

**DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND
APPLIED SCIENCES**

**NEURO – FUZZY CLASSIFICATION OF
WISCONSIN BREAST CANCER DATABASE**

**by
Sedat KIRTULUKOĞLU**

September, 2009

İZMİR

NEURO – FUZZY CLASSIFICATION OF WISCONSIN BREAST CANCER DATABASE

**A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Master of Science
in Electrical and Electronics Engineering Program**

**by
Sedat KIRTULUKOĞLU**

September, 2009

İZMİR

M. Sc. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled “**NEURO – FUZZY CLASSIFICATION OF WISCONSIN BREAST CANCER DATABASE**” completed by **SEDAT KIRTULUKOĞLU** under supervision of **ASST. PROF. DR. METEHAN MAKİNACI** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Metehan MAKİNACI

Supervisor

(Jury Member)

(Jury Member)

Prof. Dr. Cahit HELVACI
Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENT

I would like to thank my advisor Asst. Prof. Dr. Metehan MAKİNACI for his valuable guidance and support for this project.

Besides, I would like to thank my family for their helping, supporting and encouraging me in my whole life.

Sedat KIRTULUKOĞLU

NEURO – FUZZY CLASSIFICATION OF WISCONSIN BREAST CANCER DATABASE

ABSTRACT

The automatic diagnosis of breast cancer is an important, real-world medical problem. In this paper a Fuzzy Logic system design for diagnosing and analyzing the breast cancer and the learning procedure of this system was described. For this purpose we dealt with Wisconsin Breast Cancer Database (WBCD). This system extracts classification rules from trained network based on Fuzzy Logic. Analyzing both malignant and benign cell features, we could also generate the rules for classification depending on the cell features using Fuzzy Inference System (FIS) editor using MATLAB. In this project, we describe the accuracy of the trained networks and compare the result with the outputs of the classifiers constructed by using both k-nearest neighbor (KNN) and Bayes classifier. Finally we could say that our approach to the disease diagnosis using fuzzy logic had a high classification rate of over 96.93 % average and 99.12 % best.

Keywords: Fuzzy logic, fuzzy systems, fuzzy classifier, k-nearest neighbor, Bayes classifier, Wisconsin Breast Cancer Diagnosis.

WISCONSIN GÖĞÜS KANSERİ VERİTABANININ NÖRAL – BULANIK SINIFLANDIRILMASI

ÖZ

Göğüs kanseri gerçek dünyanın önemli bir medikal problemidir. Bu tezde göğüs kanserini tanımlama ve analiz etme için dizayn edilen bir bulanık mantık sistemi ve bu sistemin öğrenme prosedürü açıklandı. Bu amaçla Wisconsin göğüs kanseri veritabanı ele alındı. Bu sistem bulanık mantık kullanarak eğitilmiş bir ağdan türetilen sınıflandırma kurallarını oluşturur. Aynı zamanda iyi huylu ve kötü huylu hücrelerin özelliklerini inceleyip, MATLAB'daki bulanık çıkarım sistem düzenleyici kullanarak sınıflandırma için gerekli olan kuralları da oluşturduk. Bu projede eğitilmiş ağların doğruluğunu açıkladık ve çıkan sonuçları hem en yakın k-komşu hem de Bayes sınıflandırıcı kullanarak da karşılaştırdık. Sonuç olarak söyleyebiliriz ki, hastalığın tanısında kullandığımız bulanık mantık ortalama 96.93 % ve en iyi 99.12 % gibi yüksek sınıflandırma başarısına sahip.

Anahtar sözcükler: Bulanık mantık, bulanık sistem, bulanık sınıflandırıcı, en yakın k komşu, Bayes sınıflandırıcı, Wisconsin göğüs kanseri tanısı.

CONTENTS

	Page
M. Sc. THESIS EXAMINATION RESULT FORM.....	ii
ACKNOWLEDGEMENT	iii
ABSTRACT	iv
ÖZ	v
CHAPTER ONE – INTRODUCTION	1
1.1 Breast Cancer.....	1
1.2 Wisconsin Breast Cancer Database	1
1.3 Literature Review	2
1.4 Outline	5
CHAPTER TWO – FUZZY LOGIC	6
2.1 Fuzzy Sets.....	6
2.2 Operations with Fuzzy Sets	8
2.3 Membership Functions	8
2.3.1.7 Properties of Membership Functions	9
2.4 Fuzzy Relations, Fuzzy Implications	10
2.5 Fuzzy Propositions and Fuzzy Logic	10
2.6 If-Then Rules.....	10
2.7 Fuzzy Inference Method.....	11
2.8 Fuzzification, Rule Evaluation, Defuzzification	12
CHAPTER THREE – FUZZY INFERENCE SYSTEMS IN MATLAB	13
3.1 Fuzzy Inference System Process	13
3.1.1 Step 1. Fuzzify Inputs	13

3.1.2 Step 2. Apply Fuzzy Operator	14
3.1.3 Step 3. Apply Implication Method	14
3.1.4 Step 4. Aggregate All Outputs.....	14
3.1.5 Step 5. Defuzzify	14
CHAPTER FOUR – APPLICATION OF FUZZY INFERENCE SYSTEM.....	16
4.1 FIS Editor	16
4.2 Membership Function Editor.....	17
4.3 The Rule Editor	18
4.4 The Rule Viewer.....	21
CHAPTER FIVE – ADAPTIVE NEURO – FUZZY INFERENCE SYSTEM..	22
5.1 Model Learning and Inference Through ANFIS	23
5.1.2 FIS Structure and Parameter Adjustment	23
5.3 Some Constraints of ANFIS.....	23
CHAPTER SIX – OTHER CLASSIFICATION METHODS	25
6.1 Bayes Classification	25
6.1.1 Minimization of misclassification	26
6.1.2 Classification with reject-option.....	26
6.2 K-Nearest Neighbor Classification.....	27
CHAPTER SEVEN – RESULTS.....	30
7.1 ANFIS Classification	30
7.1.1 2 membership function compositions.....	30
7.1.1.1 2 Rule	30
7.1.1.2 4 Rule	33
7.1.1.3 8 Rule	35

7.1.1.4 16 Rule	38
7.1.1.4 32 Rule	39
7.1.1.5 64 Rule	40
7.1.1.6 128 Rule	40
7.1.1.7 256 Rule	41
7.1.2 3 Membership Function Compositions.....	42
7.1.2.1 3 Rule	42
7.1.2.2 9 Rule	43
7.1.2.3 27 Rule	44
7.1.2.4 81 Rule	44
7.2 FIS Classification	45
7.3 KNN Classification	46
7.4 Bayes Classification	47
CHAPTER EIGHT – CONCLUSION.....	48
REFERENCES.....	52

CHAPTER ONE

INTRODUCTION

1.1 Breast Cancer

Cancer is a group of diseases in which cells in the body grow, change, and multiply out of control. Usually, cancer is named after the body part in which it originated; thus, breast cancer refers to the erratic growth and proliferation of cells that originate in the breast tissue. A group of rapidly dividing cells may form a lump or mass of extra tissue. These masses are called tumors. Tumors can either be cancerous (malignant) or non-cancerous (benign). Malignant tumors penetrate and destroy healthy body tissues. This malignant tumor that has developed from cells is referred as breast cancer. (Imaginis, 1999)

1.2 Wisconsin Breast Cancer Database

The Wisconsin Breast Cancer Database (WBCD) is a popular choice for evaluating classifiers developed by the statistics, neural network and machine learning communities. This database was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg (Mangasarian, Wolberg, 1990). It represents a reasonably two-class problem with 9 continuous-valued inputs. A total of 683 instances (441 benign, 242 malignant) with complete input specification are provided. We are interested in classifying this database as benign and malignant by dividing the instances into training and testing sets. Detailed description of WBCD is given in section 4.3. Some rule extraction strategies have also been applied on this data set by using fuzzy logic, K-nearest neighbor and bayes classifiers.

1.3 Literature Review

In the past few years lots of research has been made in order to diagnose the cancer disease. Related to our project we examined both the breast cancer classification with any method and the classification of any cancer data made by fuzzy logic. While searching we attached mostly on the methods they used, the percentage of the trained data in the whole data and the efficiency of the methods. For this purpose we investigated some thesis and article mentioned below.

In the first article (Jain & Abraham, 2003) the used data is Wisconsin breast cancer data but somehow different from our data; this data has 32 attributes (30 real-valued input features) and 569 instances of which 357 are benign and 212 are of malignant class. The main method Jain and Abraham used is fuzzy classification constituting four fuzzy rule generation method. All the rules generate one of the fuzzy if-then rules using some methods. The successes of the classification methods are: mean and standard deviation is 92.2 %, histogram of attribute values is 86.7 %, modified grid is 62.57 % and the simple grid method has a high classification of 99.73 %.

In 2003 a fuzzy expert system design for diagnosis of prostate cancer has been built (Saritas, Allahverdi & Sert, 2003). The method Saritas and his friends used is just like the method we used which will be described in chapter 3 and chapter 4 named as fuzzy inference system. The success of the fuzzy expert system classification is 86 %, bayes classification is 79 and the k-nearest neighbor classification is 78 %.

A neural network was designed (Setiono, 1999) in order to classify the breast cancer diagnosis using the Wisconsin breast cancer data which is slightly different from our data. The attributes of the data are the same as ours. Setiono used lots of methods in neural network and the success of the classification reaches up to 98 %.

In 2001 another efficient fuzzy classifier with the ability of feature selection based on a fuzzy entropy measure (Lee, Chen, Chen, Jou, 2001) was designed. In this work same data with ours is used, but the method is different. The success when 6 of the 9 cell features had taken into account is 95.14 %, and the success when all the features used is a little lower 94.67 %.

Neuro-fuzzy classification (NEFCLASS) method was used to classify the prostate cancer (Keles, Hasiloglu, Keles, Aksoy, 2007). This new approach, NEFCLASS, is a tool having batch learning, automatic cross validation, automatic determination of the rule base size and handling of missing values to increase its interpretability. This system is like our system but works on a java platform. Using NEFCLASS, Keles and his friends were able to classify in some different methods with the success of 98.89 % by using triangular classifier, 98.89 % by using trapezoidal classifier, 92.22 % by using bell-shaped classifier and 95.99 % by using adaptive neuro-fuzzy inference system (ANFIS) that we also used and explained in chapter 4.

A self-adaptive neuro-fuzzy inference system (Wang, Lee, 2002) is constructed in order to classify iris, Wisconsin breast cancer and wine. The main idea of the system is the same as our adaptive system with no prior knowledge of the data. The entire algorithm depends on having training and testing data sets so that the system can develop its own rules. Wang and his friends used the same data as we used in our classification project. So a good comparison can be made between our and their systems. There are three different methods they used having a success of 96.3 %, 96.07 % and 96.28 %.

In 2004 statistical neural network structures are applied to classify the Wisconsin breast cancer data (Kıyan, Yıldırım, 2004). The used data are the same as we used. Kıyan and her friend constructed four different neural network structures. The radial basis network has a classification success of is 96.18 %, probabilistic neural network has 97 %, generalized regression neural network has 98.8 % and multilayer perception has a classification success of 95.74 %.

Another different classification prostate cancer data was made in 1997 (Lorenz, Blüm, Ermert, Senge). Lorenz and his friends used neuro-fuzzy classification systems. Two of the methods Lorenz and his friends used were like our methods. One has 16 rules, 2 membership functions and 50 epochs. The other method is done using adaptive neuro – fuzzy inference system with 3 rules and 3 membership functions. Also one of the methods used in the project is the same as Keles and his friends' method called NEFCLASS. First method trainable fuzzy system Lorenz and his friends used has a classification success of 84.7 %, histogram based fuzzy system has 85 %, adaptive neuro-fuzzy inference system has 87.2 % and NEFCLASS has a success of 87.9 %.

