	T60155	2794	U14603	3018	GENE461X
27	1790	U10362	6732	X89399	1274	GENE3325X
28	625	X12671	6462	X73358	2438	GENE3967X
29	515	T56604	5567	D13720	2437	GENE3966X
30	1494	X86693	4061	X04391	35	GENE4007X
31	43	T57619	2324	M88108	3015	GENE464X
32	26	T95018	6126	L40386	2593	GENE867X
33	513	M22382	3594	U67171	675	GENE2297X
34	444	T59878	6516	U26312	36	GENE4008X
35	440	M94556	5191	Z69881	64	GENE3940X
36	360	L09604	4017	X00274	2435	GENE51X
37	127	T51529	1541	L38696	683	GENE2291X
38	1897	U19969	3969	U93049	2137	GENE3923X
39	1635	M36634	6571	U47686	1017	GENE2105X
40	1411	H77597	4133	X13482	276	GENE3760X
41	897	H43887	976	HG4128	2845	GENE740X
42	1326	M94203	2061	M59807	2205	GENE1212X
43	1058	M80815	5130	Z35227	1144	GENE3214X
44	1060	U09564	3307	U49395	809	GENE2106X
45	413	H25136	1268	L10373	1337	GENE3240X

	Colon		Leukemia		Lymphoma	
Rank	ID	Gene name	ID	Gene name	ID	Gene name
46	384	T56940	6168	M13560	3892	GENE1543X
47	138	M26697	5995	X52979	3149	GENE374X
48	72	H55758	4156	X14975	1633	GENE2703X
49	15	U14971	6555	X93511	704	GENE2326X
50	1900	X56597	1373	L19686	213	GENE3512X

Appendix B

Lectures, Meetings, Seminars and Teaching

Lecture Courses Attended

- Prof. Samson Abramsky, Intelligent System I. This course introduced some fundamental issues in intelligent system: Search, logic, knowledge, representation and reasoning. It also included 10 classes, 4 assignments, 1 Prolog practical and 1 examination.

- Dr Ian Sobey, Practical Numerical Analysis. This course covered a range of numerical analysis topics: interpolation, integration, gobal optimization, finite difference methods, iterative solvers, and finite element method. It includes 8 classes, 6 assignments.

- Dr Stephen Cameron, Intelligent System II. Topics covered uncertain information, timeliness, and inexact control that are found when dealing with

embodied agents (e.g., robots). It includeed 10 lectures, 4 assignments, and 1 examination.

- Dr Vasile Palade, Machine Learning for ECS. Topics covered decision and regression trees, learning using neural networks, probabilistic modeling and fuzzy systems. It included 4 classes, and 1 practical report.

Project Meetings

Weekly meetings with my supervisor, Dr. Vasile Palade.

Seminars and Reading Groups

I have attended about 10 seminars each term in both Computing Laboratory and Department of Engineering Science. Since May, 2005, I have attended Bioinformatics Discussion Group in Welcome Trust Centre for Genetics (WTCHG).

Teaching

- Michaelmas Term 2004: Demonstrator for Computing Design-Built-Test;
- Michaelmas Term 2004: Demonstrator for Object-Oriented Programming;
- Hilary Term 2005: Demonstrator for Computing Design-Built-Test;
- Michaelmas Term 2005: Demonstrator for Computing Design-Built-Test;
- Michaelmas Term 2005: Demonstrator for Compilers;
- Michaelmas Term 2005: Marker for Intelligent Systems I.

