



MASTER THESIS INFORMATICS & ECONOMICS

---

# Model Selection Under Sampling Uncertainty Using Fuzzy Numbers

---

*Author:*

Bei WEN

ID: 309210

*Supervisor:*

Uzay KAYMAK

Rob POTHARST

October 1, 2009

## Abstract

Model selection is one of the fundamental tasks of scientific inquiry. The most widely used methods such as ROC analysis do not take sampling uncertainty into account. To improve the robustness of model selection, we develop a model selection method capable to incorporate sampling uncertainty. We capture the sampling uncertainty by using the bootstrap technique, and quantify the sampling uncertainty by introducing fuzzy numbers. We apply our model selection system to a variety of real-world databases with respect to binary classifications. Among the tested datasets, our method performs in line with the traditional ROC analysis, whereas it provides the fuzzy presentation of ROC curves based on which not only the predictive accuracy but also the degree of sampling uncertainty can be addressed. In addition, we develop a computer tool implementing our system, which eases the tedious procedures in model selection.

**Keywords:** model selection, sampling uncertainty, binary classification, bootstrap, ROC analysis, fuzzy numbers, the comparison of fuzzy numbers

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>The background of model selection and the introduction of fuzzy numbers</b>	<b>7</b>
2.1	Model selection on classification systems . . . . .	7
2.2	Predictive performance evaluating techniques . . . . .	8
2.3	The receiver operating characteristic (ROC) curve . . . . .	9
2.4	The introduction of fuzzy set theory . . . . .	11
2.5	The method for comparing fuzzy numbers . . . . .	12
<b>3</b>	<b>Methodology to incorporate sampling uncertainty into model selection</b>	<b>15</b>
3.1	The architecture of the model selection system . . . . .	15
3.2	The algorithm of the model selection system . . . . .	18
3.3	The method for capturing sampling uncertainty - Bootstrap .	19
3.4	The method for mapping fuzzy intervals with determined membership functions . . . . .	20
3.5	The method for comparing fuzzy numbers . . . . .	25
3.6	The method for forming AUC fuzzy numbers . . . . .	27
<b>4</b>	<b>Experimental design</b>	<b>33</b>
4.1	Real-life data sets . . . . .	33
4.2	Selected models . . . . .	38
4.2.1	Logistic Regression (LR) . . . . .	38
4.2.2	Fuzzy Inference System (FIS) . . . . .	39

4.2.3	Fuzzy C-means Clustering (FCM) . . . . .	41
<b>5</b>	<b>Experimental results</b>	<b>43</b>
5.1	Dutch charity database . . . . .	43
5.1.1	Ranking TPR fuzzy intervals . . . . .	43
5.1.2	Ranking AUC fuzzy intervals . . . . .	52
5.2	Haberman's survival database . . . . .	60
<b>6</b>	<b>Conclusion and discussion</b>	<b>67</b>
6.1	Main findings . . . . .	67
6.2	Future research . . . . .	69
<b>A</b>	<b>Matlab scripts</b>	<b>77</b>
A.1	Main script . . . . .	77
A.2	The function of obtaining TPRs and FPRs for different classifiers	92
A.3	The function of plotting figures . . . . .	94

# Chapter 1

## Introduction

With the increasing modern computer power and the expansion of computational methods, researchers are more and more frequently confronted with the problem of selecting a particular prediction algorithm efficiently among many alternatives, while suffering uncertainties from sampling.

### **Model selection**

Model selection, one of the fundamental tasks of scientific inquiry, is the process of identifying the best approximating model among alternatives. To determine the principle behind a series of observations is often linked directly to a mathematical model predicting those observations.

Many predictive models have been proposed to predict the output from the input, given training data consisting of input-output pairs. In addition to traditional linear models, scientists have developed a class of flexible nonlinear models such as artificial neural networks, fuzzy inference systems, classification and regression trees, and kernel methods in the field of machine learning and data mining. A crucial question is how to evaluate the fitness and robustness of these models and select the best model accordingly.

## Sampling uncertainty

Uncertainty from sampling can be generated via various sources. Some sampling errors are due to the institutional or distributional heterogeneity. The former refers to the uncertainty of measurement. The later occurs when the samples are not randomly selected. The issue of measuring uncertainty from sampling becomes increasingly important because worldwide data nowadays can be relatively easily collected via internet and pooled for analysis with little quality control.

Our study concentrates on “sampling uncertainty”, one of uncertainties from sampling. As the fact, we cannot usually analyze the whole population, because it would be extremely expensive. We need to take a sample, a portion of the population that is enough to be computational analyzed. Additionally, it is the most common situation in data analysis where researchers need to make a decision based on the result of analyzing a smaller-sized data set from original large data sets due to computational complexity and cost. We assume that models essentially derived based on a smaller-sized data set from a large population or a large database often have poor performance, and the bias arisen from any single random data set contributes to so-called “sampling uncertainty”.

## Research scope

The work focuses on the model selection with respect to binary classification problems, where multi-class dependent variables are dichotomized to one vs. the remaining classes. The algorithm is implemented for solving real-world problems in the domain of economics, medical and management science.

## Research question

The research question is how to incorporate sampling uncertainty into model selection.

Uncertainty characteristics are identified as *fuzziness* treated on the basis of the fuzzy set theory. In quite a few applications of fuzzy set theory to deci-

sion making, we are confronted with the problem to select the best one from a collection of possible solutions. It has been addressed in connection with introducing *fuzziness* into some special types of classical decision problem for making a choice among the alternatives, and dealt with independently of any specific application. In our study, such connection is introduced for model selection problems.

Therefore, the objective of our study is to develop a tool to present and analyze *fuzziness* with respect to predictive performance, so that it can efficiently select the best model among alternatives under sampling uncertainty.

## Methodology

To achieve the aim, we propose a fully automatic and computationally efficient method to select the best prediction model taking sampling uncertainty into account by introducing (1) the bootstrap technique to capture the sampling uncertainty and (2) the fuzzy numbers to quantify the sample uncertainty.

In our model selection system, the receiver operating characteristic (ROC) curve is used as the measure of predictive performance. Bootstrap is used for capturing sampling uncertainty. We map the measurement variation of predictive performance into fuzzy intervals with determined membership function, including “the regular crisp interval function”, “the triangular mean function”, “the triangular median function”, “the trapezoidal function with mean and standard deviation”, and “the fitting function of envelope”. Some widely used techniques proposed by Yager are used for ranking the fuzzy intervals, including so-called “center of gravity” and “mean area” method. Therefore, to determine which model is best is based on the resulting outcome of comparing fuzzy intervals. According to “fuzzy resolution principle”, we address how to connect the fuzzy intervals of true positive rates with the overall performance, represented by the fuzzy number of the area under the ROC curve (AUC).

**Overview**

The following thesis is organized with Chapter 2 introducing the theoretical background of the model selection and the relevant fuzzy set theory. Chapter 3 outlines briefly the framework and algorithm of our model selection approach firstly, and presents in details the method used to capture sampling uncertainty, the method used for measuring model performance, the method used for mapping the variation from sampling uncertainty into fuzzy intervals, the method based on which we compare the fuzzy intervals, as well as the method for associating with a global AUC fuzzy interval. In Chapter 4, we describe the experimental designs including which real-life databases to be tested, which models we choose to predict outcomes, as well as which measures we adopt to improve the system. We illustrate the results accordingly in Chapter 5. Finally, we come up with some conclusions and discussions in Chapter 6.



## Chapter 2

# The background of model selection and the introduction of fuzzy numbers

### 2.1 Model selection on classification systems

Classification systems have been built traditionally by experimenting with many different classifiers, comparing their performance and choosing the best accordingly. Experiments with different inductive algorithms, parameter settings, and training regimes yield a large number of classifiers to be evaluated and compared. Unfortunately, comparison is often difficult in real world environments because key parameters of the target environment are not known, and the optimal cost & benefit tradeoffs and the target class priors seldom are known precisely and often are subject to change [7][28][40][50][64].

Many algorithms have been proposed to predict the output from the input, given training data consisting of input-output pairs. In addition to traditional linear models, scientists have developed a class of flexible nonlinear models such as artificial neural networks, fuzzy inference systems, classification and regression trees, and kernel methods in the field of machine learning and data mining. The application of a technique depends on its practical use and the purpose of the model. If the structure of the database is un-

known, whatsoever, non-linear methods or even non-parametric methods can be useful.

An important question is how to select the best model under the uncertainties from sampling. Complex models may fit the data better than simple models, but often require additional uncertain assumptions to be made implicitly. A good model selection technique should incorporate uncertainties and balance goodness of fit with simplicity [38].

## 2.2 Predictive performance evaluating techniques

Evaluating the fitness of models is one of the key issues which has been investigated extensively in literature. David M W Powers discussed several commonly used evaluation measures from *Precision*, *Recall*, and *F-Factor*, to *ROC*, *Informedness*, *Markedness* and *Correlation* [48]. Pierre Baldi et al. provided a unified overview of measures that are used to assess the accuracy of prediction algorithms for classification [6]. These measures are mostly based on confusion matrix, from raw percentages, quadratic error measures and other distances, and correlation coefficients, to information theoretic measures such as relative entropy and mutual information [6]. They also briefly discussed the advantages and disadvantages of each approach. Recently, in the work of C. Ferri et al., they analyzed experimentally the behavior of 18 different performance measures in 30 data sets, and tried to identify clusters and relationship between these measures [26]. Moreover, the enthusiasm for information criterion has burgeoned in the past three decades [1][2].

Different measures have been defined in extant literature with the aim of making better model selection in general or for a specific application area. One of the most widely used measures is the receiver operating characteristic (ROC) curve. There is a huge literature associated with ROC curves, and the applications spread across medicine, statistics, and most recently, machine learning and data mining [10][24][25][33][44][45][55].