A study of data-driven generation of compact and linguistically-sound fuzzy classifiers based on a decision-tree initialization (Abonyi, Roubos, Szeifert, 2002) was made to classify the Wisconsin breast cancer data. Two different methods were applied: decision-tree initialization with 10-fold cross validation and neuro-fuzzy classification method with 135 rules. Also the used data is the same as ours. The decision-tree initialization with 10-fold cross validation has a classification success of 96.82 % and the neuro-fuzzy classification method with 135 rules has 95.06 % of success.

A fuzzy expert system (FES) design was constructed in 2004 (Chang, Lilly, 2004). Chang and his friend used the same data as ours for classification. The method is also resembles our method, but somehow different. As mentioned earlier the data has 9 features. But Chang and his friend used 2 of the 9 features with 2 membership functions and created 3 rules only. But the success of the classification is satisfying with a rate of 96.5 %.

Another different data classification was made on vibration signals of cylindrical shells (Marwala, Tettey, Chakraverty, 2006). The type of the data totally different from ours but the reasoning of the method is the same. Marwala and his friends used neuro – fuzzy classification method in their project. They changed the threshold of

the system and had two successful classifications. Varied threshold had the classification success of 91.62 % and fixed threshold method had 90.42 %.

Another system to classify Wisconsin breast cancer data, same as our data, was designed in 1996 (Nauck, Nauck, Kruse, 1996). The main method Nauck and his friends used for neuro – fuzzy model was NEFCLASS. In the study of classifying the cancer data, Nauck and his friends used all the 9 cell features. In order to see the efficiency of the system, rule and the epoch number were changed. Fuzzy clustering method with 3 rules and 80 epochs had the classification success of 92.7 % and the method with 4 rules and 100 epochs had 96.5 %.

In the last study we investigated, a fuzzy genetic approach was used (Pena-Reyes, Sipper, 1998). Reyes and his friend used Wisconsin breast cancer data. In their study they worked the effect of the train and the test data sets' importance in percentage. They tried several percentages of train and test data sets and concluded on the results that; 75 % train, 25 % test method had a success of 96.76 % classification and 50 % train 50 % test method had a success of 96.23 % classification of the data.

1.4 Outline

In the first chapter an introduction to breast cancer and WBCD are given. Typical attributes of the database are also given in the first chapter. A long literature review is given and outline ends this chapter. In chapter 2 an introduction and theoretical background of fuzzy logic is given. Fuzzy inference system built in MATLAB is studied in chapter 3. In the next section chapter 4, the application of fuzzy inference system is explained. The illustration of this application is done using MATLAB on WBCD to be able to compare the results with the adaptive system. Chapter 5 explains the adaptive neuro – fuzzy inference system (ANFIS). In the subsequent part, chapter 6, k-nearest neighbor and Bayes classification methods explained briefly. Results of the all classification methods are given in chapter 7. The final chapter, chapter 8, finishes the thesis with the overall conclusion.

CHAPTER TWO

FUZZY LOGIC

The term "fuzzy logic" emerged in the development of the theory of fuzzy sets by Lotfi Zadeh (1965). A fuzzy subset A of a (crisp) set X is characterized by assigning to each element x of X the degree of membership of x in A (e.g., X is a group of people, A the fuzzy set of old people in X). Now if X is a set of propositions then its elements may be assigned their degree of truth, which may be "absolutely true," "absolutely false" or some intermediate truth degree: a proposition may be more true than another proposition. This is obvious in the case of imprecise propositions like "this person is old" (beautiful, rich, etc.). In the analogy to various definitions of operations on fuzzy sets (intersection, union, complement, ...) one may ask how propositions can be combined by connectives (conjunction, disjunction, negation, ...) and if the truth degree of a composed proposition is determined by the truth degrees of its components, i.e. if the connectives have their corresponding truth functions (like truth tables of classical logic). Saying "yes" (which is the mainstream of fuzzy logic) one accepts the truth-functional approach; this makes fuzzy logic to something distinctly different from probability theory since the latter is not truth-functional (the probability of conjunction of two propositions is not determined by the probabilities of those propositions). (Stanford encyclopedia of philosophy, 2006).

2.1 Fuzzy Sets

In this section all the figures are taken from and the theoretical background is based on the book Anonymous, (1999), *Matlab – Fuzzy Logic Toolbox User's Guide* (version 2) Natick: MathWorks.

Fuzzy logic starts with the concept of a fuzzy set. A fuzzy set is a set without a crisp, clearly defined boundary. It can contain elements with only a partial degree of membership.

Now consider the set of days comprising a weekend. The figure 2.1 is one attempt at classifying the weekend days using a continuous scale time plot of weekend-ness.

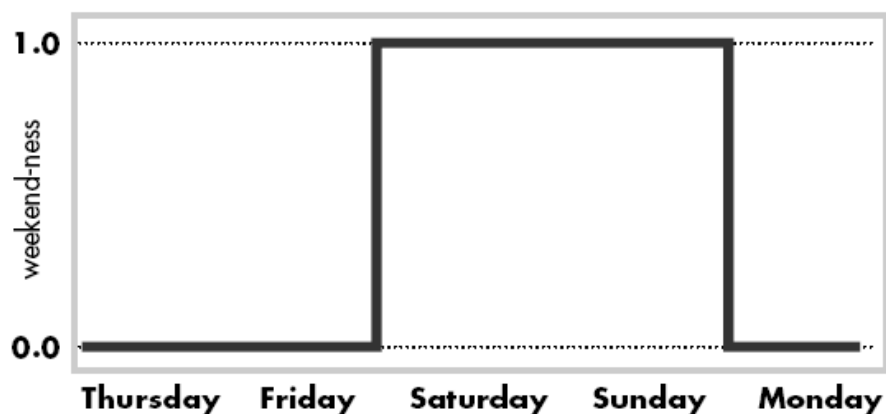


Figure 2.1 Days of the weekend two-valued membership.

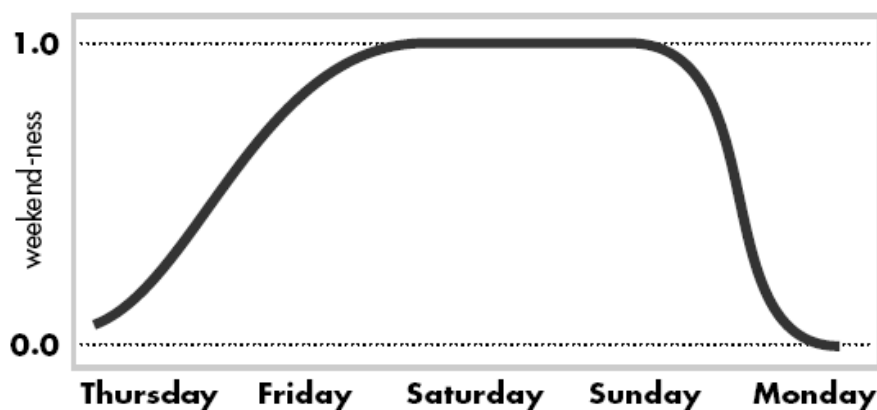


Figure 2.2 Days of the weekend multi-valued membership.

The figure 2.2 shows a smoothly varying curve that accounts for the fact that all of Friday, and, to a small degree, parts of Thursday, participate in weekend-ness and thus deserve partial membership in the fuzzy set of weekend moments. The curve that defines the weekend-ness of any instant in time is a function that maps the input space (time of the week) to the output space (weekend-ness). Specifically it is known as a membership function.

2.2 Operations with Fuzzy Sets

The theoretical background in this part is based on the book Kasabov, N., K., (1998). *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering* (2nd ed.). London: MIT. Detailed description is also given in the same book.

Ordinary (crisp) sets are a special case of fuzzy sets, when two membership degrees only, 0 and 1 are used, and crisp borders between the sets are defined. All definitions, proofs, and theorems that apply to fuzzy sets must also be valid in the case when the fuzziness becomes zero, that is, when the fuzzy set turns into an ordinary one.

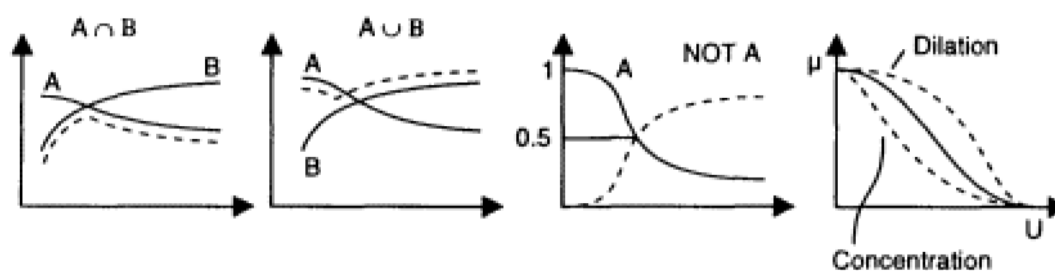


Figure 2.3 Five operations with two fuzzy sets A and B approximately represented in a graphical form.

2.3 Membership Functions

This section is based on the book Anonymous, (1999), *Matlab – Fuzzy Logic Toolbox User's Guide* (version 2) Natick: MathWorks.

A membership function (MF) is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1. The input space is sometimes referred to as the universe of discourse, an interesting name for a simple concept. Some of the membership functions used in MATLAB is given below. More membership functions are given in the reference book.

Triangular membership function is shown in figure 2.4.

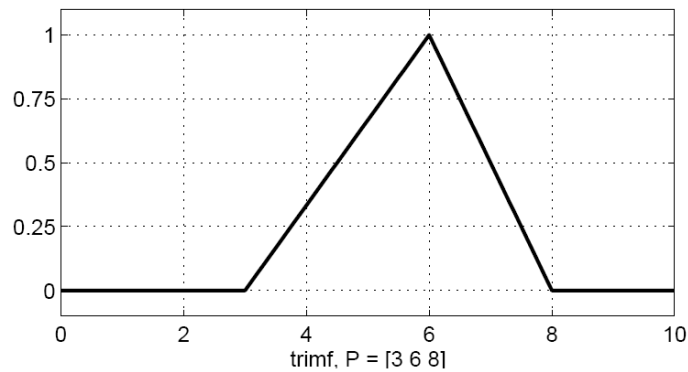


Figure 2.4 Triangular membership function (trimf)

Gaussian membership function is shown in figure 2.5.

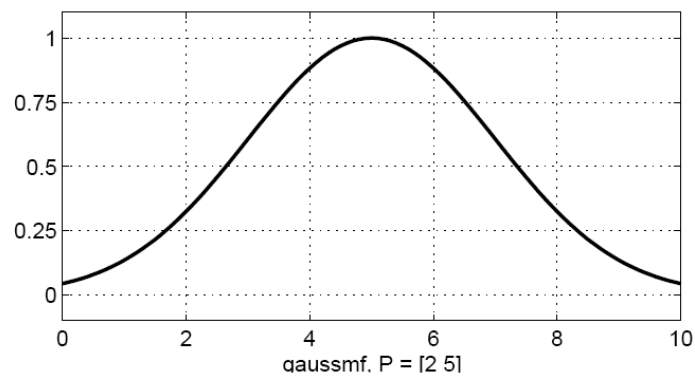


Figure 2.5 Gaussian membership function (gaussmf)

2.3.1.7 Properties of Membership Functions

- Fuzzy sets describe vague concepts
- A fuzzy set admits the possibility of partial membership in it.
- The degree an object belongs to a fuzzy set is denoted by a membership value between 0 and 1.
- A membership function associated with a given fuzzy set maps an input value to its appropriate membership value.