Appendix C

Related Publications

Bibliography

Al-Jarrah, O. & Halawani, A. (2001). Recognition of gestures in arabic sign language using neuro-fuzzy systems. *Artif. Intell.*, **133**, 117–138. 1.1

Alberts, B., Johnson, A., Lewis, J., Roberts, K., Raff, M. & Walter, P. (2002). *Molecular biology of the cell*. Garland publishing. 3.1

Alon, U., Barkai, N., Notterman, D., Gish, K., Ybarra, S., Mack, D. & Levine, A. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natnl. Acad. Sci. USA.*, **96**, 6745–6750. 1, 1.1

Basak, J., De, R.K. & Pal, S.K. (1998). Unsupervised feature selection using a neuro-fuzzy approach. *Pattern Recogn. Lett.*, **19**, 997–1006. 8.1

Baumgartner, R., Windischberger, C. & Moser, E. (1998). Quantification in functional magnetic resonance imaging: fuzzy clustering vs. correlation analysis. *Magn Reson Imaging*, **16**, 115–25. 1

Belal, S., Taktak, A., Nevill, A., Spencer, S., Roden, D. & Bevan, S. (2002). Automatic detection of distorted plethysmogram pulses in neonates and

paediatric patients using an adaptive-network-based fuzzy inference system. *Artificial Intelligence in Medicine*, 149C165. 5, 5.2

Ben-Dor, A., Bryhn, L., Friedman, N., nachman, I., Schummer, M. & Yakhini, Z. (2000). Tissue classification with gene expression profiles. *Journal of Computational Biology*, **4**, 290–2301. 4.2

Bennett, K.P. & Campbell, C. (2000). Support vector machines: hype or hallelujah? *SIGKDD Explor. Newsl.*, 1–13. 4.3.2

Berenji, H.R., Chen, Y.Y., Lee, C.C., Jang, J.S. & Murugesan, S. (1991). A hierarchical approach to designing approximate reasoning-based controllers for dynamic physical systems. In P.B. Bonissone, M. Henrion, L.N. Kanal & J.F. Lemmer, eds., *Uncertainty in Artificial Intelligence 6*, 331–343, North-Holland, Amsterdam. 1.1, 8.1

Bergh, F. & Engelbrecht, A.P. (2000). Cooperative learning in neural networks using particle swarm optimizers. *SACJ / SART*, 224–226. 2.2.1, 8.1

Berkan, R.C. & Trubatch, S. (1997). *Fuzzy System Design Principles*. Wiley-IEEE Press. 2.1.3

Bishop, C.M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford. 1.1, 2, 2.2.1, 2.2.1

Boser, B., Guyon, I. & Vapnik, V. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning*. 4.3.2

Brändle, N., Bischof, H. & Lapp, H. (2003). Robust dna microarray image analysis. *Mach. Vision Appl.*, **15**, 11–28. 3.1

Bressan, M. & Vitri;, J. (2003). Nonparametric discriminant analysis and nearest neighbor classification. *Pattern Recogn. Lett.*, **24**, 2743–2749. 4.3.3

Brown, M.P.S., Grundy, W.N. & Lin, D. (2000). Knowledge-based analysis of microarray gene expression data using support vector machines. In *Proceedings of National Academy of Science*, 262–267. 1, 4.3, 4.3.2

Burgess, N. (1994). A constructive algorithm that converges for real-valued input patterns. *Neural Systems*, **5**, 59–66. 1

Carmona, E., Mira, J., Feijoo, J.G. & Rosa, M.G. (2001). Neuro-fuzzy nets in medical diagnosis: The diagen case study of glaucoma. In *IWANN '01: Proceedings of the 6th International Work-Conference on Artificial and Natural Neural Networks*, 401–409, Springer-Verlag, London, UK. 1.1

Chakraborty, D. & Pal, N.R. (2004). A neuro-fuzzy scheme for simultaneous feature selection and fuzzy rule-based classification. *IEEE Transactions on Neural Network*, **15**, 110–123. 8.1

Cho, S. & Won, H. (2003). Machine learning in dna microarray analysis for cancer classification. In *APBC*, vol. 34, 189–198. 3.1, 7.1.1

Curran, D. & O'Riordan, C. (2002). Applying evolutionary computation to designing neural networks: A study of the state of the art. 2.2.1, 8.1

David, W.W. (1994). *GANNet: A genetic algorithm for searching topology and weight spaces in neural network design*. Phd thesis, University of Maryland College Park. 2.2.1, 8.1