## 2.3 The receiver operating characteristic (ROC) curve

The ROC curve was first used during World War II and was developed in the 1950's [23]. Since 1997, the idea of using ROC curves has been brought in the area of machine learning by Foster Provost and Tom Fawcett from the previous applications [49].

ROC curves have been employed in popularity as a useful measure of predictive performance and visualization tool for analyzing trade-offs between true positives and false positives in binary class scenarios. ROC curves are a graphical plot of the true positive rate (*sensitivity*) vs. false positive rate ( $1 - \textit{specificity}$ ) for a binary classifier system as its discrimination threshold is varied.

At each false positive rate, a two by two confusion table can be derived. A confusion table cross-tabulates the observed classes and the classes that a model predicts. With the observed classes and the predicted classes from the data used to build the predictive classification, the table can be used to understand how skillful the model has been in fitting the data. Therefore, a confusion table is a critical diagnostic tool and contains important information which can easily be extracted. As shown in Table 2.1, the four cells in the confusion table represent the number of correctly predicted positives, or true positives (TP), the number of correctly predicted negatives, or true negatives (TN), the number of wrongly predicted positives, or false positives (FP), and the number of wrongly predicted negatives, or false negatives (FN). Then the true positive rate represented by the Y-axis is calculated as,

$$\textit{True Positive Rate (TPR)} = \textit{Sensitivity} = \frac{TP}{TP + FN} \quad (2.1)$$

at each false positive rate represented by the X-axis,

$$\textit{False Positive Rate (FPR)} = 1 - \textit{Specificity} = \frac{FP}{TN + FP} \quad (2.2)$$

		Predicted Value	
		1	0
Observed Value	1	TP	FN
	0	FP	TN

Table 2.1: Confusion Matrix

ROC curves begin from the bottom-left corner and rise to the top-right corner. Moving along the ROC curve represents trading off false positives for false negatives. Generally, random models will run up the diagonal, and the more the ROC curve bulges toward the top-left corner, the better the model separates one class from the other class. This property is clearly a virtue in binary classification problems.

Computing the area under the ROC curve (AUC) results in a single number that can efficiently measure the overall predictive accuracy, based on which one can select possibly the optimal model and discard suboptimal ones independently from the cost context and the class distribution. Therefore, the ROC curve is an efficient way to consider the overall accuracy represented by a AUC score and is mostly useful in binary class scenarios.

However, a serious criticism about ROC curves is comparing classifiers according to their ROC curves is not trivial because ROC curves tend to intersect [7][33][50]. In that common situation, one classifier might be better than another for some certain values of false positive rates, but might be worse for other false positive rates [7]. The intersection problem will mislead the resulting evaluation, especially when handling a set of ROC curves with inconsistencies from some random selective samples. Therefore, the ROC curve alone does a poor job of aiding the choice of classifier, because AUC as a global value fails to take account of when one classifier does not clearly dominate others over the entire performance [33][50].

Furthermore, in the real life applications, evaluating the performance at a specific false positive rate is often required rather than analyzing the ROC curve in the whole range. The cutoff of false positive rate is often trait dependent. For example, in clinical medicine applications, the decision of giving a treatment depends on the seriousness of side-effects of the treatment,

where the required test specificity is more stringent than that of screening or prevention proposes. Whether or not to give an operation on a cancer patient requires the diagnose of cancer to be 100% sure (*specificity*), whereas the physical examination of population individuals who are predicted to have high risk of certain cancer requires much less specificity. In our study, we also specifically investigate test accuracy at each false positive rate.

## 2.4 The introduction of fuzzy set theory

Fuzzy logic system was first introduced by L.A.Zadeh in the mid-1960s for representing approximated knowledge that cannot be respected by traditional crisp approaches [60][61][63]. In recent years, fuzzy logic systems have been successfully applied in a variety of fields and their applications have increased dramatically in popularity for various aspects. The four basic elements of each fuzzy logic system are *fuzzifier*, *rule base*, *inference engine*, and *defuzzifier*, as shown in Figure 2.1. The *fuzzifier* maps the inputs into fuzzy systems, which are subsequently used as inputs to the inference engine, whereas the *defuzzifier* maps the fuzzy sets produced by the inference engine into crisp numbers. The *fuzzy rule base* is a collection of rules from knowledge base encoding the general relation between the inputs and outputs, which is combined in the *inference engine*, to produce fuzzy outputs.

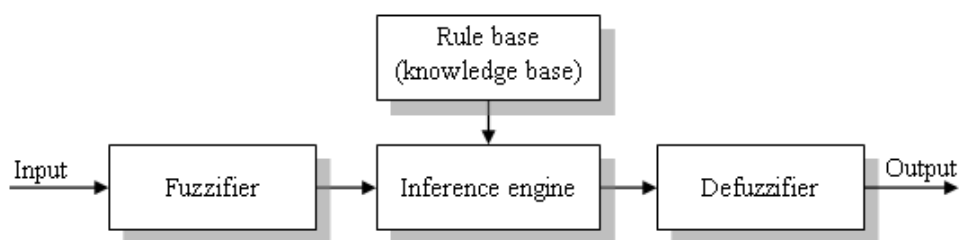


Figure 2.1: The building block of fuzzy logic system

In quite a few applications of fuzzy set theory in the context of decision making, we are confronted with the problem to select the best one from a collection of possible solutions [5][14][29][31][65]. It has been addressed in

connection with introducing *fuzziness* into some special types of classical decision problem for making a choice among the alternatives, and dealt with independently of any specific application. In our study, such connection is introduced for model selection problems.

Uncertainty characteristics are identified as *fuzziness* treated on the basis of the fuzzy set theory [43]. In contrast to classical set theory, the fuzzy set theory permits a gradual assessment of the membership of elements in relation to a set. This gradual membership is described by a membership function, leading to the definition of fuzzy numbers.

A fuzzy number is a convex, normalized fuzzy set  $\tilde{A} \subseteq \mathbb{R}$  represented through a membership function  $\mu_A(x)$ .

$$\tilde{A} = (x, \mu_A(x)) \mid x \in X \quad (2.3)$$

where each element in the range of the defined fuzzy set has an associated grade of membership function. The grade of membership function is a real number between 0 and 1, and represents the degree to which the particular element belongs to the fuzzy set. In this manner, it is possible for elements to belong to the set at some degree, thereby allowing a non-crisp membership [62][63].

## 2.5 The method for comparing fuzzy numbers

The key issue in operationalizing fuzzy set theory particularly in decision analysis therefore is how to defuzzify or compare fuzzy numbers. Since the beginning of the fuzzy set theory, the problem of ranking fuzzy subsets has been studied [41]. As the problem of comparing fuzzy numbers plays a crucial role in making decisions under a fuzzy environment, the techniques of comparing fuzzy numbers have received considerable attention in the fuzzy set theory literature [9][15][17][18][19][27][39][41][42][54][56].

The importance of the problem of comparing fuzzy subsets lies in the fact

that these subsets can be obtained in a decision-making problem to represent the performances of a set of alternatives. Therefore, a comparison between these subsets is a comparison between the alternatives [9]. G. Bortolan and R. Degani provided a review of a number of ranking methods which have been investigated in the extent literature in [9]. Most of these methods attempt to compare fuzzy sets through two ways. One approach simply consists of the definition of a ranking function mapping each fuzzy set into the real line, where a natural order exists. Another approach considers a different formalization of the problem to obtain a fuzzy set of optimal alternatives.

Yager's indices are some particular ranking methods [19][56][57][58] based upon the former idea by associating a fuzzy number with a crisp value and using this value to compare and rank the fuzzy numbers. It supposes that there are  $n$  normal convex fuzzy subsets  $\tilde{u}_i$ ,  $i \in 1, 2, \dots, n$ . Simple methods to order these subsets  $\tilde{u}_i$  depend on the definition of a ranking function  $F$  mapping each fuzzy interval into the real line, where a natural order exists [9][56][58].

$$F(\tilde{u}_i) < F(\tilde{u}_j) \Rightarrow \tilde{u}_i < \tilde{u}_j \quad (2.4)$$

$$F(\tilde{u}_i) = F(\tilde{u}_j) \Rightarrow \tilde{u}_i \approx \tilde{u}_j \quad (2.5)$$

$$F(\tilde{u}_i) > F(\tilde{u}_j) \Rightarrow \tilde{u}_i > \tilde{u}_j \quad (2.6)$$

### Yager's index of the center of gravity

Yager's index of the center of gravity, so-called centroid method, is the most common and physically attractive of all the defuzzification methods [13][16][59]. The defuzzified scalar value is nothing else than the center of gravity of the represented fuzzy set. It does not assume any hypothesis of normality or convexity of the fuzzy sets. Mathematically,  $\tilde{u}_i = \{z, \mu_{\tilde{u}_i}\}$ ,  $z \in S_i \subset I$  ( $S_i \rightarrow [0, 1]$  is called the domain or universe of discourse):

$$F(\tilde{u}_i) = \frac{\int_0^1 g(z)\mu_{\tilde{u}_i}(z)dz}{\int_0^1 \mu_{\tilde{u}_i}(z)dz} \quad (2.7)$$

The weight  $g(z)$  is a measure of the importance of the value  $z$ . If we assume linear weights, that is  $g(z)=z$ , then this function is equivalent to calculating the center of gravity of the fuzzy set.

### Yager's index of the mean area

In another Yager ranking function [56][57][58], if  $U_i^\alpha$  is the  $\alpha$ -level set of  $\tilde{u}_i$  and  $M(U_i^\alpha)$  is the mean value of the elements of  $U_i^\alpha$ , then:

$$F(\tilde{u}_i) = \int_0^{\alpha_{max}} M(U_i^\alpha)d\alpha, \text{ where: } \alpha_{max} = \text{hgt}(\tilde{u}_i) \quad (2.8)$$

where  $hgt$  refers to the height of a fuzzy set which is the maximum value of its membership function.