2.4 Fuzzy Relations, Fuzzy Implications

Fuzzy relations make it possible to represent ambiguous relationship like “the grades of the third and second year classes are similar” or “team A performed slightly better than team B” or “the more fat you eat, the higher the risk of cancer attack”. Fuzzy relations link two fuzzy sets in a predefined manner.

If a fuzzy set defined over a universe U , and B is a fuzzy set defined over a universe V , then a fuzzy relation $R(A,B)$ is any fuzzy set defined on the cross-product universe $U \times V = \{(u,v) / u \in U, v \in V\}$. A fuzzy relation is characterized by its membership function.

$$\mu_{R(u,v)} : U \times V \rightarrow [0,1] \quad (2.1)$$

2.5 Fuzzy Propositions and Fuzzy Logic

The biggest restriction in classic propositional and predicate logic is the fact that the propositions can have their truth-values as either “true” or “false”. This restriction has its assets as well as its drawbacks. The main asset is that the decision obtained is exact and precise. The main drawback, however, is that it can not reflect the enormous diversity of the real world, which is analog and not digital. The truth value of a proposition in classical logic can not be unknown.

In order to overcome this limitation of classic logic, multi-valued logic has been developed.

2.6 If-Then Rules

Fuzzy sets and fuzzy operators are the subjects and verbs of fuzzy logic. These if-then rule statements are used to formulate the conditional statements that

compromise fuzzy logic. A generalized form of the fuzzy rule is the following form:

$$\text{If } x_1 \text{ is } A1 \text{ AND } x_2 \text{ is } A2 \text{ AND } \dots \text{ AND } x_k \text{ is } Ak, \text{ THEN } y \text{ is } B, \quad (2.2)$$

where x_1, x_2, \dots, x_k, y are fuzzy variables (attributes) over different universes of discourse $Ux_1, Ux_2, \dots, Ux_k, Uy$ and $A1, A2, \dots, Ak, B$ are their possible values over the same universes.

2.7 Fuzzy Inference Method

Fuzzy inference method is a matching in a wider sense that is, matching a domain space with a solution space.

A fuzzy inference method combines the results B_i' for the output variable y inferred by all the fuzzy rules for a given set of input facts. In a fuzzy production system, which performs cycles of inference, all the fuzzy rules are fired at every cycle and they all contribute to the final result. Some of the main *else-links* between fuzzy rules are:

OR-link: The results obtained by the different rules are “OR-ed” in a monotonic fashion, so the more that is inferred by any of the rules, the higher the resulting degree of the membership function for B' . Max operation is applied to achieve this operation.

AND-link: The final result is obtained after a min operation over the corresponding values of the inferred by all the rules or fuzzy membership functions.

The selection of the “else-link” depends on the context in which the rules are written.

2.8 Fuzzification, Rule Evaluation, Defuzzification

When the input data are crisp and the output values are expected to be crisp too, then the “fuzzification, rule evaluation, defuzzification” inference method is applied over fuzzy rules of the type of if x_1 is A_1 and x_2 is A_2 THEN y is B .

Fuzzification is the process of finding the membership degrees $\mu_{A_1}(x_1')$ and $\mu_{A_2}(x_2')$ to which input data (x_1') and (x_2') belong to the fuzzy sets A_1 and A_2 antecedent part of a fuzzy rule.

Defuzzification is the process of calculating a single-output numerical value for a fuzzy output variable on the basis of the inferred resulting membership function for this variable.

These methods are explained in the application of Fuzzy Inference System.

CHAPTER THREE

FUZZY INFERENCE SYSTEMS IN MATLAB

This section is based on the book Anonymous, (1999), *Matlab – Fuzzy Logic Toolbox User’s Guide* (version 2) Natick: MathWorks.

Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic. The mapping then provides a basis from which decisions can be made, or patterns discerned. The process of fuzzy inference involves all of the pieces that are described in the previous sections: membership functions, fuzzy logic operators, and if-then rules. There are two types of fuzzy inference systems that can be implemented in the Fuzzy Logic Toolbox: Mamdani-type and Sugeno-type. These two types of inference systems vary somewhat in the way outputs are determined. Descriptions of these two types of fuzzy inference systems can be found in the references (Jang, Sun, 1997).

3.1 Fuzzy Inference System Process

3.1.1 Step 1. Fuzzify Inputs

The first step is to take the inputs and determine the degree to which they belong to each of the appropriate fuzzy sets via membership functions. In the Fuzzy Logic Toolbox, the input is always a crisp numerical value limited to the universe of discourse of the input variable (in this case the interval between 0 and 10) and the output is a fuzzy degree of membership in the qualifying linguistic set (always the interval between 0 and 1). Fuzzification of the input amounts to either a table lookup or a function evaluation

3.1.2 Step 2. Apply Fuzzy Operator

Once the inputs have been fuzzified, we know the degree to which each part of the antecedent has been satisfied for each rule. If the antecedent of a given rule has more than one part, the fuzzy operator is applied to obtain one number that represents the result of the antecedent for that rule. This number will then be applied to the output function. The input to the fuzzy operator is two or more membership values from fuzzified input variables. The output is a single truth value.

3.1.3 Step 3. Apply Implication Method

Before applying the implication method, we must take care of the rule's weight. Every rule has a weight (a number between 0 and 1), which is applied to the number given by the antecedent. Generally this weight is 1 (as it is for this example) and so it has no effect at all on the implication process. From time to time you may want to weight one rule relative to the others by changing its weight value to something other than 1.

3.1.4 Step 4. Aggregate All Outputs

Since decisions are based on the testing of all of the rules in an FIS (Fuzzy Inference Systems), the rules must be combined in some manner in order to make a decision. Aggregation is the process by which the fuzzy sets that represent the outputs of each rule are combined into a single fuzzy set. Aggregation only occurs once for each output variable, just prior to the fifth and final step, defuzzification. The input of the aggregation process is the list of truncated output functions returned by the implication process for each rule. The output of the aggregation process is one fuzzy set for each output variable.

3.1.5 Step 5. Defuzzify

The input for the defuzzification process is a fuzzy set (the aggregate output fuzzy set) and the output is a single number. As much as fuzziness helps the rule evaluation

during the intermediate steps, the final desired output for each variable is generally a single number. However, the aggregate of a fuzzy set encompasses a range of output values, and so must be defuzzified in order to resolve a single output value from the set.

CHAPTER FOUR

APPLICATION OF FUZZY INFERENCE SYSTEM

4.1 FIS Editor

The FIS Editor displays general information about a fuzzy inference system. There's a simple diagram at the top that shows the names of each input variable on the left, and those of each output variable on the right. The sample membership functions shown in the boxes are just icons and do not depict the actual shapes of the membership functions.

For our example we will construct a nine-input, one output system. Nine inputs are cell features and their names are input_1, input_2 and so on. The output is the class of the cell. Our editor is shown in figure 4.1

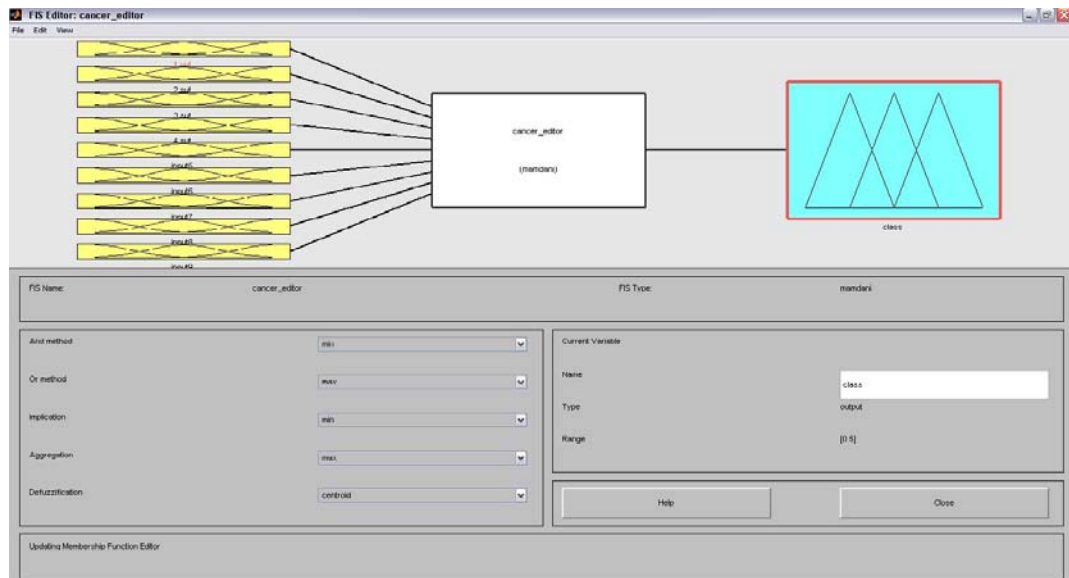


Figure 4.1 Cancer FIS editor

As seen in above figure, our “And method” is min, “Or method” is max, “implication” is min, “aggregation” is max and “defuzzification” is centroid.

4.2 Membership Function Editor

Next we created the membership functions for the input variables, cell features. To create the input variable membership functions we used a scale from 0 to 10 to represent the variables. We created these membership functions according to the rules explained in the next section. Our membership function editor with 9 inputs is given in figure 4.2.

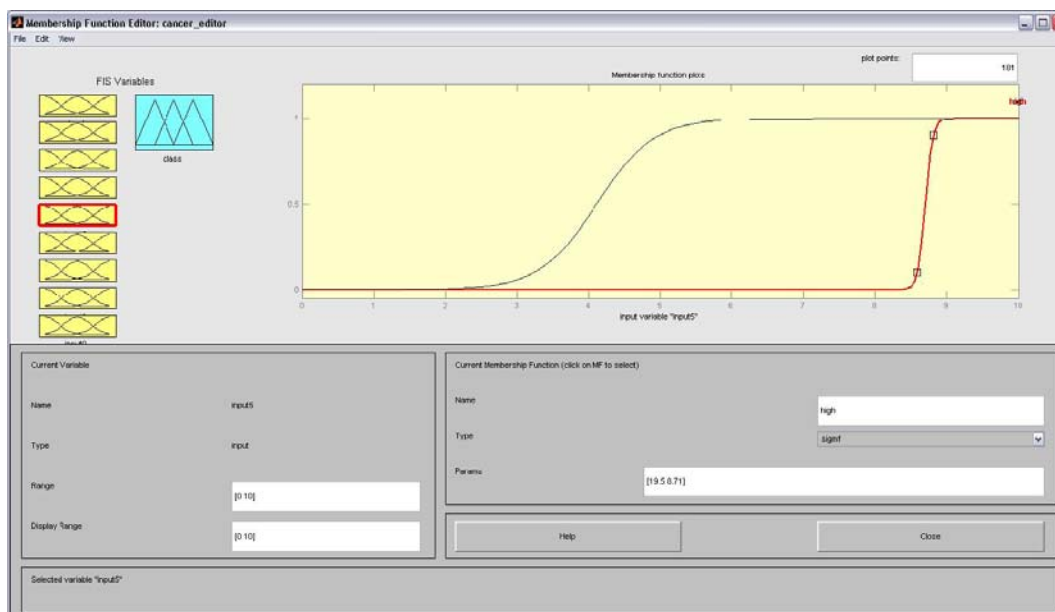


Figure 4.2 Membership function editor with 9 inputs

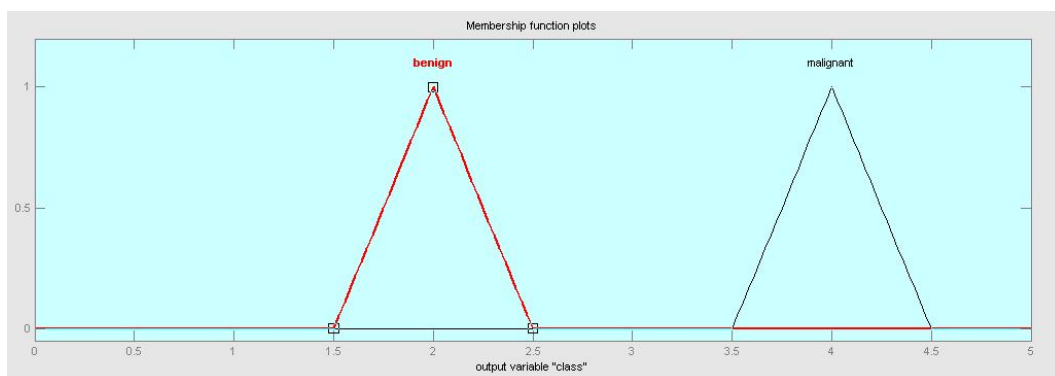


Figure 4.3 Membership functions of output variable "class"

We used triangular membership function types for the output. First, benign is just over the value 2 and the malignant is just over the value 4. Our membership functions of output variables are given in figure 4.3.