Dettling, M. (2004). *Supervised learning in very high dimensional problems with application to microarray data*. Phd thesis, Swiss Federal Institute of Technology Zurich. 3.1

Diaz-Verdejo, J.E., Herreros, A.P., Luna, J.C.S., Ortüzar, M.C.B. & Ayuso, A.R. (1991). Recurrent neural networks for speech recognition. In *IWANN*, 361–369. 2.2.1

Ding, C.H.Q. & Peng, H. (2003). Minimum redundancy feature selection from microarray gene expression data. In *CSB*, 523–529. 1, 5.3

Dubitzky, W., Granzow, M., Downes, C.S. & Berrar, D. (2002). *Introduction to Microarray Data Analysis*. Kluwer Academic Publishers, Boston/Dordrecht/London. 3.1, 3.1

Ezziane, Z. (2006). Dna computing: applications and challenges. *Nanotechnology*, 27–39. 8.1

Feng, J.C. & Teng, L.C. (1998). An online self constructing neural fuzzy inference network and its applications. *IEEE Transactions on Fuzzy Systems*, **6**, 12–32. 2.3.2

Furey, T.S., Cristianini, N., Duffy, N., Bednarski, D.W., Schummer, M. & Haussler, D. (2000). Support vector machine classification and valida-

tion of cancer tissue samples using microarray expression data. *Bioinformatics*, **16**, 906–914. 1

Fürnkranz, J. (1997). Pruning algorithms for rule learning. *Machine Learning*, **27**, 139–171. 1.1, 1, 8.1

Garey, M.R., Johnson, D.S. & Stockmeyer, L. (1974). Some simplified np-complete problems. In *STOC '74: Proceedings of the sixth annual ACM symposium on Theory of computing*, 47–63, ACM Press, New York, NY, USA. 8.1

Garzon, M., Ankaraju, P., Drumwright, E. & Kozma, R. (2002). Neurofuzzy recognition and generation of facial features in talking heads. In *Proc. of the IEEE Intl. Conf. on Fuzzy Systems (FUZZ-IEEE)*, Computer Society Press, Hawaii, USA. 5

Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D. & Lander, E.S. (2002). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, **286**, 531–537. (document), 1, 1.1, 3.4, 3.2, 4.2, 4.2.2, 4.2.2, 7.1.2

Gregory, P.S. & Pablo, T. (2003). Microarray data mining: facing the challenges. *SIGKDD Explorations*, **5**, 1–5. 1, 3.1, 3.2, 3

Guthke, R., Schmidt-Heck, W., Hahn, D. & Pfaff, M. (2002). Gene expression data mining for functional genomics using fuzzy technology. In *Advances in Computational Intelligence and Learning: Methods and Applications*, 475–487, Kluwer, B.V. 1

Hagan, M.T., Demuth, H.B. & Beale, M. (1996). *Neural Network Design*. PWS Publishing Company. 2.2.1, 2.2.1

Hansen, L.K. & Sakamon, P. (1990). Neural network ensemble. *IEEE on pattern analysis and machine intelligents*, **12**, 456–472. 1.1, 6.1, 8.1

Haykin, S. (1994). *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, Upper Saddle River, NJ, USA. 2

Haykin, S. (1996). *Neural Network: A Comprehensive Foundation.*. Macrnillan Company, New York. 6.1

Hegyi, A., De Schutter, B., Babuška, R., Hoogendoorn, S., van Zuylen, H. & Schuurman, H. (2000). A fuzzy decision support system for traffic control centers. In *Proceedings of the TRAIL 6th Annual Congress 2000 — Transport, Infrastructure and Logistics,* Part 2, The Hague/Scheveningen, The Netherlands. 2.1.3

Hertz, J., Krogh, A. & Palmer, R.G. (1991). *Introduction to the theory of neural computation*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA. 2

Hoffmann, F. (2001). Evolutionary algorithms for fuzzy control system design. 8.1