## Chapter 3

# Methodology to incorporate sampling uncertainty into model selection

### 3.1 The architecture of the model selection system

The aim of this study is to illustrate a new comprehensive model selection system by which the predictive performance of difference binary classifiers is evaluated to select the most trusted and robust predictive model under sampling uncertainty, based on the combination of ROC analysis and fuzziness.

The main idea and architecture of this model selection system are illustrated in Figure 3.1. Given any database, the bootstrap method is used to generate a small portion of the sample a number of times. The reason to generate a small portion is to simulate the situation where we analyze a sample from a population. In addition, when database is large, modeling can be computational complex and costly. For the strapped samples, multiple alternative modeling methods are performed to predict a binary outcome. The algorithms used in our system are “Logistic Regression” (LR), “Fuzzy Inference System” (FIS), and “Fuzzy C-means” (FCM) classification models. ROC curves are derived for each model, and the TPRs under each FPR are

estimated accordingly. Assuming the TPRs generated from different models belong to a particular member at some degree, the membership of each data point can be calculated based on the distribution of the data points. To address the question which method performs best, a strategy is therefore based on comparing these fuzzy numbers. In addition to the micro performance at each FPR, combining the fuzzy intervals of TPRs into a AUC fuzzy interval will provide an overall predictive accuracy. The proposed model selection system is fully automatic and computationally efficient using Matlab 2008 software, with the interactive environment which we enable to perform computationally intensive tasks.

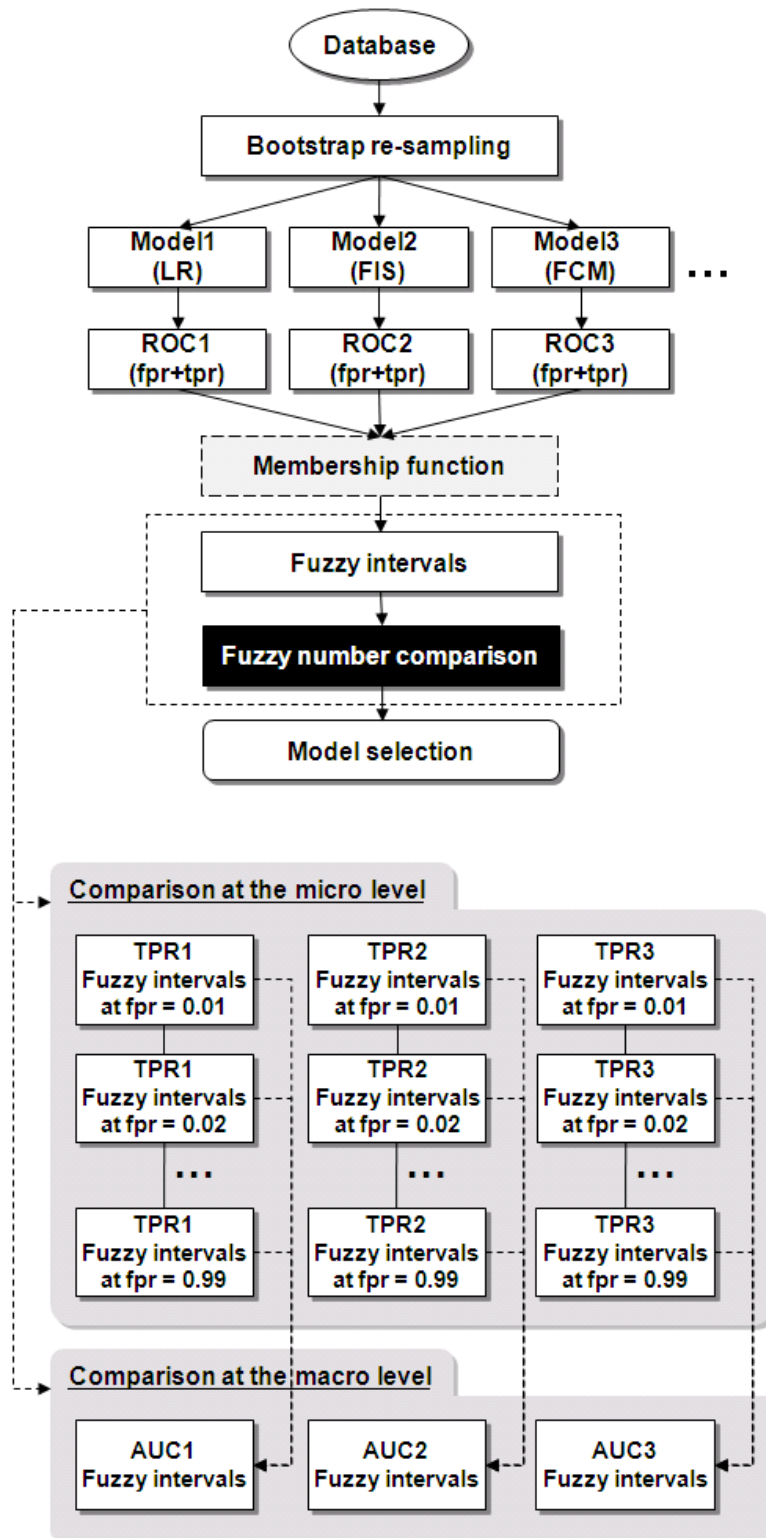


Figure 3.1: The architecture of the model selection system

## 3.2 The algorithm of the model selection system

The algorithm for each step of the model selection system is presented as below.

- The bootstrap algorithm is used to generate re-samples for modeling.
- “Logistic Regression” (LR), “Fuzzy Inference System” (FIS), and “Fuzzy C-means” (FCM) classifiers are chosen to train the re-samples and make predictions based on the training models.
- Five methods, so-called “the regular crisp interval function”, “the triangular mean function”, “the triangular median function”, “the trapezoidal function with mean and standard deviation”, and “the fitting function of envelope”, are introduced for mapping TPRs into fuzzy intervals.
- We adopt Yager’s index of the center of gravity and the mean area as the method of ranking fuzzy numbers.
- Additionally, a *resolution principle* based method is used to connect TPR fuzzy intervals with a AUC fuzzy interval.

These algorithms are illustrated in details in the following sections.

### 3.3 The method for capturing sampling uncertainty - Bootstrap

To take sampling uncertainty fully into account into model selection problems, we use the bootstrap approach.

The bootstrap technique was introduced by Efron and was fully described in Efron and Tibshirani (1993) [22]. The idea allows re-samples to be generated from the single original sample, instead of collecting many samples from the population. Given a data set of size  $n$ , a bootstrap sample is obtained by sampling  $n$  instances uniformly from the data set with replacement. To this end, the bootstrap technique firstly finds the sampling distribution, and accordingly creates as many re-samples as you want by repeatedly sampling with replacement from the original sample. Sampling with replacement refers to that we randomly draw an observation from the original database, and after that we put it back before drawing the next observation. As a result, any individual sample can be drawn more than once, or not be drawn at all. If the sampling does without replacement, we will get the same set of numbers as the original database, though in a different order.

The bootstrap approach is a recently developed technique for statistical inference with the increasing modern computer power and has been widely used as a key tool of simulated inference to improve upon asymptotic approximations. The bootstrap technique has been increasingly used to capture the information about sampling uncertainty [11][12][36]. In our study, we use the bootstrap distribution as a way to visualize how ROC curves perform differently and to estimate the variation in a statistic based on the given data set. By performing the multiple models on each set of the strapped samples, we attempt to capture the differences in predictive accuracy and sensitivity under each false positive rates represented by ROC curves. Therefore, a simple way to incorporate sampling uncertainty into inference is to apply the model selection procedure independently to each of these re-samples [11].

### 3.4 The method for mapping fuzzy intervals with determined membership functions

From the bootstrapped samples, we obtain the set of ROC curves and TPRs at each specified FPR based on different prediction models. Due to sampling uncertainty, there is a variation among the ROC curves. We analyze this variation by transferring the TPRs at each FPR into fuzzy intervals. Therefore, we provide an approach to select the model given different FPR.

For example, we draw 100 sets of bootstrap samples. Then 100 ROC curves are formed from “Logistic Regression”, “Fuzzy Inference System”, and “Fuzzy C-means” models respectively as shown in Figure 3.2. A tailor-made program enables us to present the corresponding TPR for each ROC curve at each cutoff of FPR from 0.01, 0.02, ..., to 0.99, as shown in Figure 3.3 (For example, the TPR points at FPR = 0.1 - 0.9). The x-axis is the value of TPR, whereas the y-axis is the number of TPR. Sometimes due to the discrete nature of the data, there is no exact value of FPR at these cutoffs. The solution is finding the FPR value which is the nearest value of 0.01, 0.02, ..., and 0.99, and obtaining the corresponding nearest TPR value at each FPR cutoff by using the “interpolation” function in Matlab. Interpolation function is the same operation as “table lookup”. For example, for the command line “`YI = INTERP1(X,Y,XI)`”, the “table” is  $[X,Y]$  and “INTERP1” “looks-up” the elements of  $XI$  in  $X$ , and returns values  $YI$  interpolated within the elements of  $Y$ .

Next, fuzzy intervals are introduced to map the variation of ROC curves, represented by the variation of the TPRs at any given FPR. Fuzzy intervals are represented through a membership function, where each element in the range of the defined fuzzy set has an associated grade of membership function. The grade of membership function is a real number between 0 and 1, and represents the degree to which the particular element belongs to the fuzzy set. In this manner, it is possible for elements to belong to the set at some degree, thereby allowing a non-crisp membership [62][63].