4.3 The Rule Editor

Constructing rules using the graphical rule editor interface is fairly self-evident. Based on the descriptions of the input and output variables defined with the FIS Editor, the Rule Editor allows you to construct the rule statements automatically, by clicking on and selecting one item in each input variable box, one item in each output box, and one connection item.

Since we dealt with Wisconsin Breast Cancer Database (WBCD) for this project, it will be better to explain the rules we created via this database. WBCD represents a two-class problem with 9 continuous valued inputs. A total of (441 benign, 242 malignant) with complete input specification are provided. In the process of getting used to the data what rule it is based on, we dealt with 11 different attributes. First attribute is the id number of the cell. The last, 11th attribute is class. The remaining attributes are the cell features. The database information is given in table 4.1.

Table 4.1 WBCD information

Attribute	Domain
1. Sample code number (Sc)	id number
2. Clump Thickness (Ct)	assigned between 1-10
3. Uniformity of Cell Size (C. Size)	assigned between 1-10
4. Uniformity of Cell Shape (C. Shape)	assigned between 1-10
5. Marginal Adhesion (Ma)	assigned between 1-10
6. Single Epithelial Cell Size (Ecs)	assigned between 1-10
7. Bare Nuclei (Bn)	assigned between 1-10
8. Bland Chromatin (Bc)	assigned between 1-10
9. Normal Nucleoli (Nn)	assigned between 1-10
10. Mitoses (M)	assigned between 1-10
11. Class	(2 for benign, 4 for malignant)

As explained earlier in order to be able use fuzzy logic to classify, we had to organize the rules we would use in our FIS. Some of the data is shown below, since we are going to give reasons about how we examined them. We will show them in four parts according to their characteristics. While examining the data we recognized that the class attribute is classified into two groups according to four rules:

Table 4.2 Referans table for rule 1

Id Number	Ct	C. Size	C. Shape	Ma	Ecs	Bn	Bc	Nn	M	Class
1050718	6	1	1	1	2	1	3	1	1	2
1113483	5	2	3	1	6	10	5	1	1	4
1116132	6	3	4	1	5	2	3	9	1	4

It is shown in table 4.2 that if at least one of the cell features contains the number “9” or “10”, the cell is class 4, malignant.

Table 4.3 Referans table for rule 2

Id Number	Ct	C. Size	C. Shape	Ma	Ecs	Bn	Bc	Nn	M	Class
1113038	8	2	4	1	5	1	5	4	4	4
859164	5	3	3	1	3	3	3	3	3	4
1240337	5	2	2	2	2	2	3	2	2	2

It is shown in table 4.3 that if the cell feature ”10” (mitoses) is numbered 3 or higher, the cell is class is 4,malignant.

Table 4.4 Referans table for rule 3

Id Number	Ct	C. Size	C. Shape	Ma	Ecs	Bn	Bc	Nn	M	Class
1148278	3	3	6	4	5	8	4	4	1	4

It is shown in table 4.4 that if the cell features are high (there is not any reference to say it is high) the cell is class is 4, malignant.

Table 4.5 Referans table for rule 4

Id Number	Ct	C. Size	C. Shape	Ma	Ecs	Bn	Bc	Nn	M	Class
1152331	4	1	1	1	2	1	3	1	1	2
1155546	2	1	1	2	3	1	2	1	1	2
1156272	1	1	1	1	2	1	3	1	1	2
1156948	3	1	1	2	2	1	1	1	1	2
1157734	4	1	1	1	2	1	3	1	1	2
1158247	1	1	1	1	2	1	2	1	1	2
1160476	2	1	1	1	2	1	3	1	1	2

Otherwise cell class is 2, benign. Table 4.5 shows examples of benign cells.

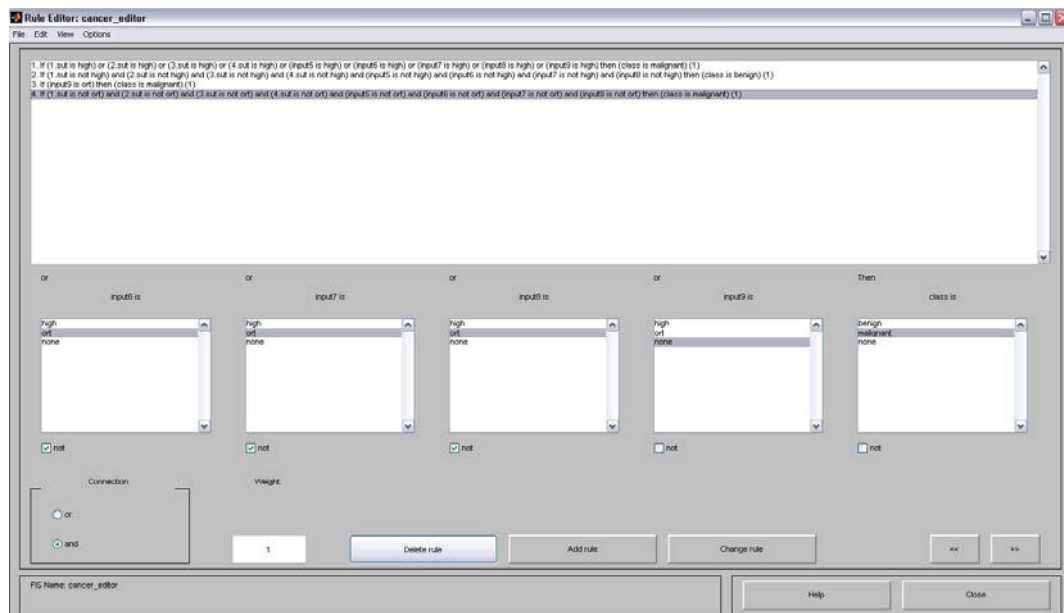


Figure 4.4 Rule editor with 4 rules, 9 inputs and 1 output (class)

Our rule editor is shown in figure 4.4. It has 4 rules whose creations are described earlier, 9 inputs which are the cell features and 1 output which is the class of the cell.

4.4 The Rule Viewer

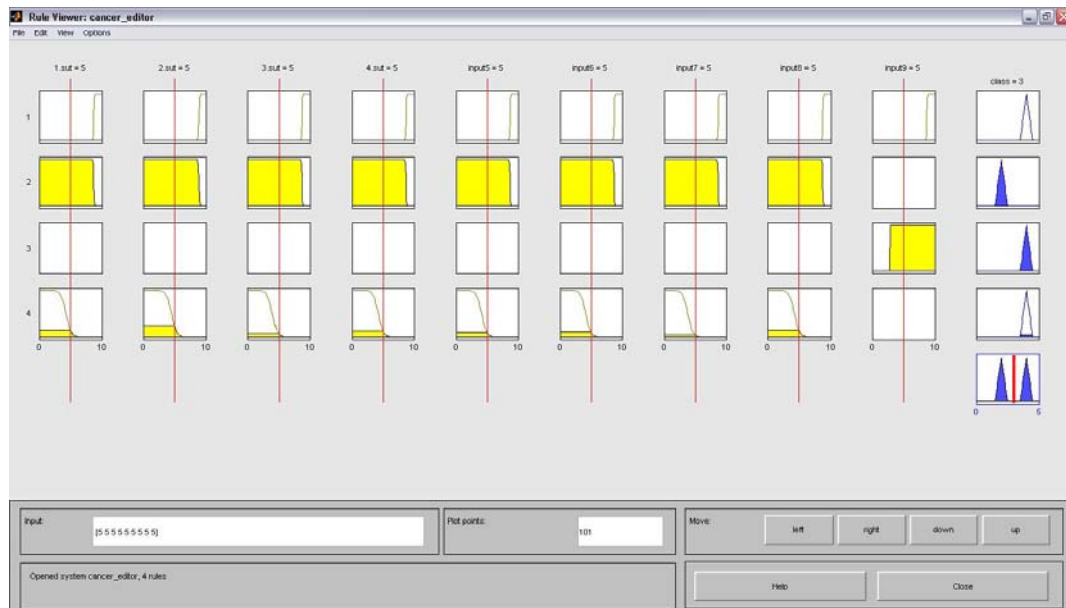


Figure 4.5 Rule viewer with 4 rules, 9 inputs and 1 output

Our rule viewer is shown in figure 4.5. The Rule Viewer displays a roadmap of the whole fuzzy inference process. It's based on the fuzzy inference diagram described in the previous section. You see a single figure window with 41 small plots nested in it. The ten small plots across the top of the figure represent the antecedent and consequent of the first rule. Each rule is a row of plots, and each column is a variable. The first four columns of plots (the thirty six yellow plots) show the membership functions referenced by the antecedent, or the if-part of each rule. The last column of plots (the four blue plots) shows the membership functions referenced by the consequent, or the then-part of each rule.

The red line above the first nine columns is for changing the input values to generate a new output response. The red line above last blue box provides a defuzzified value. The bottom-right plot shows how the output of each rule is combined to make an aggregate output and then defuzzified.

In the next chapter we will describe the classification of the cells adaptive neuro-fuzzy inference system (ANFIS) which is capable of learning and creating rules.

CHAPTER FIVE

ADAPTIVE NEURO – FUZZY INFERENCE SYSTEM

The basic structure of the type of fuzzy inference system is a model that maps input characteristics to input membership functions, input membership function to rules, rules to a set of output characteristics, output characteristics to output membership functions, and the output membership function to a single-valued output or a decision associated with the output. We have only considered membership functions that have been fixed, and somewhat arbitrarily chosen. Also, we've only applied fuzzy inference to modeling systems whose rule structure is essentially predetermined by the user's interpretation of the characteristics of the variables in the model.

In this section we discuss the use of the function *anfis* (adaptive neuro-fuzzy inference system). This system applies fuzzy inference techniques to data modeling. The shape of the membership functions depends on parameters, and changing these parameters will change the shape of the membership function. Instead of just looking at the data to choose the membership function parameters, it is possible to choose membership function parameters automatically.

There will be some modeling situations in which you can't just look at the data and discern what the membership functions should look like. Rather than choosing the parameters associated with a given membership function arbitrarily, these parameters could be chosen so as to tailor the membership functions to the input/output data in order to account for these types of variations in the data values. This is where the so-called neuro-adaptive learning techniques incorporated into *anfis* in the Fuzzy Logic Toolbox can help.