Holmes, C. (2002). A probabilistic nearest neighbour method for statistical pattern recognition. *Jounal Roy. Statist. Soc. B*, 295–306. 4.3.3

Hwang, K.B., Cho, D.Y., Park, S.W., Park, S.D. & Zhang, B.T. (2002). Applying machine learning techniques to analysis of gene expression data: can-

cer diagnosis. In S. Lin & K. Johnson, eds., *Methods of Microarray Data Analysis*, 167–182, Kluwer Academic. 1

Inza, I., Larrañaga, P., Blanco, R. & Cerrolaza, A.J. (2004). Filter versus wrapper gene selection approaches in dna microarray domains. *Artificial Intelligence in Medicine*, 91–103. 4.2

Jain, L., Johnson, R. & van Rooij, A. (1996). *Neural Network Training Using Genetic Algorithms*. World Scientific. 2.2.1, 8.1

Jang, J.S.R. (1992). *Neuro-Fuzzy Modeling: Architectures, Analyses and Application*. Phd thesis, University of California Berkeley. 1, 2.1.1, 2.1.2, 5, 5.1

Jang, J.S.R. & Gulley, N. (1997). *MATLAB/Fuzzy Logic Toolbox, MathWorks*. Inc. Natick, MA. 2.1.3, 2.3.2

Jang, J.S.R. & Sun, C.T. (1995). Neuro-fuzzy modeling and control. *Proceedings of the IEEE*. (document), 1, 1.1, 2.1.3, 2.3, 2.1.3, 5

Jang, J.S.R. & Sun, C.T. (1997). *Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA. (document), 1, 2, 5, 5.1

Jiang, X.R. & Gruenwald, L. (2005). Microarray gene expression data association rules mining based on bsc-tree and fis-tree. *Data Knowl. Eng.*, **53**, 3–29. 1

Jirapech-Umpai, T. & Aitken, S. (2005). Feature selection and classification for microarray data analysis: Evolutionary methods for identifying predictive genes. *Bioinformatics*, **6**, 168–174. 4.3

KANDEL, A., ed. (1987). *Fuzzy Expert Systems*. Addison–Wesley, Reading, MA. 2

KASABOV, N. & SONG, Q. (1999). Dynamic evolving fuzzy neural networks with 'm-out-of-n' activation nodes for on-line adaptive systems. Technical TR99/04, Department of information science University of Otago. 2.3.2

KHAN, J., WEI, J.S., RINGNER, M., SAAL, L.H., LADANYI, M., WESTERMANN, F., BERTHOLD, F., SCHWAB, M., ANTONESCU, C.R., PETERSON, C. & MELTZER, P.S. (2001). Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine*, **7**, 673–679. 1, 1.1, 4.3

KIM, H.C., PANG, S., JE, H.M., KIM, D. & BANG, S.Y. (2002). Pattern classification using support vector machine ensemble. In *ICPR (2)*, 160–163. 1.1

KITANO, H. (1990). Designing neural networks using genetic algorithms with graph generation system. *Complex Systems*, 461–476. 2.2.1, 8.1

KOHONEN, T., ed. (1997). *Self-organizing maps*. Springer-Verlag New York, Inc., Secaucus, NJ, USA. 1

KOSKO, B. (1992). *Neural networks and fuzzy systems: a dynamical systems approach to machine intelligence*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA. 2.3.1

Krishnapuram, B. & et al. (2004). Joint classifier and feature optimization for comprehensive cancer diagnosis using gene expression data. *Journal of Computational Biology*, 227–242. 4.2

Kuncheva, L.I. & Whitaker, C. (2003). Measure of diversity in classifer ensemble. *Machine learning*, **51**, 181–207. 1.1, 6.1, 8.1

Li, L., Weinberg, C.R., Darden, T.A. & Pedersen, L.G. (2001). Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the ga/knn method. *Bioinformatics*, **17**, 1131–1142. 1, 4.2, 4.3