Primarily, fuzzification is a subjective assessment. In this context, the fuzzification approach for acquiring the shape of any particular membership

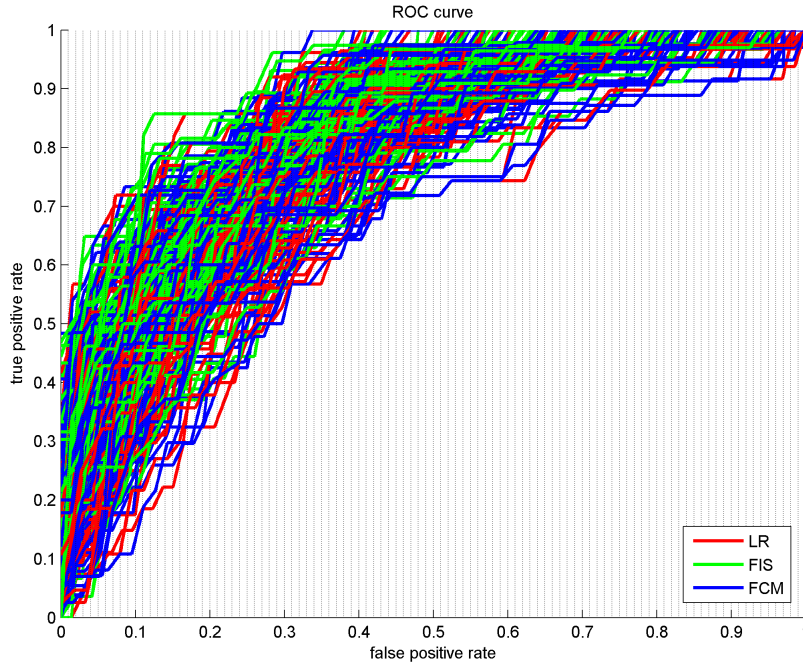


Figure 3.2: The sample ROC curves derived from three classifiers

function is often dependent on the application, i.e. the available information [35][43]. For most fuzzy logic control problems, it is assumed that the membership functions are linear, usually with a triangular or trapezoidal shape. For the other alternatives, a statistical approach is used to automatic generation the shapes. The goal of uncertainty quantification is assign an appropriate membership function to real-world information with respect to sampling uncertainty. In this context, We employ five statistical approaches to formulate the TPRs into fuzzy intervals, including “the regular crisp interval function”, “the triangular mean function”, “the triangular median function”, “the trapezoidal function with mean and standard deviation”, and “the fitting function of envelope”. These membership functions are data driven, basically determined by the distribution of TPRs. The information derived from the distribution increases for forming the fuzzy intervals from the first to the last function, illustrated as following with an example shown in Figure

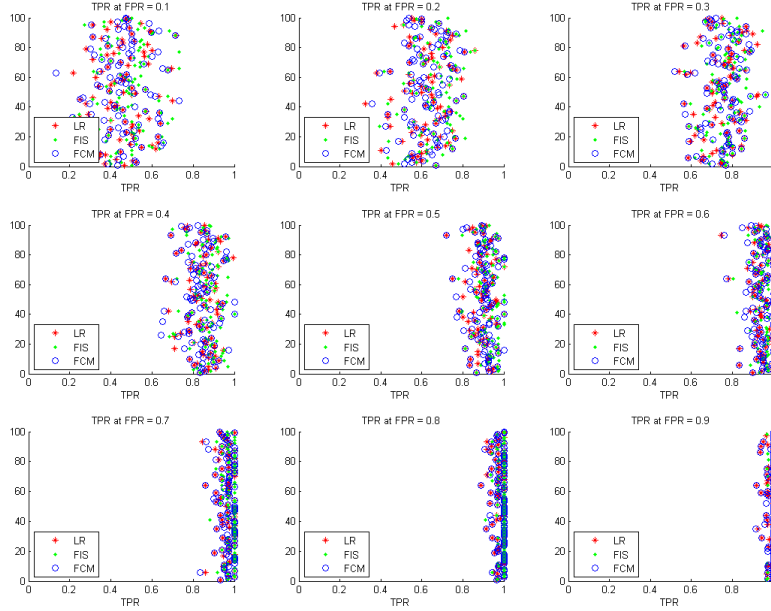


Figure 3.3: The corresponding TPR at each cutoff of false positive rate

### 3.4.

1. **The regular crisp interval function ranged from minimum and maximum:** The membership function is defined as  $M(TPR_{min} \leq TPR_i \leq TPR_{max}) = 1$ , and  $M(TPR_i < TPR_{min} \text{ or } TPR_i > TPR_{max}) = 0$ . As we assume the membership function is linear, the fuzzy intervals are plotted by connecting these four points with defined membership value of  $(TPR_{min}, 0)$ ,  $(TPR_{min}, 1)$ ,  $(TPR_{max}, 1)$ , and  $(TPR_{max}, 0)$ .

$$M_i = \begin{cases} 0 & \text{if } TPR_i < TPR_{min} \\ 1 & \text{if } TPR_{min} \leq TPR_i \leq TPR_{max} \\ 0 & \text{if } TPR_i > TPR_{max} \end{cases} \quad (3.1)$$

2. **The triangular mean function:** The membership function is defined as  $M(TPR_i = TPR_{mean}) = 1$ , and  $M(TPR_i < TPR_{min} \text{ or } TPR_i > TPR_{max}) = 0$ . As we assume the membership function is lin-



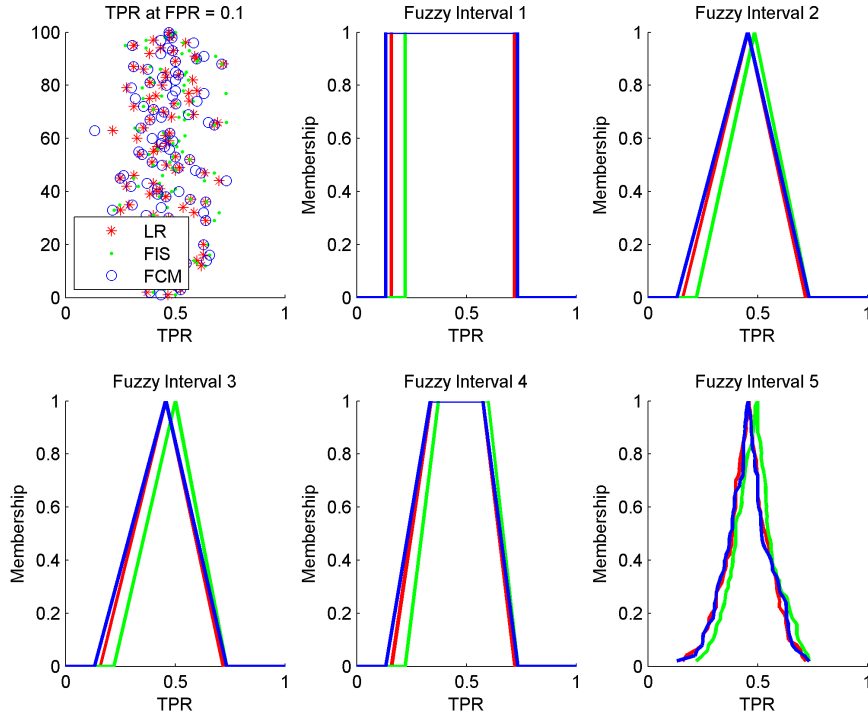


Figure 3.4: The distribution of TPR at  $FPR = 0.1$  and the corresponding fuzzy intervals based on different membership functions: (1) The regular crisp interval function ranged from minimum and maximum (2) The triangular mean function; (3) The triangular median function; (4) The trapezoidal function with mean and standard deviation calculated; and (5) The fitting function of envelope

ear with a triangular shape, the fuzzy intervals are plotted by connecting these three points with defined membership value of  $(TPR_{min}, 0)$ ,  $(TPR_{mean}, 1)$ , and  $(TPR_{max}, 0)$ .

$$M_i = \begin{cases} 0 & \text{if } TPR_i < TPR_{min} \\ \frac{TPR_i - TPR_{min}}{TPR_{mean} - TPR_{min}} & \text{if } TPR_{min} \leq TPR_i < TPR_{mean} \\ 1 & \text{if } TPR_i = TPR_{mean} \\ \frac{TPR_i - TPR_{max}}{TPR_{mean} - TPR_{max}} & \text{if } TPR_{mean} < TPR_i \leq TPR_{max} \\ 0 & \text{if } TPR_i > TPR_{max} \end{cases} \quad (3.2)$$

3. **The triangular median function:** The membership function is defined as  $M(TPR_i = TPR_{median}) = 1$ , and  $M(TPR_i < TPR_{min}$  or  $TPR_i > TPR_{max}) = 0$ . As we assume the membership function is linear with a triangular shape, the fuzzy intervals are plotted by connecting these three points with defined membership value of  $(TPR_{min}, 0)$ ,  $(TPR_{median}, 1)$ , and  $(TPR_{max}, 0)$ .

$$M_i = \begin{cases} 0 & \text{if } TPR_i < TPR_{min} \\ \frac{TPR_i - TPR_{min}}{TPR_{median} - TPR_{min}} & \text{if } TPR_{min} \leq TPR_i < TPR_{median} \\ 1 & \text{if } TPR_i = TPR_{median} \\ \frac{TPR_i - TPR_{max}}{TPR_{median} - TPR_{max}} & \text{if } TPR_{median} < TPR_i \leq TPR_{max} \\ 0 & \text{if } TPR_i > TPR_{max} \end{cases} \quad (3.3)$$

4. **The trapezoidal function with mean and standard deviation calculated:** The membership function is defined as  $M(TPR_{mean-std} \leq TPR_i \leq TPR_{mean+std}) = 1$ , and  $M(TPR_i < TPR_{min}$  or  $TPR_i > TPR_{max}) = 0$ . As we assume the membership function is linear with a trapezoidal shape, the fuzzy intervals are plotted by connecting these four points with defined membership value of  $(TPR_{min}, 0)$ ,  $(TPR_{mean-std}, 1)$ ,  $(TPR_{mean+std}, 1)$  and  $(TPR_{max}, 0)$ .

$$M_i = \begin{cases} 0 & \text{if } TPR_i < TPR_{min} \\ \frac{TPR_i - TPR_{min}}{TPR_{mean-std} - TPR_{min}} & \text{if } TPR_{min} \leq TPR_i < TPR_{mean-std} \\ 1 & \text{if } TPR_{mean-std} \leq TPR_i \leq TPR_{mean+std} \\ \frac{TPR_i - TPR_{max}}{TPR_{mean+std} - TPR_{max}} & \text{if } TPR_{mean+std} < TPR_i \leq TPR_{max} \\ 0 & \text{if } TPR_i > TPR_{max} \end{cases} \quad (3.4)$$

where  $std$  is the standard deviation depending on the distribution of TPRs.