5.1 Model Learning and Inference Through ANFIS

The basic idea behind these neuro-adaptive learning techniques is very simple. These techniques provide a method for the fuzzy modeling procedure to learn information about a data set, in order to compute the membership function parameters that best allow the associated fuzzy inference system to track the given input/output data. This learning method works similarly to that of neural networks.

5.1.2 FIS Structure and Parameter Adjustment

A network-type structure similar to that of a neural network, which maps inputs through input membership functions and associated parameters, and then through output membership functions and associated parameters to outputs, can be used to interpret the input/output map.

The parameters associated with the membership functions will change through the learning process. The computation of these parameters (or their adjustment) is facilitated by a gradient vector, which provides a measure of how well the fuzzy inference system is modeling the input/output data for a given set of parameters. Once the gradient vector is obtained, any of several optimization routines could be applied in order to adjust the parameters so as to reduce some error measure (usually defined by the sum of the squared difference between actual and desired outputs). ANFIS uses either back propagation or a combination of least squares estimation and back propagation for membership function parameter estimation.

5.3 Some Constraints of ANFIS

ANFIS is much more complex than the fuzzy inference systems discussed so far, and is not available for all of the fuzzy inference system options. Specifically, ANFIS only supports Sugeno-type systems, and these must be:

- First or zeroth order Sugeno-type systems
- Single output, obtained using weighted average defuzzification (linear or constant output membership functions)
- Of unity weight for each rule

Detailed theoretical background of adaptive neuro – fuzzy inference system is given in the symposium *Advances in Neural Networks* (Liu, Fei, Hou, Zhang, Sun, 1998).

CHAPTER SIX

OTHER CLASSIFICATION METHODS

To compare the results of the fuzzy systems described in the preceding chapter, we also used Bayes and k-nearest neighbor classification methods.

6.1 Bayes Classification

Based on the book *Statistical Pattern Recognition Toolbox for Matlab* (Franc, Hlavac, 2004).

The object under study is assumed to be described by a vector of observations $x \in X$ and hidden state $y \in Y$. The x and y are realizations of random variables with joint probability distribution $P_{XY}(x, y)$. A decision rule $q: X \rightarrow D$ takes a decision $d \in D$ based on the observation $x \in X$. Let $W: D \times Y \rightarrow \mathfrak{R}$ be a loss function which penalizes the decision $q(x) \in D$ when the true hidden state is $y \in Y$. Let $X \subseteq \mathfrak{R}^n$ and the sets Y and D be finite. The Bayesian risk $R(q)$ is an expectation of the value of the loss function W when the decision rule q is applied, i.e.,

$$R(q) = \int_x \sum_{y \in Y} P_{XY}(x, y) W(q(x), y) dx \quad (6.1)$$

The optimal rule q^* which minimizes the Bayesian risk (6.1) is referred to as the Bayesian rule

$$q^*(x) = \arg \min_y \sum_{y \in Y} P_{XY}(x, y) W(q(x), y), \quad \forall x \in X \quad (6.2)$$

The STPR (Statistical Pattern Recognition) tool built in MATLAB, we used in our

classification project, implements the Bayesian rule for two particular cases:

6.1.1 Minimization of misclassification

The set of decisions D coincides to the set of hidden states $Y = \{1, \dots, c\}$. The 0/1-loss function

$$W_{0/1}(q(x), y) = \begin{cases} 0 & \text{for } q(x) = y, \\ 1 & \text{for } q(x) \neq y. \end{cases} \quad (6.3)$$

is used. The Bayesian risk (6.1) with the 0/1-loss function corresponds to the expectation of misclassification. The rule $q: X \rightarrow Y$ which minimizes the expectation of misclassification is defined as

$$\begin{aligned} q(x) &= \arg \max_{y \in Y} P_{Y|X}(y | x), \\ &= \arg \max_{y \in Y} P_{X|Y}(x | y)P_Y(y). \end{aligned} \quad (6.4)$$

6.1.2 Classification with reject-option

The set of decisions D is assumed to be $D = Y \cup \{dont_know\}$. The loss function is defined as

$$W_{\varepsilon}(q(x), y) = \begin{cases} 0 & \text{for } q(x) = y, \\ 1 & \text{for } q(x) \neq y, \\ \varepsilon & \text{for } q(x) = dont_know, \end{cases} \quad (6.5)$$

where ε is penalty for the decision $dont_know$. The rule $q: X \rightarrow Y$ which minimizes the Bayesian risk with the loss function (5.5) is defined as

$$q(x) = \begin{cases} \arg \max_{y \in Y} P_{X|Y}(x | y)P_Y(y) & \text{if } 1 - \max_{y \in Y} P_{Y|X}(y | x) < \varepsilon, \\ dont_know & \text{if } 1 - \max_{y \in Y} P_{Y|X}(y | x) \geq \varepsilon. \end{cases} \quad (6.6)$$

To apply the optimal classification rules one has to know the class-conditional distributions $P_{x|Y}$ and priory distribution P_Y (or their estimates).

6.2 K-Nearest Neighbor Classification

Based on the book *Pattern Classification* (Duda, Hart, Stork, 2001).

K-nearest neighbor (kNN) classification is one of the most fundamental and simple classification methods and should be one of the first choices for a classification study when there is little or no prior knowledge about the distribution of the data. K-nearest neighbor classification was developed from the need to perform discriminant analysis when reliable parametric estimates of probability densities are unknown or difficult to determine. (Scholarpedia, 2009)

Theoretical background is based on the book

As expected, this rule classifies x by assigning it the label most frequently represented among the k -nearest samples; in other words, a decision is made by examining the labels on the k -nearest neighbors and taking a vote (Figure 6.1). We shall not go into a through analysis of the k -nearest neighbor rule. For two-class cases as in our classification project, one should avoid to have even k values in order not to have equal number of nearest sample or samples for each class.

We notice that if k is fixed and the number n of samples is allowed to approach infinity, then all of the k -nearest neighbors will converge to x . Hence, as in the single-nearest neighbor cases, the labels on each of the k -nearest neighbors are random variables, which independently assume the values w_i with probabilities $P(w_i | x), i = 1, 2$. If $P(w_m | x)$ is the larger a posteriori probability, then the Bayes decision rule always selects w_m . The single-nearest neighbor rule selects w_m with probability $P(w_m | x)$. The k -nearest neighbor rule selects w_m if a majority of the k -nearest neighbors are labeled w_m , an event of probability

$$\sum_{i=(k+1)/2}^k \binom{k}{i} P(w_m | x)^i [1 - P(w_m | x)]^{k-i} \quad (6.7)$$

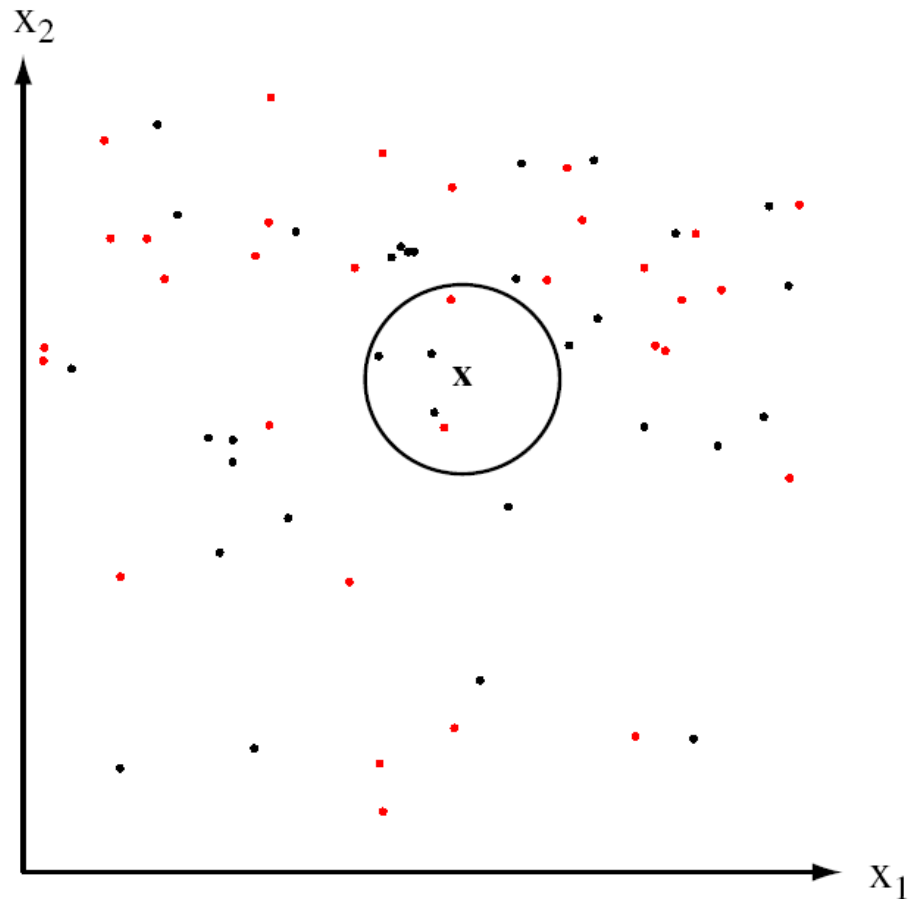


Figure 6.1 The k -nearest neighbor query starts at the test point and grows a spherical region until it encloses k training samples and the labels at the test point by a majority vote of these samples. In this $k = 5$ case, the test point would be labeled the category of the black points (Duda, Hart, Stork, 2001)

In general, the larger the value of k , the greater the probability that w_m will be selected. In figure 6.1 $k = 5$ case is given.

It can be shown that if k is odd, the large-sample two-class error rate for the k -nearest neighbor rule is bounded above by the function $C_k(P^*)$, where $C_k(P^*)$ is defined to be the smallest concave function of P^* greater than

$\sum_{i=0}^{(k-1)/2} \binom{k}{i} [(P^*)^{i+1} (1-P^*)^{k-i} + (P^*)^{k-i} (1-P^*)^{i+1}]$. Here the summation over the first

bracketed term represents the probability of error due to i points coming from the category having the minimum probability and $k-i > i$ points from the other category. The summation over the second term in the brackets is the probability that $k-i$ points are from the minimum probability category and $i+1 < k-i$ from the higher probability category. Both of these cases constitute under the k-nearest neighbor rule, and thus we must add them to find the full probability of error.

CHAPTER SEVEN

RESULTS

7.1 ANFIS Classification

In order to be able to explain the efficiency of fuzzy classification method we gave the results of different compositions of ANFIS. Here only the results are given and the detailed explanation of these compositions is discussed in the next chapter. For all of the classifications we used 680 instances out of 683. 170 benign and 170 malignant train data is used in order to make the prior probability the same for each class. The remaining 340 instances are used for testing. For each cross – validation these training data is changed among all of the data.

7.1.1 2 membership function compositions

7.1.1.1 2 Rule

In order to run the program faster we reduced the rule number to 2. To reduce the rule number to 2 we assigned 2 membership functions to one attribute at a time. Since we had 9 attributes, we had 9 different compositions.

Table 7.1 Different training compositions for 2 rule method

Training Compositions	Feature								
	1	2	3	4	5	6	7	8	9
1	2	1	1	1	1	1	1	1	1
2	1	2	1	1	1	1	1	1	1
3	1	1	2	1	1	1	1	1	1
4	1	1	1	2	1	1	1	1	1
5	1	1	1	1	2	1	1	1	1
6	1	1	1	1	1	2	1	1	1
7	1	1	1	1	1	1	2	1	1
8	1	1	1	1	1	1	1	2	1
9	1	1	1	1	1	1	1	1	2

Table 7.1 shows that for training composition 1, only the first feature of the cell has 2 membership functions and the remaining has only 1. For training composition 7, only the 7th feature has 2 membership functions and the remaining has only 1, and it goes on.