Lin, C.T. & Lee, C.S.G. (1991). Neural network based fuzzy logic control and decision system. *IEEE Teansaction on Comput.*, **40**, 1320–1336. 2.3.2

Liu, Y., Yao, X. & Higuchi, T. (2003). Evolutionary ensemble with negative correlation learning. *IEEE Transactions on Evolutoinary Computation*, **4**, 380–387. 6.1, 8.1

Lossos, I.S., Alizadeh, A.A., Eisen, M.B., Chan, W.C., Brown, P.O., Bostein, D., Staudt, L.M., & Levy, R. (2000). Ongoing immunoglobulin somatic mutation in germinal center b cell-like but not in activated b cell-like diffuse large cell lymphomas. In *Proc. of the Natl. Acad. of Sci. USA*, vol. 97, 10209–10213. 7.1.3

Mamdani, E. & Assilina, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, **7**, 1–13. 2.1.2

Manfred, F. & Yee, L. (1998). A genetic-algorithms based evolutionary computational neural network for modelling spatial interaction data. *Annals of Regional Science*, **32**, 437–458. 2.2.1, 8.1

Mitchell, T. (1997). *Machine learning*. McGraw-Hill, New York. 1, 4.2.1

Molla, M., Waddell, M., Page, D. & Shavlik, J. (2004). Using machine learning to design and interpret gene-expression microarrays. *AI Mag.*, **25**, 23–44. 1

Nauck, D. (1997). Neuro-fuzzy systems: Review and prospects. Proc. Fifth European Congress on Intelligent Techniques and Soft Computing (EUFIT'97). 2, 2.3.2, 5.1, 5.2

Nauck, D. & Kruse, R. (1997). Neuro-fuzzy systems for function approximation. In *4th International Workshop Fuzzy-Neuro Sy stems*. 2.3.2

Nauck, D., Klawonn, F., Kruse, R. & Klawonn, F. (1997). *Foundations of Neuro-Fuzzy Systems*. John Wiley & Sons, Inc., New York, NY, USA. 2, 1

Newton, R. & Xu, Y. (1993). Neural network control of a space manipulator. *IEEE Control Systems*, **13**, 14–22. 2.2.1

Nguyen, H.S., Szczuka, M.S. & Slezak, D. (1997). Neural networks design: Rough set approach to continuous data. In *PKDD '97: Proceedings of the First European Symposium on Principles of Data Mining and Knowledge Discovery*, 359–366, Springer-Verlag, London, UK. 2

Ohno-Machado, L., Vinterbo, S.A. & Weber, G. (2002). Classification of gene expression data using fuzzy logic. *Journal of Intelligent and Fuzzy Systems*, **12**, 19–24. 1

Palade, V., Patton, R.J., Uppal, F., Quevedo, J. & Daley, S. (2002). Fault diagnosis of an industrial gas turbine using neuro-fuzzy methods. In *Proc. of the 15th IFAC World Congress*, 2477–2482. 1.1

Paul, S. & Kumar, T. (2004). Gene expression based cancer classification using evolutionary and non-evolutionary methods. Technical 041105A1, Department of Frontier Informatics The University of Tokyo. 3.1

Paul, T.K. & Iba, H. (2004). Selection of the most useful subset of genes for gene expression-based classification. In *Proceedings of the IEEE Congress on Evolutionary Computation*, 2076–2083, Portland USA. 4.2

Peng, Y. & Flach, P. (2001). Soft discretization to enhance the continuous decision tree induction. In C. Giraud-Carrier, N. Lavrac & S. Moyle, eds., *Integrating Aspects of Data Mining, Decision Support and Meta-Learning*, 109–118, ECML/PKDD'01 workshop notes. 1

Prabakaran, S., Sahu, R. & S.Verma (2005). Genomic signal processing using micro arrays, submitted to hybrid system. 1, 3.1, 3.1, 3.1, 5, 4.2

Reed, R. (1993). Pruning algorithms - a survey. *IEEE Trans. Neural Networks*, **4**, 740–747. 1.1, 1, 8.1