5. **The fitting function of envelope:** The membership function is de-

defined as  $M(TPR_i = TPR_{median}) = 1$ , and  $M(TPR_i < TPR_{min}$  or  $TPR_i > TPR_{max}) = 0$ . TPRs are sorted ascending. The fuzzy intervals are plotted based on the value of TPR as x-axis against the ranks as y-axis.

$$M_i = \begin{cases} 0 & \text{if } TPR_i < TPR_{min} \\ \frac{2i}{N} & \text{if } TPR_{min} \leq TPR_i \leq TPR_{max} \text{ and } i < \frac{N}{2} \\ 1 & \text{if } TPR_i = TPR_{median} \text{ i.e. } i = \frac{N}{2} \\ \frac{2i}{N} - 1 & \text{if } TPR_{min} \leq TPR_i \leq TPR_{max} \text{ and } i > \frac{N}{2} \\ 0 & \text{if } TPR_i > TPR_{max} \end{cases} \quad (3.5)$$

where  $i$  is the rank of TPR and  $N$  is the total number of TPR. For an instance with 100 sets of ROC curves, we obtain 100 TPR points at each FPR. The rank from the minimum to the median is 1 – 50 corresponding an ascending value from 0 to 1, whereas the ranks from the median to the maximum is 51 – 100 corresponding a descending value from 1 to 0.

### 3.5 The method for comparing fuzzy numbers

We adopt two Yager's indices to compare fuzzy numbers in our system, because they are the most common used and physically attractive among defuzzification methods.

#### Yager's index of the center of gravity

In general, determining the center of gravity is a complicated procedure because the mass and weight may not be uniformly distributed throughout the object. In our case, the mass is uniformly distributed, so that the center of gravity is greatly simplified at the average location of the physical dimensions. For example of a triangle of height  $h$ , the center of gravity is at  $\frac{h}{3}$ ,

as shown in Figure 3.5(a). The center of gravity in terms of x-value for a triangle is obtained by calculating the average x-value of the three vertices given the same height.

For the envelope-shaped fuzzy subsets, we employ the  $\alpha$ -weight mean of maxima method [19] to obtain the valuation by calculating the average of the elements at all  $\alpha$ -cuts <sup>1</sup>.

For the trapezoidal fuzzy intervals, we refer to the center of maxima method [19], a simplified version of the  $\alpha$ -weight mean of maxima method. Instead of taking the elements at all  $\alpha$ -cuts, we consider the smallest and the largest only and we take the middle as valuation.

In the case of the convex fuzzy sets with the same height, these three methods are equivalent [59][63], and useful in order to reduce the complexity of the calculation. In other words, a natural choice to obtain a single real number from the mean value of a fuzzy number could be to calculate the arithmetic mean of its bounds.

### Yager's index of the mean area

Graphically  $F(\tilde{u}_i)$  can be represented by the area between the mean of up-line and bottom-line for trapezoidal shaped fuzzy intervals, shown as the shaded area in Figure 3.5(b). The dotted line represents the average value of the elements with at least that grade of membership. This simplified method is easy to calculate for these differently shaped fuzzy sets and therefore it is also the most widely used in ranking fuzzy subsets techniques [30].

---

<sup>1</sup>**Definition 1:** The  $\alpha$ -cut of a fuzzy set is the crisp set formed by those elements whose membership function grade is equal to or greater than a specified threshold value  $\alpha$ . Mathematically,

$$A_\alpha = \begin{cases} x \in X \mid \mu_A(x) \geq \alpha & \alpha > 0 \\ cl(sup(A)) & \alpha = 0 \end{cases} \quad (3.6)$$

**Definition 2:** The *support* of a fuzzy set A in X domain is the crisp subset of X whose elements have non-zero membership in A. Mathematically,

$$sup(A) = x \in X \mid \mu_A(x) > 0 \quad (3.7)$$

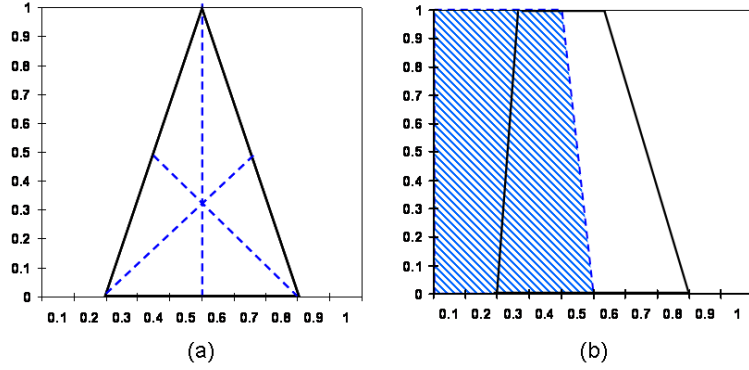


Figure 3.5: Yager index: (a) The center of gravity (b) The mean area

### 3.6 The method for forming AUC fuzzy numbers

The comparison of the TPR fuzzy sets at each FPR evaluates the predictive performance at a micro level, whereas the AUC value is a global measure at a macro level. Our aim in this section is to establish a connection between the TPR fuzzy sets and a AUC fuzzy set.

In fuzzy logic, the *resolution principle* assumes that every fuzzy set A can be uniquely represented as a collection of  $\alpha$ -cut sets according to:

$$\mu_A(x) = \sup_{\alpha \in [0,1]} [\alpha \cdot \mu_{A_\alpha}(x)] \quad (3.8)$$

The fuzzy *resolution principle* implies that the fuzzy set theory is a generalization of the classical set theory, and that its results can also be represented in terms of the classical set theory. This implication makes an possible application of the resolution principle. With this implication, a global AUC fuzzy set is associated to the combination of the TPR fuzzy sets of each  $\alpha$ -level at each FPR, by repeatedly applying the *resolution principle* as inference rules [20][37].

The procedure to represent the global AUC fuzzy set is realized in four steps:

1. Fuzzy sets can be described effectively using  $\alpha$ -cut sets [5]. Firstly, we obtain the  $\alpha$ -cut sets of TPR fuzzy sets from  $\alpha=0$  (*support set*) to  $\alpha=1$  (*core set*).
2. Each crisp TPR set with two values from  $FPR = 0.01$  to  $FPR = 0.99$  provides the border lines for forming ROC curves at a particular  $\alpha$  value. The best and worst ROC curve of each  $\alpha$  value from 0 to 1 can be completed from the bottom-left corner to the top-right corner. Accordingly we obtain eleven sets of the best and worst ROC curves.
3. The corresponding best and worst AUC values are calculated for all eleven sets of ROC curves.
4. Therefore, the best and worst AUC value at each  $\alpha$  value are associated to a crisp *alpha*-cut set. Based on the *resolution principle*, a AUC fuzzy interval is generated by the combination of these crisp  $\alpha$ -cut sets.

The resulting shape of AUC fuzzy interval is in line with the TPR fuzzy intervals from which the AUC fuzzy interval is derived. For example, the triangle-shaped TPR intervals result in an exactly triangle-shaped AUC interval. This is mainly due to the AUC calculating algorithm **TRAPZ**(X,Y) we used, which computes the integral of Y with respect to X using the *trapezoidal* method. The finding is useful to simplify the procedure of generating the AUC fuzzy interval <sup>2</sup> by taking TPR values at  $\alpha$ -cut = 0 and 1 only into consideration, instead of cutting them into 11 from  $\alpha$ -cut = 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, to 1, or even more precisely. This is because the precision of  $\alpha$ -cutoff makes no difference for the resulting shape. We provide mathematical proofs as follows.

**Theorem:**

The AUC fuzzy interval derived from triangular TPR fuzzy intervals is exactly triangle-shape.

---

<sup>2</sup>Only for the linear-shaped fuzzy intervals: (1) the regular crisp interval, (2) the triangular mean interval, (3) the triangular median interval, and (4) the trapezoidal interval; whereas (5) the fitting envelope interval is an exception.

**Proof:**

We will use mathematical induction to prove that the distance of any two successive points of the left line and the right line is constant. In other words, the TPR points in terms of each two successive  $\alpha$ -cuts is an arithmetic sequence.

**Antecedents:**

As illustrated in Figure 3.6, we make some assumptions for mathematical induction as presented in the following tables 3.1 & 3.2.