Table 7.2 Classification results for cross-validation between 1 – 8

Training Compositions According to Table 7.1	Classification Result (%)							
	97,35	96,76	96,18	97,35	97,94	96,76	95,88	97,06
	97,35	97,94	97,65	97,35	96,18	96,76	97,06	97,35
	97,06	96,18	96,76	96,76	97,94	97,06	97,65	97,35
	97,35	97,94	97,35	98,82	97,65	96,76	97,35	97,35
	97,35	97,94	97,06	97,35	97,94	96,76	97,06	96,18
	97,65	97,94	97,35	97,35	97,65	98,24	97,06	96,76
	96,18	95,88	95,29	95,29	96,47	96,47	96,18	97,06
	95,59	97,06	97,06	95,88	96,76	95,59	97,06	96,18
97,65	95,88	96,76	98,24	96,47	96,76	95,88	97,94	

Table 7.3 Classification results for cross-validation between 9 – 16

Training Compositions According to Table 7.1	Classification Result (%)							
	96,47	96,47	97,35	95,29	96,47	97,35	97,06	97,06
	97,94	96,76	97,94	97,94	97,06	97,35	97,06	97,35
	97,35	97,65	96,76	96,76	97,06	96,47	97,06	97,06
	97,06	97,06	97,65	98,24	96,18	96,76	97,06	97,35
	97,35	97,35	96,76	97,65	96,18	95,29	97,06	97,06
	97,06	97,35	96,76	97,06	98,53	97,94	97,65	97,06
	95,29	95,59	97,06	95,88	97,06	96,76	94,71	96,76
	96,47	97,35	96,18	97,06	96,47	95,88	96,76	98,24
97,35	96,76	97,06	96,18	97,94	96,76	97,65	97,06	

Table 7.4 Classification results for cross-validation between 17 – 24

Training Compositions According to Table 7.1	Classification Result (%)							
	95,88	97,06	96,47	97,06	97,35	96,76	97,65	96,18
	96,47	96,47	97,35	97,65	97,06	97,65	97,35	96,47
	96,76	97,94	97,94	97,65	95,29	97,35	97,35	97,94
	97,94	99,12	96,47	97,06	97,06	97,35	97,94	96,47
	97,06	96,47	96,47	97,94	97,06	97,06	98,82	96,76
	97,06	97,94	97,94	97,06	97,35	97,65	97,35	97,65
	96,47	95,59	97,06	95,88	95,59	96,47	96,18	95,88
	96,47	96,76	96,47	97,35	96,18	97,35	95,29	95,88
95,59	96,47	96,18	97,35	97,65	96,47	97,06	97,35	

Table 7.5 Classification results for cross-validation between 25 – 30 and the final results

	Classification Result (%)						Average Result (%)	Maximum Result (%)
Training Compositions According to Table 7.1	97,94	95,88	97,06	97,06	97,94	96,18	96,88	97,94
	97,35	96,18	97,35	97,35	98,24	97,94	97,30	98,24
	97,65	96,47	95,88	97,06	97,06	98,24	97,16	98,24
	97,06	96,76	98,24	97,94	95,59	97,06	97,34	99,12
	97,35	95,88	98,24	97,06	96,18	97,35	97,08	98,82
	97,06	96,18	96,76	96,76	98,53	96,76	97,41	98,53
	94,71	95,29	96,47	96,47	95,59	96,47	96,13	97,06
	95,59	95,59	96,47	95,88	97,65	96,47	96,61	98,24
	97,94	97,35	97,35	97,35	97,65	97,65	97,14	98,24

Table 7.2, 7.3, 7.4 and 7.5 represent the classification results for 9 different compositions with 30 times of cross-validation.

Table 7.6 shows the overall results for 2 rule method. Since we had 9 different compositions for 2 rule method and 30 times of cross-validation, we had 270 different tests.

Table 7.6 Overall results for 2 rule method of 270 tests

	Average Classification Result (%)	Maximum Classification Result (%)	Minimum Classification Result (%)
2 Rule	97,01	99,12	93,82

Table 7.7 shows the confusion matrix for 2 rule method out of 340 test data.

Table 7.7 Confusion matrix of one of the classifications for 2 rule method

		Predicted	
		Benign	Malignant
Classification	Negative	62	10
	Positive	4	264

7.1.1.2 4 Rule

Table 7.8 Different training compositions for 4 rule method

Training Compositions	Feature								
	1	2	3	4	5	6	7	8	9
1	2	2	1	1	1	1	1	1	1
2	2	1	2	1	1	1	1	1	1
3	2	1	1	2	1	1	1	1	1
4	2	1	1	1	2	1	1	1	1
5	2	1	1	1	1	2	1	1	1
6	2	1	1	1	1	1	2	1	1
7	2	1	1	1	1	1	1	2	1
8	2	1	1	1	1	1	1	1	2
9	1	2	2	1	1	1	1	1	1
10	1	2	1	2	1	1	1	1	1
11	1	2	1	1	2	1	1	1	1
12	1	2	1	1	1	2	1	1	1
13	1	2	1	1	1	1	2	1	1
14	1	2	1	1	1	1	1	2	1
15	1	2	1	1	1	1	1	1	2
16	1	1	2	2	1	1	1	1	1
17	1	1	2	1	2	1	1	1	1
18	1	1	2	1	1	2	1	1	1
19	1	1	2	1	1	1	2	1	1
20	1	1	2	1	1	1	1	2	1
21	1	1	2	1	1	1	1	1	2
22	1	1	1	2	2	1	1	1	1
23	1	1	1	2	1	2	1	1	1
24	1	1	1	2	1	1	2	1	1
25	1	1	1	2	1	1	1	2	1
26	1	1	1	2	1	1	1	1	2
27	1	1	1	1	2	2	1	1	1
28	1	1	1	1	2	1	2	1	1
29	1	1	1	1	2	1	1	2	1
30	1	1	1	1	2	1	1	1	2

To make the ANFIS process with 4 rules we gave 2 of the features 2 membership functions again in order such as we did in 2 rules method. A small demonstration of this composition is shown in table 7.8. Table 7.8 shows that for training composition 1, only the first 2 features of the cell have 2 membership functions and the remaining has only 1. For training composition 12, only the 2nd and 6th feature have 2 membership functions and the remaining has only 1, and it goes on.

Table 7.9 Classification results for cross-validation between 1 – 6 and the final results

	Classification Results (%)						Overall Average Results (%)	Overall Maximum Results (%)
	Training Compositions According to Table 7.8	97,35	96,47	97,65	97,35	96,47	98,82	97,55
96,47		97,94	96,18	98,24	96,76	97,65	97,18	98,24
97,35		96,47	97,06	96,47	96,47	97,94	97,05	97,94
97,06		96,47	98,24	96,18	96,18	96,18	96,79	99,12
97,06		96,76	96,76	97,35	96,47	96,18	97,37	99,12
97,94		96,76	97,06	97,06	96,76	97,65	96,92	98,24
97,65		95,59	97,06	95,59	96,47	97,06	96,99	98,53
96,76		97,06	97,35	97,65	97,35	95,88	96,79	98,82
97,06		97,35	97,35	96,76	97,06	97,94	97,13	97,94
97,94		96,47	95,88	96,47	98,24	97,65	97,09	98,53
96,76		97,35	96,76	97,06	96,18	97,06	96,91	98,24
97,35		96,18	96,18	97,06	97,94	96,18	96,88	97,94
97,06		97,94	97,65	97,94	97,06	95,29	97,18	98,53
96,76		97,94	97,06	97,35	96,18	96,18	97,25	98,53
98,24		97,35	97,35	95,88	98,24	96,47	97,33	98,53
97,06		97,65	96,47	96,76	96,18	96,76	97,25	98,53
96,76		98,24	97,65	97,35	97,35	97,35	97,00	98,82
97,65		96,76	97,35	97,06	97,06	97,94	97,04	98,82
97,06		96,47	97,06	95,00	95,88	97,35	96,52	97,94
96,47		95,88	97,35	96,47	98,24	96,76	96,79	98,24
98,24		95,29	96,76	96,47	96,18	96,47	96,80	98,24
95,88		95,88	97,65	97,06	97,06	97,65	96,97	98,24
98,82		97,94	97,35	95,00	96,18	97,65	97,14	98,82
97,65		96,18	97,06	96,18	95,88	97,35	96,76	98,24
97,35		96,18	97,65	96,47	97,06	97,35	96,89	98,53
98,24		96,47	95,59	97,06	95,88	97,35	96,60	98,24
98,24		97,94	98,24	97,65	97,65	97,94	97,42	98,53
96,18		97,06	96,47	96,76	97,94	96,47	96,58	98,24
98,24		96,18	97,35	95,88	97,06	97,06	96,72	98,24
96,76		96,47	97,65	95,29	96,76	96,18	96,86	98,24
95,59	95,88	96,47	97,65	98,24	97,06	96,74	98,53	
97,35	96,18	98,24	97,94	96,76	98,24	97,57	98,82	
94,41	96,47	97,06	96,76	95,00	97,65	96,71	98,53	
97,65	98,24	96,76	97,65	96,47	98,24	97,13	98,53	

Just like in the method of 2 rules, we had 30 epochs to be sure of the process. Since it will be too much to show all the epochs' results here, we gave only first 6 epoch results with all the combinations of 4 rule method in table 7.9. Also the overall average and the overall maximum results of all of the epochs are given in table 7.9.

Table 7.10 shows the overall results for 4 rule method. Since we had 36 different compositions for 4 rule method and 30 times of cross-validation, we had 1080 different tests.

Table 7.10 Overall results for 4 rule method of 1080 tests

	Average Classification Result (%)	Maximum Classification Result (%)	Minimum Classification Result (%)
4 Rule	96,93	99,12	92,35

Table 7.11 shows the confusion matrix for 4 rule method out of 340 test data.

Table 7.11 Confusion matrix of one of the classifications for 4 rule method

		Predicted	
		Benign	Malignant
Classification	Negative	70	2
	Positive	10	258

7.1.1.3 8 Rule

For 8 rule method we assigned 2 membership functions to 3 of the features out of 9. There occur 84 different combinations of these 8 rule method. In order not to cover a lot of pages, we only gave a small number of these combinations.

Table 7.12 shows that for training composition 1, only the first 3 features of the cell have 2 membership functions and the remaining has only 1. For training composition 17, only the 1st, 4th and 8th feature have 2 membership functions and the remaining has only 1, and it goes on.

Table 7.12 Different training compositions for 8 rule method

Training Compositions	Feature								
	1	2	3	4	5	6	7	8	9
1	2	2	2	1	1	1	1	1	1
2	2	2	1	2	1	1	1	1	1
3	2	2	1	1	2	1	1	1	1
4	2	2	1	1	1	2	1	1	1
5	2	2	1	1	1	1	2	1	1
6	2	2	1	1	1	1	1	2	1
7	2	2	1	1	1	1	1	1	2
8	2	1	2	2	1	1	1	1	1
9	2	1	2	1	2	1	1	1	1
10	2	1	2	1	1	2	1	1	1
11	2	1	2	1	1	1	2	1	1
12	2	1	2	1	1	1	1	2	1
13	2	1	2	1	1	1	1	1	2
14	2	1	1	2	2	1	1	1	1
15	2	1	1	2	1	2	1	1	1
16	2	1	1	2	1	1	2	1	1
17	2	1	1	2	1	1	1	2	1
18	2	1	1	2	1	1	1	1	2
19	2	1	1	1	2	2	1	1	1
20	2	1	1	1	2	1	2	1	1
21	2	1	1	1	2	1	1	2	1
22	2	1	1	1	2	1	1	1	2
23	2	1	1	1	1	2	2	1	1
24	2	1	1	1	1	2	1	2	1
25	2	1	1	1	1	2	1	1	2
26	2	1	1	1	1	1	2	2	1
27	2	1	1	1	1	1	2	1	2
28	2	1	1	1	1	1	1	2	2
29	1	2	2	2	1	1	1	1	1
30	1	2	2	1	2	1	1	1	1

Also the results of all of the combinations with 30 epochs take to much paper. We gave a small amount of the result for demonstration in. Table 7.13 shows the classification results of only 24 combinations and 6 epochs. Also the overall average and the overall maximum results of all of the epochs are given in table 7.13.