Ressom, H., Reynolds, R. & Varghese, R.S. (2003). Increasing the efficiency of fuzzy logic-based gene expression data analysis. *Physiol. Genomics*, **13**, 107–117. 1

Ringhut, E. & Kooths, S. (2003). Modeling expectations with genefer c an artificial intelligence approach. *Comput. Econ.*, **21**, 173–194. 1.1

Ripley, R.M. (1998). *Neural Network Models for Breast Cancer Prognosis*. Phd thesis, Oxford University. 2.2.1

Rogers, S. (2004). *Machine learning techniques for microarray analysis*. Ph.D. thesis, University of Bristol, UK. 4.2

Sambrook, J., Fritsch, E.F. & T. Maniatis, T. (1989). *Cloning: A Laboratory Manual*. Cold Spring Harbor Laboratory Press, New York, USA. 3.1

Schaffer, C. (1993). Selecting a classification method by cross-validation. *Machine learning*, **13**, 135–143. 5.3

Schmidt, T.H. (1989). Model-based fault diagnosis: knowledge acquisition and system design. In *IEA/AIE '89: Proceedings of the 2nd international conference on Industrial and engineering applications of artificial intelligence and expert systems*, 21–25, ACM Press, NY, USA. 2.1.3

Settles, M. & Rylander, B. (2002). Neural network learning using particle swarm optimizers. *Advances in Information Science and Soft Computing,*, 224–226. 2.2.1, 8.1

Shi, C. & Chen, L. (2005). Feature dimension reduction for microarray data analysis using locally linear embedding. In *APBC*, 211–217. 4.3

Slonim, D.K., Tamayo, P., mesirov, J.P., Golub, T.R. & Lander, E.S. (2000). Class prediction and discovery using gene expression data. In *RECOMB*, 263–272. 1, 1.1, 4.2, 4.2.2

Souza, F.J., Vellasco, M.M.R. & Pacheco, M.A.C. (2002). Hierarchical neuro-fuzzy quadtree models. *Fuzzy Sets Syst.*, **130**, 189–205. 8.1

Sugeno, M. & Kang, G.T. (1988). Structure identification of fuzzy model. *Fuzzy Sets and Systems*, **26**, 15–33. 2.1.2, 5.1

Sulzberger, S.M., Tschicholg-Gurman, N.N. & Vestli (1993). Fun: Optimization of fuzzy rule based systems using neural networks. In *Proceedings of IEEE Conference on Neural Networks*, 312–316. 2.3.2

Sulzberger, S.M., Tschicholg-Gurman, N.N. & Vestli, S.J. (1998). Optimization of fuzzy rule based systems using neural network. *IEEE Transactions on NN*, **9**, 83–105. 2.3.2

Takagi, T. & Sugeno, M. (1985). Fuzzy identification of systems and its applications to modelling and control. *IEEE Transactions on Systems, Man and Cybernetics*, **SMC-15**, 116–132. 2.1.2, 5.1

Tan, A.C. & Gilbert, D. (2003). Ensemble machine learning on gene expression data for cancer classification. *Applied Bioinformatics*, **2**, S75CS83. 1

Tano, S., Oyama, T. & Arnould, T. (1996). Deep combination of fuzzy inference and neural network in fuzzy inference. *Fuzzy Sets and Systems*, **82**, 151–160. 2.3.2

TAYLOR, J.G. (1993). *The Promise of Neural Networks*. Springer-Verlag New York, Inc., Secaucus, NJ, USA. 2.2.1

TESCHIONI, A., OBERTI, F. & REGAZZONI, C.S. (1999). A neural network approach for moving objects recognition in color image sequences for surveillance applications. In *NSIP*, 28–32. 2.2.1

THE-GENOME-INTERNATIONAL-SEQUENCING-CONSORTIUM (2001). Initial sequencing and analysis of the human genome. *Nature,*, **409**, 860–921. 1