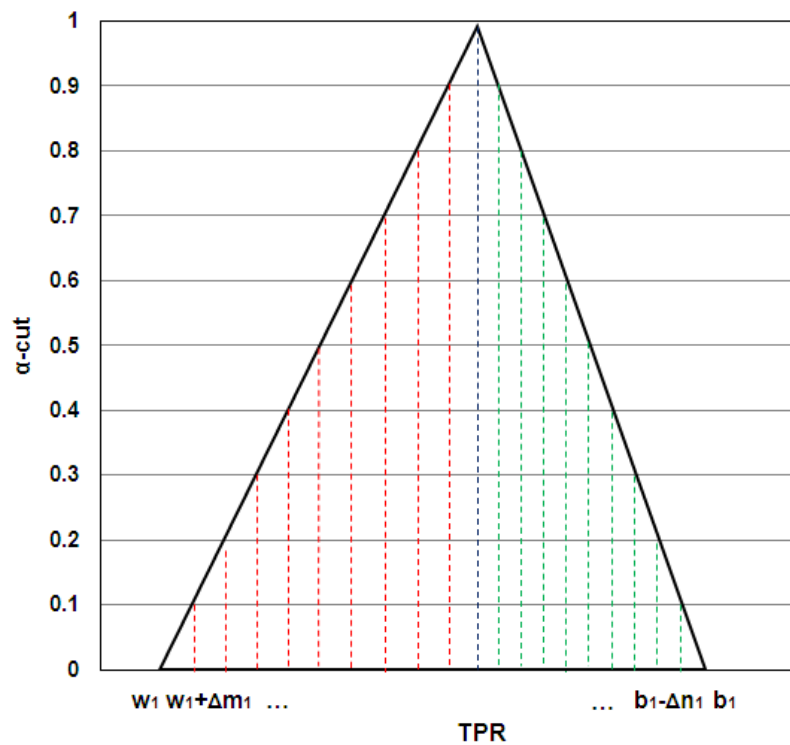


Figure 3.6: The example TPR fuzzy interval at FPR = 0.1

Table 3.1: The assuming information from triangular TPR fuzzy intervals - the left line

	$FPR=0$	$FPR=0.1$	$FPR=0.2$	$FPR=0.3$	$FPR=0.4$	$FPR=0.5$	$FPR=0.6$	$FPR=0.7$	$FPR=0.8$	$FPR=0.9$	$FPR=1$
$\alpha = 0$	0	$w_1$	$w_2$	$w_3$	$w_4$	$w_5$	$w_6$	$w_7$	$w_8$	$w_9$	1
$\alpha = 0.1$	0	$w_1 + \Delta m_1$	$w_2 + \Delta m_2$	$w_3 + \Delta m_3$	$w_4 + \Delta m_4$	$w_5 + \Delta m_5$	$w_6 + \Delta m_6$	$w_7 + \Delta m_7$	$w_8 + \Delta m_8$	$w_9 + \Delta m_9$	1
$\alpha = 0.2$	0	$w_1 + 2\Delta m_1$	$w_2 + 2\Delta m_2$	$w_3 + 2\Delta m_3$	$w_4 + 2\Delta m_4$	$w_5 + 2\Delta m_5$	$w_6 + 2\Delta m_6$	$w_7 + 2\Delta m_7$	$w_8 + 2\Delta m_8$	$w_9 + 2\Delta m_9$	1
$\alpha = 0.3$	0	$w_1 + 3\Delta m_1$	$w_2 + 3\Delta m_2$	$w_3 + 3\Delta m_3$	$w_4 + 3\Delta m_4$	$w_5 + 3\Delta m_5$	$w_6 + 3\Delta m_6$	$w_7 + 3\Delta m_7$	$w_8 + 3\Delta m_8$	$w_9 + 3\Delta m_9$	1
$\alpha = 0.4$	0	$w_1 + 4\Delta m_1$	$w_2 + 4\Delta m_2$	$w_3 + 4\Delta m_3$	$w_4 + 4\Delta m_4$	$w_5 + 4\Delta m_5$	$w_6 + 4\Delta m_6$	$w_7 + 4\Delta m_7$	$w_8 + 4\Delta m_8$	$w_9 + 4\Delta m_9$	1
$\alpha = 0.5$	0	$w_1 + 5\Delta m_1$	$w_2 + 5\Delta m_2$	$w_3 + 5\Delta m_3$	$w_4 + 5\Delta m_4$	$w_5 + 5\Delta m_5$	$w_6 + 5\Delta m_6$	$w_7 + 5\Delta m_7$	$w_8 + 5\Delta m_8$	$w_9 + 5\Delta m_9$	1
$\alpha = 0.6$	0	$w_1 + 6\Delta m_1$	$w_2 + 6\Delta m_2$	$w_3 + 6\Delta m_3$	$w_4 + 6\Delta m_4$	$w_5 + 6\Delta m_5$	$w_6 + 6\Delta m_6$	$w_7 + 6\Delta m_7$	$w_8 + 6\Delta m_8$	$w_9 + 6\Delta m_9$	1
$\alpha = 0.7$	0	$w_1 + 7\Delta m_1$	$w_2 + 7\Delta m_2$	$w_3 + 7\Delta m_3$	$w_4 + 7\Delta m_4$	$w_5 + 7\Delta m_5$	$w_6 + 7\Delta m_6$	$w_7 + 7\Delta m_7$	$w_8 + 7\Delta m_8$	$w_9 + 7\Delta m_9$	1
$\alpha = 0.8$	0	$w_1 + 8\Delta m_1$	$w_2 + 8\Delta m_2$	$w_3 + 8\Delta m_3$	$w_4 + 8\Delta m_4$	$w_5 + 8\Delta m_5$	$w_6 + 8\Delta m_6$	$w_7 + 8\Delta m_7$	$w_8 + 8\Delta m_8$	$w_9 + 8\Delta m_9$	1
$\alpha = 0.9$	0	$w_1 + 9\Delta m_1$	$w_2 + 9\Delta m_2$	$w_3 + 9\Delta m_3$	$w_4 + 9\Delta m_4$	$w_5 + 9\Delta m_5$	$w_6 + 9\Delta m_6$	$w_7 + 9\Delta m_7$	$w_8 + 9\Delta m_8$	$w_9 + 9\Delta m_9$	1
$\alpha = 1$	0	$w_1 + 10\Delta m_1$	$w_2 + 10\Delta m_2$	$w_3 + 10\Delta m_3$	$w_4 + 10\Delta m_4$	$w_5 + 10\Delta m_5$	$w_6 + 10\Delta m_6$	$w_7 + 10\Delta m_7$	$w_8 + 10\Delta m_8$	$w_9 + 10\Delta m_9$	1

Table 3.2: The assuming information from triangular TPR fuzzy intervals - the right line

	$FPR=0$	$FPR=0.1$	$FPR=0.2$	$FPR=0.3$	$FPR=0.4$	$FPR=0.5$	$FPR=0.6$	$FPR=0.7$	$FPR=0.8$	$FPR=0.9$	$FPR=1$
$\alpha = 0$	0	$b_1$	$b_2$	$b_3$	$b_4$	$b_5$	$b_6$	$b_7$	$b_8$	$b_9$	1
$\alpha = 0.1$	0	$b_1 - \Delta n_1$	$b_2 - \Delta n_2$	$b_3 - \Delta n_3$	$b_4 - \Delta n_4$	$b_5 - \Delta n_5$	$b_6 - \Delta n_6$	$b_7 - \Delta n_7$	$b_8 - \Delta n_8$	$b_9 - \Delta n_9$	1
$\alpha = 0.2$	0	$b_1 - 2\Delta n_1$	$b_2 - 2\Delta n_2$	$b_3 - 2\Delta n_3$	$b_4 - 2\Delta n_4$	$b_5 - 2\Delta n_5$	$b_6 - 2\Delta n_6$	$b_7 - 2\Delta n_7$	$b_8 - 2\Delta n_8$	$b_9 - 2\Delta n_9$	1
$\alpha = 0.3$	0	$b_1 - 3\Delta n_1$	$b_2 - 3\Delta n_2$	$b_3 - 3\Delta n_3$	$b_4 - 3\Delta n_4$	$b_5 - 3\Delta n_5$	$b_6 - 3\Delta n_6$	$b_7 - 3\Delta n_7$	$b_8 - 3\Delta n_8$	$b_9 - 3\Delta n_9$	1
$\alpha = 0.4$	0	$b_1 - 4\Delta n_1$	$b_2 - 4\Delta n_2$	$b_3 - 4\Delta n_3$	$b_4 - 4\Delta n_4$	$b_5 - 4\Delta n_5$	$b_6 - 4\Delta n_6$	$b_7 - 4\Delta n_7$	$b_8 - 4\Delta n_8$	$b_9 - 4\Delta n_9$	1
$\alpha = 0.5$	0	$b_1 - 5\Delta n_1$	$b_2 - 5\Delta n_2$	$b_3 - 5\Delta n_3$	$b_4 - 5\Delta n_4$	$b_5 - 5\Delta n_5$	$b_6 - 5\Delta n_6$	$b_7 - 5\Delta n_7$	$b_8 - 5\Delta n_8$	$b_9 - 5\Delta n_9$	1
$\alpha = 0.6$	0	$b_1 - 6\Delta n_1$	$b_2 - 6\Delta n_2$	$b_3 - 6\Delta n_3$	$b_4 - 6\Delta n_4$	$b_5 - 6\Delta n_5$	$b_6 - 6\Delta n_6$	$b_7 - 6\Delta n_7$	$b_8 - 6\Delta n_8$	$b_9 - 6\Delta n_9$	1
$\alpha = 0.7$	0	$b_1 - 7\Delta n_1$	$b_2 - 7\Delta n_2$	$b_3 - 7\Delta n_3$	$b_4 - 7\Delta n_4$	$b_5 - 7\Delta n_5$	$b_6 - 7\Delta n_6$	$b_7 - 7\Delta n_7$	$b_8 - 7\Delta n_8$	$b_9 - 7\Delta n_9$	1
$\alpha = 0.8$	0	$b_1 - 8\Delta n_1$	$b_2 - 8\Delta n_2$	$b_3 - 8\Delta n_3$	$b_4 - 8\Delta n_4$	$b_5 - 8\Delta n_5$	$b_6 - 8\Delta n_6$	$b_7 - 8\Delta n_7$	$b_8 - 8\Delta n_8$	$b_9 - 8\Delta n_9$	1
$\alpha = 0.9$	0	$b_1 - 9\Delta n_1$	$b_2 - 9\Delta n_2$	$b_3 - 9\Delta n_3$	$b_4 - 9\Delta n_4$	$b_5 - 9\Delta n_5$	$b_6 - 9\Delta n_6$	$b_7 - 9\Delta n_7$	$b_8 - 9\Delta n_8$	$b_9 - 9\Delta n_9$	1
$\alpha = 1$	0	$b_1 - 10\Delta n_1$	$b_2 - 10\Delta n_2$	$b_3 - 10\Delta n_3$	$b_4 - 10\Delta n_4$	$b_5 - 10\Delta n_5$	$b_6 - 10\Delta n_6$	$b_7 - 10\Delta n_7$	$b_8 - 10\Delta n_8$	$b_9 - 10\Delta n_9$	1



**Induction:**

AUC calculation algorithm in our experiment = **TRAPZ**(X,Y) computes the integral of Y with respect to X using the *trapezoidal* method.