Table 7.13 Classification results for cross-validation between 1 – 6 and the final results of 24 combinations

	Classification Results (%)						Overall Average Results (%)	Overall Maximum Results (%)
	Training Compositions According to Table 7.12	97,65	97,06	96,47	97,06	95,59	97,65	96,66
95,88		97,06	96,18	95,88	96,47	95,59	96,70	98,24
97,35		98,24	96,47	96,18	98,82	96,76	96,87	98,82
96,18		97,94	96,47	97,65	95,00	97,94	97,04	98,53
97,35		95,88	97,06	97,35	96,76	96,76	96,86	98,24
95,59		97,94	97,06	96,47	98,82	97,94	97,38	99,41
96,47		96,18	95,00	96,18	96,18	95,59	96,00	97,65
97,06		95,00	95,29	97,06	97,65	97,35	96,89	97,94
96,47		96,76	95,00	96,18	96,18	96,76	96,56	97,65
97,06		95,59	96,47	96,47	96,76	97,35	96,61	98,24
96,76		96,76	96,47	96,47	97,35	96,18	96,43	98,24
97,65		97,35	97,65	96,76	97,65	97,06	97,04	98,53
94,71		96,76	95,29	95,88	96,18	96,76	95,88	97,94
96,18		97,06	97,06	96,76	96,47	95,88	96,57	97,94
96,76		97,94	95,00	97,94	95,29	94,71	96,61	98,24
97,06		97,65	95,88	96,47	96,76	98,53	96,94	98,53
97,06		96,47	96,18	95,59	97,94	95,88	97,10	98,53
95,00		95,59	95,29	94,41	97,06	95,00	95,66	97,35
96,18		97,65	97,35	97,65	96,47	97,65	97,11	98,24
96,18		97,94	95,00	94,71	97,06	96,76	96,31	97,94
95,59		97,94	97,06	96,76	96,18	95,29	96,54	97,94
95,59		95,00	96,18	95,59	95,00	93,53	95,19	97,65
97,35		97,06	96,76	98,24	96,47	96,18	97,02	98,24
95,88		97,06	98,24	97,65	97,06	96,47	97,25	98,53

Table 7.14 Overall results for 8 rule method of 2520 tests

	Average Classification Result (%)	Maximum Classification Result (%)	Minimum Classification Result (%)
8 Rule	96,27	99,41	92,06

Table 7.14 shows the overall results for 8 rule method. Since we had 84 different compositions for 4 rule method and 30 times of cross-validation, we had 2520 different tests.

Table 7.15 shows the confusion matrix for 8 rule method out of 340 test data.

Table 7.15. Confusion matrix of one of the classifications for 8 rule method

		Predicted	
		Benign	Malignant
Classification	Negative	56	16
	Positive	3	265

7.1.1.4 16 Rule

From now on we will not give all the classification results, since there are too much of them. Only the first training composition, ANFIS info, overall average, overall maximum and overall minimum results, and a confusion matrix of all of the combinations and all 30 epochs (confusion matrix of all of the tests) will be shown.

Table 7.16 shows only the first combination of 126 different combinations.

Table 7.16 First training composition for 16 rule method

Training Compositions	Feature								
	1	2	3	4	5	6	7	8	9
1	2	2	2	2	1	1	1	1	1

Table 7.17 shows the overall results for 16 rule method. Since we had 126 different compositions for 16 rule method and 30 times of cross-validation, we had 3780 different tests.

Table 7.17 Overall results for 16 rule method of 3780 tests

	Average Classification Result (%)	Maximum Classification Result (%)	Minimum Classification Result (%)
16 Rule	92,90	97,06	86,18

Table 7.18 shows the confusion matrix for 16 rule method out of 340 test data.

Table 7.18. Confusion matrix of one of the classifications for 16 rule method

		Predicted	
		Benign	Malignant
Classification	Negative	58	14
	Positive	8	260

7.1.1.4 32 Rule

Table 7.19 shows only the first combination of 126 different combinations.

Table 7.19 First training composition for 32 rule method

Training Compositions	Feature								
	1	2	3	4	5	6	7	8	9
1	2	2	2	2	2	1	1	1	1

Table 7.20 shows the overall results for 32 rule method. Since we had 126 different compositions for 32 rule method and 30 times of cross-validation, we had 3780 different tests.

Table 7.20 Overall results for 32 rule method of 3780 tests

	Average Classification Result (%)	Maximum Classification Result (%)	Minimum Classification Result (%)
32 Rule	88,27	93,82	80,88

Table 7.21 shows the confusion matrix for 32 rule method out of 340 test data.

Table 7.21 Confusion matrix of one of the classifications for 32 rule method

		Predicted	
		Benign	Malignant
Classification	Negative	48	24
	Positive	11	257

Table 7.26 shows the overall results for 128 rule method. Since we had 36 different compositions for 128 rule method and 30 times of cross-validation, we had 1080 different tests.

Table 7.26 Overall results for 128 rule method of 1080 tests

	Average Classification Result (%)	Maximum Classification Result (%)	Minimum Classification Result (%)
128 Rule	89,84	95,88	80,41

Table 7.27 shows the confusion matrix for 128 rule method out of 340 test data.

Table 7.27 Confusion matrix of one of the classifications for 128 rule method

		Predicted	
		Benign	Malignant
Classification	Negative	46	26
	Positive	2	266

7.1.1.7 256 Rule

Table 7.28 shows only the first combination of 9 different combinations.

Table 7.28 First training composition for 256 rule method

Training Compositions	Feature								
	1	2	3	4	5	6	7	8	9
1	2	2	2	2	2	2	2	2	1

Table 7.29 shows the overall results for 256 rule method. Since we had 9 different compositions for 256 rule method and 30 times of cross-validation, we had 270 different tests.

Table 7.29 Overall results for 256 rule method of 270 tests

	Average Classification Result (%)	Maximum Classification Result (%)	Minimum Classification Result (%)
256 Rule	90,88	95,88	81,22

Table 7.30 shows the confusion matrix for 256 rule method out of 340 test data.

Table 7.30 Confusion matrix of one of the classifications for 256 rule method

		Predicted	
		Benign	Malignant
Classification	Negative	57	15
	Positive	10	258

7.1.2.3 Membership Function Compositions

7.1.2.1.3 Rule

Table 7.31 shows only the first combination of 9 different combinations.

Table 7.31 First training composition for 3 rule method

Training Compositions	Feature								
	1	2	3	4	5	6	7	8	9
1	3	1	1	1	1	1	1	1	1

Table 7.32 shows the overall results for 3 rule method. Since we had 9 different compositions for 9 rule method and 30 times of cross-validation, we had 270 different tests.

Table 7.32 Overall results for 3 rule method of 270 tests

	Average Classification Result (%)	Maximum Classification Result (%)	Minimum Classification Result (%)
3 Rule	96,81	98,53	93,23

Table 7.33 shows the confusion matrix for 3 rule method out of 340 test data.

Table 7.33 Confusion matrix of one of the classifications for 3 rule method

		Predicted	
		Benign	Malignant
Classification	Negative	65	7
	Positive	6	262

7.1.2.2 9 Rule

Table 7.34 shows only the first combination of 36 different combinations.

Table 7.34 First training composition for 9 rule method

Training Compositions	Feature								
	1	2	3	4	5	6	7	8	9
1	3	3	1	1	1	1	1	1	1

Table 7.35 shows the overall results for 9 rule method. Since we had 36 different compositions for 9 rule method and 30 times of cross-validation, we had 1080 different tests.

Table 7.35 Overall results for 9 rule method of 1080 tests

	Average Classification Result (%)	Maximum Classification Result (%)	Minimum Classification Result (%)
9 Rule	95,62	98,82	89,11

Table 7.36 shows the confusion matrix for 9 rule method out of 340 test data.

Table 7.36 Confusion matrix of one of the classifications for 9 rule method

		Predicted	
		Benign	Malignant
Classification	Negative	64	8
	Positive	6	262

7.1.2.3 27 Rule

Table 7.37 shows only the first combination of 84 different combinations.

Table 7.37 First training composition for 27 rule method

Training Compositions	Feature								
	1	2	3	4	5	6	7	8	9
1	3	3	3	1	1	1	1	1	1

Table 7.38 shows the overall results for 27 rule method. Since we had 84 different compositions for 27 rule method and 30 times of cross-validation, we had 2520 different tests.

Table 7.38 Overall results for 27 rule method of 2520 tests

	Average Classification Result (%)	Maximum Classification Result (%)	Minimum Classification Result (%)
27 Rule	90,16	96,47	82,13

Table 7.39 shows the confusion matrix for 27 rule method out of 340 test data.

Table 7.39 Confusion matrix of one of the classifications for 27 rule method

		Predicted	
		Benign	Malignant
Classification	Negative	46	26
	Positive	5	263

7.1.2.4 81 Rule

Table 7.40 shows only the first combination of 126 different combinations.

Table 7.40 First training composition for 81 rule method

Training Compositions	Feature								
	1	2	3	4	5	6	7	8	9
1	3	3	3	3	1	1	1	1	1

Table 7.41 shows the overall results for 81 rule method. Since we had 126 different compositions for 81 rule method and 30 times of cross-validation, we had 3780 different tests.

Table 7.41 Overall results for 81 rule method of 3780 tests

	Average Classification Result (%)	Maximum Classification Result (%)	Minimum Classification Result (%)
81 Rule	89,47	97,35	84,15

Table 7.42 shows the confusion matrix for 81 rule method out of 340 test data.

Table 7.42 Confusion matrix of one of the classifications for 81 rule method

		Predicted	
		Benign	Malignant
Classification	Negative	56	16
	Positive	9	259

7.2 FIS Classification

No training data used; only the rules are created. Table 7.43 shows the FIS classification result.

Table 7.43 FIS classification result

	Result (%)
FIS Classification	96.48

Table 7.44 shows the confusion matrix for FIS method out of 683 test data.

Table 7.44 Confusion matrix of FIS classification

		Predicted	
		Benign	Malignant
Classification	Negative	225	17
	Positive	7	434

7.3 KNN Classification

For the KNN classification method we used 1-nearest neighbor, 3-nearest neighbor, 5-nearest neighbor, 7-nearest neighbor and 9-nearest neighbor classifications. As we did in ANFIS classification, we used 30 epochs to make the program decisive. For all of the classifications we used 400 train and 283 test data. To make the prior probabilities same, we used 200 benign and 200 malignant instances. Below in table 7.45 we presented these results in average, maximum and minimum classification results.

Table 7.45 Results for KNN classification method

	1-nearest (%)	3-nearest (%)	5-nearest (%)	7-nearest (%)	9-nearest (%)	Overall (%)
Average	96,77	97,42	97,81	97,63	97,27	97,38
Max	97,87	99,29	98,94	98,93	98,23	99,29
Min	94,69	95,40	96,81	96,11	95,40	94,69

Table 7.46 shows the confusion matrix for KNN classification method out of 283 test data.