THOMAS, G.D. (2000). Ensemble methods in machine learning. In *Proc. of the first International Workshop on Multiple Classifier System (MCS 2000)*, 1–15. (document), 6.1, 6.1.1, 6.3, 8.1

TRUSSELL, H.J. & VRHEL, M.J. (1999). Modelling of color scanners using neural networks. In *NSIP*, 19–22. 2.2.1

VALPOLA, H. & KARHUNEN, J. (2002). An unsupervised ensemble learning method for nonlinear dynamic state-space models. *Neural Comput.*, 2647–2692. 1.1

VAPNIK, V. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, USA. 4.3.2

VAPNIK, V. (1999). *Statistical Learning Theory*. John Wiley and Sons, New York, USA. 4.3.2

VAROGLU, E. & HACIOGLU, K. (1999). A recurrent neural network speech predictor based on dynamical systems approach. In *NSIP*, 316–320. 2.2.1

VENTER, J. (2001). The sequence of the human genome. *Science*, **291**, 1304C1351. 1

VIRANT-KLUN, I. & VIRANT, J. (1999). Fuzzy logic alternative for analysis in the biomedical sciences. *Computers and Biomedical Research*, **32**, 305C321. 5

WANG, Z., YAO, X. & XU, Y. (2004). An improved constructive neural network ensemble approach to medical diagnoses. In *Proc. of the Fifth International Conference on Intelligent Data Engineering and Automated Learning (IDEAL'04)*, vol. 3177, 572–577, Lecture Notes in Computer Science, UK. 2.2.1, 1, 8.1

WOLPERT, D. & MACREADY, W. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, **1**, 67–82. 2.1.3

XIN (1995). Evolutionary artificial neural networks. In A. Kent & J.G. Williams, eds., *Encyclopedia of Computer Science and Technology*, vol. 33, 137–170, Marcel Dekker Inc. 2.2.1, 1, 8.1

XING, E.P., JORDAN, M.I. & KARP, R.M. (2001). Feature selection for high-dimensional genomic microarray data. In *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*, 601–608, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. 4.2.1

XIONG, M., LI, W., ZHAO, J., JIN, L. & BOERWINKLE, E. (2001). Feature (gene) selection in gene expression-based tumor classification. *Molecular Genetics and Metabolism*, 239–247. 4.2

Xu, Y., Selaru, M., Yin, J., Zou, T.T., Shustova, V., Mori, Y., Sato, F., Liu, T.C., Olaru, A., Wang, S., Kimos, M.C., Perry, K., Desai, K., Greenwood, B.D., Krasna, M.J., Shibata, D., Abraham, J.M. & Meltzer, S.J. (2002). Artificial neural networks and gene filtering distinguish between global gene expression profiles of barrett's esophagus and esophageal cancer. *Cancer Research*, **62**, 3493–3497. 1, 4.3

Yao, X. & Liu, Y. (1997). A new evolutionary system for evolving artificial neural networks. *IEEE Transactions on Neural Networks*, **8**, 694–713. 2.2.1, 1, 8.1

Yeung, K.Y., Bumgarner, R.E. & Raftery, A.E. (2005). Bayesian model averaging: development of an improved multi-class, gene selection and classification tool for microarray data. *Bioinformatics 21(10):*, **21**, 2394–2402. 4.2

Yu, L. & Liu, H. (2004). Redundancy based feature selection for microarray data. Technical, Department of Computer Science and Engineering Arizona State University. 4.2.1

Zadeh, L. (1965). Fuzzy sets. *Information and Control*, **3**, 338–353. 1, 2, 2.1.1

Zadeh, L. (1968). Fuzzy algorithms. *Information and Control*, **12**, 94–102. 2

Zadeh, L. (1985). Making computers think like people. *IEEE. Spectrum,*, **8**, 26–32. 2

Zhang, Q.Y. & Kandel, A. (1993). Compensatory neurofuzzy systems with fast learning algorirhms. In *Proceedings of IEEE Conference on Neural Networks*, 312–316, San Francisco. 2.3.2