**Accordingly,**

*The worst AUC value (on the left line) at  $\alpha = 0$ :*

$$W_0 = \frac{0.1 \times w_1}{2} + \frac{0.1 \times (w_1 + w_2)}{2} + \frac{0.1 \times (w_2 + w_3)}{2} + \dots + \frac{0.1 \times (w_8 + w_9)}{2} + \frac{0.1 \times (w_9 + 1)}{2}$$

$$W_0 = 0.1w_1 + 0.1w_2 + 0.1w_3 + 0.1w_4 + 0.1w_5 + 0.1w_6 + 0.1w_7 + 0.1w_8 + 0.1w_9 + 0.05$$

*The worst AUC value (on the left line) at  $\alpha = 0.1$ :*

$$W_1 = \frac{0.1 \times (w_1 + 1\Delta m_1)}{2} + \frac{0.1 \times (w_1 + 1\Delta m_1 + w_2 + 1\Delta m_2)}{2} + \frac{0.1 \times (w_2 + 1\Delta m_2 + w_3 + 1\Delta m_3)}{2} + \dots + \frac{0.1 \times (w_8 + 1\Delta m_8 + w_9 + 1\Delta m_9)}{2} + \frac{0.1 \times (w_9 + 1\Delta m_9 + 1)}{2}$$

$$W_1 = 0.1w_1 + 0.1\Delta m_1 + 0.1w_2 + 0.1\Delta m_2 + 0.1w_3 + 0.1\Delta m_3 + 0.1w_4 + 0.1\Delta m_4 + 0.1w_5 + 0.1\Delta m_5 + 0.1w_6 + 0.1\Delta m_6 + 0.1w_7 + 0.1\Delta m_7 + 0.1w_8 + 0.1\Delta m_8 + 0.1w_9 + 0.1\Delta m_9 + 0.05$$

...

*The worst AUC value (on the left line) at  $\alpha = 1$ :*

$$W_{10} = \frac{0.1 \times (w_1 + 10\Delta m_1)}{2} + \frac{0.1 \times (w_1 + 10\Delta m_1 + w_2 + 10\Delta m_2)}{2} + \frac{0.1 \times (w_2 + 10\Delta m_2 + w_3 + 10\Delta m_3)}{2} + \dots + \frac{0.1 \times (w_8 + 10\Delta m_8 + w_9 + 10\Delta m_9)}{2} + \frac{0.1 \times (w_9 + 10\Delta m_9 + 1)}{2}$$

$$W_{10} = 0.1w_1 + 1\Delta m_1 + 0.1w_2 + 1\Delta m_2 + 0.1w_3 + 1\Delta m_3 + 0.1w_4 + 1\Delta m_4 + 0.1w_5 + 1\Delta m_5 + 0.1w_6 + 1\Delta m_6 + 0.1w_7 + 1\Delta m_7 + 0.1w_8 + 1\Delta m_8 + 0.1w_9 + 1\Delta m_9 + 0.05$$

**Therefore,**  $W_0 - W_1 - W_2 - W_3 - W_4 - W_5 - W_6 - W_7 - W_8 - W_9 - W_{10}$  is an arithmetic sequence, i.e. the distance of any two successive points of the left line is  $0.1\Delta m_1 + 0.1\Delta m_2 + 0.1\Delta m_3 + 0.1\Delta m_4 + 0.1\Delta m_5 + 0.1\Delta m_6 + 0.1\Delta m_7 + 0.1\Delta m_8 + 0.1\Delta m_9$ , constantly.

Similarly,

*The best AUC value (on the right line) at  $\alpha = 0$ :*

$$B_0 = \frac{0.1 \times b_1}{2} + \frac{0.1 \times (b_1 + b_2)}{2} + \frac{0.1 \times (b_2 + b_3)}{2} + \dots + \frac{0.1 \times (b_8 + b_9)}{2} + \frac{0.1 \times (b_9 + 1)}{2}$$

$$B_0 = 0.1b_1 + 0.1b_2 + 0.1b_3 + 0.1b_4 + 0.1b_5 + 0.1b_6 + 0.1b_7 + 0.1b_8 + 0.1b_9 + 0.05$$

*The best AUC value (on the right line) at  $\alpha = 0.1$ :*

$$B_1 = \frac{0.1 \times (b_1 - 1\Delta n_1)}{2} + \frac{0.1 \times (b_1 - 1\Delta n_1 + b_2 - 1\Delta n_2)}{2} + \frac{0.1 \times (b_2 - 1\Delta n_2 + b_3 - 1\Delta n_3)}{2} + \dots + \frac{0.1 \times (b_8 - 1\Delta n_8 + b_9 - 1\Delta n_9)}{2} + \frac{0.1 \times (b_9 - 1\Delta n_9 + 1)}{2}$$

$$B_1 = 0.1b_1 - 0.1\Delta n_1 + 0.1b_2 - 0.1\Delta n_2 + 0.1b_3 - 0.1\Delta n_3 + 0.1b_4 - 0.1\Delta n_4 + 0.1b_5 - 0.1\Delta n_5 + 0.1b_6 - 0.1\Delta n_6 + 0.1b_7 - 0.1\Delta n_7 + 0.1b_8 - 0.1\Delta n_8 + 0.1b_9 - 0.1\Delta n_9 + 0.05$$

...

*The best AUC value (on the right line) at  $\alpha = 1$ :*

$$B_{10} = \frac{0.1 \times (b_1 - 10\Delta n_1)}{2} + \frac{0.1 \times (b_1 - 10\Delta n_1 + b_2 - 10\Delta n_2)}{2} + \frac{0.1 \times (b_2 - 10\Delta n_2 + b_3 - 10\Delta n_3)}{2} + \dots + \frac{0.1 \times (b_8 - 10\Delta n_8 + b_9 - 10\Delta n_9)}{2} + \frac{0.1 \times (b_9 - 10\Delta n_9 + 1)}{2}$$

$$B_{10} = 0.1b_1 - 1\Delta n_1 + 0.1b_2 - 1\Delta n_2 + 0.1b_3 - 1\Delta n_3 + 0.1b_4 - 1\Delta n_4 + 0.1b_5 - 1\Delta n_5 + 0.1b_6 - 1\Delta n_6 + 0.1b_7 - 1\Delta n_7 + 0.1b_8 - 1\Delta n_8 + 0.1b_9 - 1\Delta n_9 + 0.05$$

**Therefore,**  $B_0 - B_1 - B_2 - B_3 - B_4 - B_5 - B_6 - B_7 - B_8 - B_9 - B_{10}$  is an arithmetic sequence, i.e. the distance of any two successive points of the left line is  $-0.1\Delta n_1 - 0.1\Delta n_2 - 0.1\Delta n_3 - 0.1\Delta n_4 - 0.1\Delta n_5 - 0.1\Delta n_6 - 0.1\Delta n_7 - 0.1\Delta n_8 - 0.1\Delta n_9$ , constantly.

**In conclusion,** The interval formed by these two arithmetic sequences is triangular-shaped exactly.

# Chapter 4

## Experimental design

### 4.1 Real-life data sets

1. **Dutch Charity Database:**

#### Data relevance

Nowadays, a large amount of data is collected for marketing-related purposes. The data usually are used to establish, maintain and strengthen a direct relationship between companies and their customers in order to target the customers individually for specific product or service offers. The type of marketing for this end is so-called “direct marketing” [52].

Apart from a growing number of commercial companies such as bank and insurance companies which are adopting direct marketing as their business strategy, charity organization also employ direct marketing for fund raising. [52]

Unlike commercial companies, charity organizations do not have regular customers. They must be able to trace people who are likely to donate in order to optimize their fund raising. The people who donate money in response to a particular mailing campaign have high donating potential in the future, regarded as *high-promise supporters*. As a result, such *high-promise supporters* considered as the selected targets

will be contacted by mail.

### Data description

The data were collected by one of the largest charities in the Netherlands. The data set has been investigated by using such machine learning and data mining techniques as “frequent patterns” [46], “neural networks” [47], “fuzzy models” [51][53].

The raw database available to use contains information regarding over 725,000 supporters, described in detail by Potharst, Kaymark & Pijls [52]. The process of data reduction has been done by previous work to find a set of variables that are most explanatory as antecedents to a particular model (namely “feature selection”). The three so-called *recency*, *frequency* and *monetary* value features [52][53] as explanatory variables constructed from seven variables through feature selection are used for characterizing the donation history of the supporters in a binary system, i.e. yes = 1 whereas no = 0.

- (a) *Recency*: Number of weeks since last response (TIMELR);
- (b) *Frequency*: Number of months as a supporter (TIMECL);
- (c) *Monetary*: Fraction of mailings responded (FRQRES).

The abstract database consists of 7137 individual information used to trace people who are more likely to donate money in order to optimize the fund raising results. The targeted supporters selected who have a donation history value  $y = 1$  are then contacted by mail with priority in relation to other individuals within the database.