Table 7.46 Confusion matrix of one of the KNN method classifications

		Predicted	
		Benign	Malignant
Classification	Negative	39	3
	Positive	6	235

7.4 Bayes Classification

For all of the classifications we used 400 train and 283 test data. To make the prior probabilities same, we used 200 benign and 200 malignant instances.

Table 7.47 shows the overall results for Bayes classification method of 30 tests.

Table 7.47 Overall results for Bayes classification method of 30 tests

	Average Classification Result (%)	Maximum Classification Result (%)	Minimum Classification Result (%)
Bayes Classification	94,07	97,17	90,10

Table 7.48 shows the confusion matrix for Bayes classification method out of 283 test data.

Table 7.48 Confusion matrix of Bayes classification

		Predicted	
		Benign	Malignant
Classification	Negative	41	23
	Positive	1	218

CHAPTER EIGHT

CONCLUSION

The main classification method of this research is adaptive neuro – fuzzy inference system classification. There are two method that ANFIS learning employs for updating membership function parameters: back propagation for all parameters (a steepest descent method), and a hybrid method consisting of back propagation for the parameters associated with the input membership functions. When we applied back propagation method, because of the sharp slope for the membership functions, we learned that it was not proper for our classification of WBCD. Then we decided to use hybrid model.

ANFIS is capable of having many rules if needed. For 9 attributes we could have more than 2^9 rules, having 2 membership functions for all attributes. Moreover we could even have more than 3^9 rule, having 3 membership functions for all attributes, and so on. Besides, we could assign for example 2 membership functions for the attribute 1, 3 membership functions for the attribute 2 and any number of membership functions for any attribute. By this way we were able to give as much importance as necessary for any attribute. We then realized that when we increased the rule number we only made the program run slower, because calculation of all the rules was a time consuming process. Having less rules we had the results got nearly the same as the ones having more rules. Because ANFIS was able to give satisfying results even though we used very small number of rules. Epoch number of adaptive training is also changeable. For our first attempts of classification we used more than 10 epochs to get good classification results. Then we gradually decreased the epoch number in order to run the program faster. We noticed that ANFIS was again able to create rules capable of classifying with a good percentage even if we used 1 epoch for training.

For training purpose we had to use some of the data and the remaining data for the test of the program. Since we did not know the prior probabilities of the benign and

malignant cells, we gave the same prior probability 0.5 for each of them. This method is the same for all of the classification methods mentioned in this project.

For the FIS we created our own rules according to the data distribution. To be able to make proper classification we investigated the data in detail. As mentioned in chapter 4 we created 4 rules. While using FIS editor we were again able to change the epoch number of the system, but increasing it was not necessary. For our FIS classification we accomplished 96.48 % of true classification rate which was very satisfying.

This method had its pros and cons. For any classification of any data we are allowed to create our own rules according to the data. We can easily change the membership functions of any input and output. We can also change the rule number, the number of the attributes contained in one of the rules. But for any data that we are not allowed to investigate the data distribution, it is impossible to have proper rules. Because this system is not a learning system. All the steps except calculation process are done by the programmer. So this method can be used for getting accustomed to the easy process of FL, flexibility of FL and the linguistic way of programming of the FL, and so on. This FIS helped us to understand the FL in practical.

When we came to the ANFIS, it was a little complicated than the FIS. We were again able to change the parameters of the system very easily but in a limited way. We were not allowed to use any of the membership functions that we mentioned in chapter 2. Some of them were not suitable for adaptive learning. Also the methods for fuzzy system Sugeno and Mamdani type were not allowed to choose, we can only use Sugeno method for adaptive system. As mentioned earlier we used several different numbers of epochs and we had a decision to reduce it to 1 epoch, since we got really satisfying results. We were allowed to change the error rate of the system. After our lots of experiments we decided to choose the error rate not so small making the program run slower, but slower enough to give good results of classification.

The method of our membership function employment to the features was changing according to the rule number. When we look into 2 rules, we were giving 2 membership functions to one of the attributes and 1 membership function to the remaining ones. The combination of this distribution gave us 2 rules. But in order to give 2 membership functions to all of the features we used a diagonal matrix having 2 in the diagonal, and 1 for the other locations. For 4 rules we did the same process. We had 2 features of having 2 membership functions and 1 for the remaining, and again just in the 2 rule method we had to give 2 membership functions in order. But in this method we had 36 different combinations. So our system worked a little slower. For the other methods, we had 84 different combinations for 8 rules, 126 different combinations for 16 rules, 126 different combinations for 32 rules, 84 different combinations for 64 rules, 36 different combinations for 128 rules and 9 different combinations for 256 rules. We did not have more than 256 rules because our computer was not able to give results before 24 hours for more than 256 rules.

For the training process we had 170 benign and 170 malignant cells in order to have same prior probabilities as mentioned in the result chapter. We also changed the data chosen for training by 30 times of cross validation in order to have all of the data to be in the training part and also to see if there were classification differences between these epochs. But all the results for the chosen number of rule were close to each other.

When we come the results, we could easily say that ANFIS was able give good classification a result no matter the rule number was. There was a slight difference between them but the overall rate was very high. For the best classification we had 99.41 % in 8 rule classification and best average of 97.01 % in 2 rule method. By looking at the results we can say that for Wisconsin breast cancer data it is not necessary to have lots of rules to get good results. Also by having less number of rules we have our system run faster. So for the Wisconsin breast cancer data it is recommended to have rules between 2 and 8.

If we compare our system for other methods we experimented we can again say that Bayes classification was slightly worse than ANFIS classification with a rate of 97.17 % for the best and 94.07 for the average. The k-nearest neighbor classification method was nearly same as ANFIS with a rate of 99.29 % for the best and 97.38 for the average. But the flexibility of the ANFIS is indisputable.

For the improvement of the system a higher technology for the computer to be able use more membership functions is needed. It can be seen that for the first 2, 4 and 8 rule methods the system has a very high classification rate. But for the 16, 32 and 64 rule methods it has slightly worse classification rate than the preceding ones. When we increased the rule number the system again started to rise is classification success up to 256 rules. Further experiments can be done using high number of rules and high number of train data to train the system better.

A wise progression can be done by using very large amount of data for training. Our system was trained with a very limited number of data. To increase the classification success could be done by this way. Surely, with a well trained and well equipped fuzzy system, the diagnosing process of this disease will be used in the future projects with high efficiency.

REFERENCES

- Abonyi, J., Roubos, J. A., & Szeifert, F. (2003). Data – Driven Generation of Compact, Accurate and Linguistically – Sound Fuzzy Classifier Based on a Decision – Tree Initialization. *International Journal of Approximate Reasoning*, 32 (1), 1-21, Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.18.6853>
- Anonymous, (1999), *Matlab – Fuzzy Logic Toolbox User’s Guide* (2nd ed.) Natick: MathWorks
- Duda, R. O., Hart, P. E., Stork, D. G., (2001). *Pattern Classification* (2nd ed). Canada: Wiley
- Franc, V., & Hlavac, V. (2004). Bayesian Classifier. In *Statistical Pattern Recognition Toolbox for Matlab* (1st ed.), 78–79, Prague: Czech Technical University.
- Imaginis (1999), *What is breast cancer*. Retrieved June 20, 2009 from http://www.imaginis.com/breasthealth/breast_cancer.asp
- Jain, R., & Abraham, A. (2003). A comparative study of fuzzy classification methods on breast cancer data. *Australasian physical & engineering sciences in medicine*, 27 (4), 213 - 218 Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/15712589>
- Jang, J. S. R., Sun, C. T. (1997). *Neuro – Fuzzy and Soft Computing: A computational Approach to Learning and Machine Intelligence*, (1st ed.). Prentice Hall.
- Kasabov, N., K., (1998). *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering* (2nd ed.). London: MIT.

- Keles, A., Hasiloglu, A.S., Keles, A., & Aksoy, Y. (2007). Neuro-fuzzy classification of prostate cancer using NEFCLASS-J. *Computers in Biology and Medicine*, 37 (11), 1617–1628. Retrieved from <http://portal.acm.org/citation.cfm?id=1288045>
- Kıyan, T., & Yıldırım, T. (2004). Breast Cancer Diagnosis using statistical neural networks. *Istanbul University – Journal of Electrical & Electronics Engineering*, 4 (2), 1149 – 1153. Retrieved from <http://www.istanbul.edu.tr/eng/ee/jeee/main/pages/issues/is42/42007.pdf>
- Klir, G., J. (2006). *Uncertainty and information: foundations of generalized information theory*. (1st ed.). 268 – 269, New Jersey: Wiley
- Lee, H.M., Chen, C.M., Chen, J.M., & Jou, Y.L. (2001). An efficient fuzzy classifier with feature selection based on fuzzy entropy. *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, 31 (3),. Retrieved from http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=931536
- Liu , D., Fei, S., Hou, Z., Zhang, H., Sun, C., (2007). *Advances in Neural Networks, 4th symposium*, 151 – 178. New York: Springer
- Lorenz, A., Blüm, M., Ermert, H. & Senge, T. (1997). Comparison of Different Neuro – Fuzzy Classification Systems for the Detection of Prostate Cancer in Ultrasonic Images. *IEEE Ultrasonic Symposium*, 2, 1201 – 1204. Retrieved from http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=661794
- Mangasarian, O. L. & Wolberg, W. H., (1990). *Cancer diagnosis via linear programming*, 23 (5),1 – 18, SIAM News.

- Marwala, T., Tettey, T., & Chakraverty, S. (2006). Fault Classification using Pseudomal Energies and Neuro – Fuzzy Modeling. *In Proceedings of the Asia-Pacific Workshop on Structural Health Monitoring*. Retrieved from <http://arxiv.org/abs/0705.2236>
- Nauck, D., Nauck, U., & Kruse, R. (1996). Generating Classification Rules with the Neuro – Fuzzy System NEFCLASS. *Fuzzy Information Processing Society Biennial Conference of the North American*. Braunschweig: Retrieved from http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=534779.
- Pena – Reyes, C. A., Sipper, M., (1998) A fuzzy – genetic approach to breast cancer diagnosis. *Artificial Intelligence in Medicine*, 17 (2), 131-155. Retrieved from <http://linkinghub.elsevier.com/retrieve/pii/S09333365799000196>
- Saritas, I., Allahverdi, N., & Sert, I., U. (2003). A fuzzy expert system design for diagnosis of prostate cancer. *International Conference on Computer Systems and Technologies. Proceedings of the 4th international conference conference on Computer systems and technologies*, 345 – 351. Retrieved from <http://doi.acm.org/10.1145/973620.973677>
- Scholarpedia (2009). *K-neraest neighbor*. Retrieved June 23, 2009 from http://www.scholarpedia.org/article/K-nearest_neighbor
- Setiono, R. (1999). Generating concise and accurate classification rules for breast cancer diagnosis. *Artificial Intelligence in Medicine*, 18, 205 – 219. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.45.9593>
- Smarandache, F., Christianto, V., (2005). *Multi-Valued Logic, Neutrosophy, and Schrödinger Equation*. (1st ed.). 10 – 15. Phoenix: Hexis.

Stanford Encyclopedia of Philosophy (2006). *Fuzzy Logic*. Retrieved June 25, 2009 from <http://plato.stanford.edu/entries/logic-fuzzy/>

Wang, J. S., & Lee, C. S. G. (2002). Self-Adaptive Neuro-Fuzzy Inference System for Classification Applications. *IEEE Transactions on Systems, December, 10* (6), 790 – 802. Retrieved from http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1097778