### Data preparation

First at all, the information with missing values is filtered out. The non-parametric bootstrap is implemented in the data set to generate 100 sets of re-samples for modeling purposes - both training and validating.

By default, each set of re-samples is the same size as the original sample. In our experiment, re-samples are drawn with replacement from original data set until the sample size is a small portion (10%) of the original large data set. The reasons to restrict the size of data sets to a small portion around 500 samples from the original large data set are (1) to save computational time, (2) to balance a tradeoff between cost & precision (The maximum margin of error of analyzing a sample size  $n = 500$  at 95% confidence interval =  $+/- 4.38\%$ ), and (3) to visualize a relevantly clear variation for analysis.

## 2. Haberman’s Survival Database:

Besides the Dutch charity database, we test our model selection system with some other databases from “The UC Irvine Machine Learning Repository” [4], such as “Haberman’s Survival Data Set”<sup>1</sup>. The data set contains 306 cases from a study that was conducted between 1958 and 1970 at the University of Chicago’s Billings Hospital on the survival of patients who had undergone surgery for breast cancer. There are 3 numerical explanatory variables: the age of patient at time of operation, the patient’s year of operation, and the number of positive axillary nodes detected. A binary-class attribute of survival status will be modeled and predicted: 1 = the patient survived 5 years or longer, whereas 2 = the patient died within 5 year.

Similar with Dutch charity database, we implement nonparametric bootstrap to obtain 100 sets of re-samples for use. Due to its small sample size, the full re-samples from the original data set are modeled for use.

## 3. Other Databases:

Other databases selected for our experiments from “The UC Irvine Machine Learning Repository” are ‘Blood Transfusion Service Center Data

---

<sup>1</sup><http://mlr.cs.umass.edu/ml/datasets/Haberman's+Survival>

Set’<sup>2</sup>, “Iris Data Set”<sup>3</sup>, “Hayes-Roth Data Set”<sup>4</sup>, “Mammographic Mass Data Set”<sup>5</sup>, and “Teaching Assistant Evaluation Data Set”<sup>6</sup>. The brief information of these data sets is described in Table 4.1.

---

<sup>2</sup><http://mlr.cs.umass.edu/ml/datasets/Blood+Transfusion+Service+Center>

<sup>3</sup><http://mlr.cs.umass.edu/ml/datasets/Iris>

<sup>4</sup><http://mlr.cs.umass.edu/ml/datasets/Hayes-Roth>

<sup>5</sup><http://mlr.cs.umass.edu/ml/datasets/Mammographic+Mass>

<sup>6</sup><http://mlr.cs.umass.edu/ml/datasets/Teaching+Assistant+Evaluation>

<b>Dataset</b>	<b>Information</b>	<b>Year</b>	<b>Sample size</b>	<b>Explanatory variables</b>	<b>Dependent variables</b>
<i>Blood transfusion service center</i>	The donor database of Blood Transfusion Service Center in Taiwan (China)	2008	748	3	A binary variable whether the individual donated blood
<i>Mammographic mass</i>	The database of breast cancer screening information collected at the Institute of Radiology of the University Erlangen-Nuremberg	2003-2006	961	5	A binary variable of the severity (benign or malignant)
<i>Teaching assistant evaluation</i>	The database of evaluations of teaching performance of teaching assistant assignments at the University of Wisconsin-Madison	1997	151	5	3 classes of Class attribute; we predict “low” against “medium” and “high”
<i>Hayes-Roth</i>	The database of classification and recognition confidence ratings in a concept learning study that used rule-described categories	1977	160	5	3 classes; we predict Class 1 against Class 2 and Class 3
<i>Iris</i>	The widely used database about the class of iris plant	1936	150	4	3 classes; we predict Iris Setosa against Iris Versicolour and Iris Virginica

Table 4.1: The brief description of the data sets used for experiment

## 4.2 Selected models

### 4.2.1 Logistic Regression (LR)

Traditionally, Logistic Regression (LR) is used extensively in a variety of disciplines, such as medical and social sciences as well as marketing applications. In statistics, LR is a generalized linear model used for binary or binomial regression, i.e. predicting the probability of occurrence of an event by fitting data to a logistic curve. The form of LR follows a linear regression whereas the binary response variable follows a logit link. The goal of modeling is to correctly predict the category of outcome. Binary LR is used in our case because it fits when the dependent variable is dichotomous (i.e. the dependent variable can take the value 1 with a probability of success  $\rho$ , or the value 0 with probability of failure  $1 - \rho$ ), and the independents are of any type.

A mathematical form of the LR equation is:

$$\text{logit}[\rho(x)] = \log\left[\frac{\rho(x)}{1 - \rho(x)}\right] = \alpha + \beta_1x_1 + \beta_2x_2 + \dots + \beta_ix_i, \quad (4.1)$$

where  $\alpha$  = the constant of the equation, and  $\beta$  = the coefficient of the predictor variables.

Sometimes when some predictors are highly correlated, the covariance matrix does not have an inverse, and thus regression based methods may fail. This is also true for logistic regression. Then we add a feature in the model selection tool that can automatically exclude the predictors in very high correlation (e.g.  $r > 0.9$ ). This is achieved by examination of the correlation matrix between predictors.

Besides LR, a number of different models will be evaluated through the proposed theoretical framework as described below. These learning algorithms have an advantage that they can be adapted to the nonlinearity in the data to capture the complex relations. This is an important motivation for applying them for performing classification and prediction. These models give an interesting and differently illuminating way of investigating the data



in terms of classification problems. They do add a flexible parametric tool to the data analyst's arsenal, and they should not be used to the exclusion of each other methods, instead of depending on the predictive performance based on different characterized databases.

## 4.2.2 Fuzzy Inference System (FIS)

### Model description

A fuzzy knowledge base can represent a complex system as a simple set of input-output rules. Since Japanese researcher Sugeno proposed the famous Sugeno model (or Takagi-Sugeno-Kang model or TSK model or T-S model) of fuzzy inference in the 1980s [37], it has been applied successfully in different fields. The main feature of the Sugeno fuzzy model is to express the local dynamics of each fuzzy implication rule by a linear (nonlinear) polynomial. In this case, fuzzy rule consequents are assumed to be a linear combination of the input variables, and the output is a convex combination of consequents with the coefficients regarding to the input membership function in the antecedents. This architecture of the model provides the feasibility of stability analysis, as well as reduces the computational efforts of fuzzy logic. This is the main reason why the Sugeno-type models are preferred and commonly used in applications of conventional crisp control and in simplified fuzzy models [32][63].

The Sugeno model maps the antecedent (input) part with a crisp function as the consequent (output) part, of the form: IF  $x$  is  $A_i$  AND  $y$  is  $B_i$ , THEN  $z_i = f_i(x, y)$ . As illustrated in Figure 4.1, the Sugeno FIS architecture consists four layers: fuzzy layer, product layer, de-fuzzy layer and aggregation layer. In *fuzzy layer*, the crisp inputs are converted to the fuzzy numbers by the input membership functions. In *product layer*, the weighting factor or firing strength of each rule is determined by evaluating the membership expressions in the antecedent of the rule. This is accomplished by applying the “and” operator to the membership values converted from crisp input value to fuzzy membership value. In *de-fuzzy layer*, the output  $z_i$  of each

rule is weighted by the firing strength  $\omega_i$  of the rule.

$$\omega_i z_i = \omega(a_i x + b_i y + c_i), i = 1 \dots \text{number of rules} \quad (4.2)$$

where  $a_i$ ,  $b_i$ , and  $c_i$  are the consequent parameters characterizing the shapes of the linear output membership function. In *aggregation layer*, the final output of the Sugeno system is the weighted sum of all rule outputs.

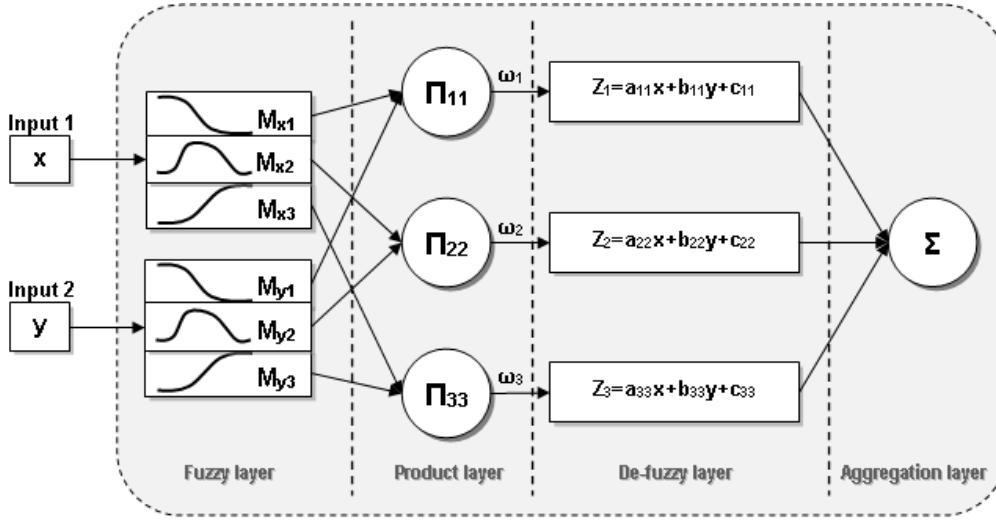


Figure 4.1: The Sugeno-type FIS

### Model configuration

In the experimental settings, we use the “GENFIS1” Matlab command to generate a standard Zero-Order Sugeno-type Fuzzy Inference System using a grid partition on data. For a zero-order Sugeno-type model, the output level  $z$  is a constant. We do not conduct parameter optimization, instead of focusing on comparing the difference among these different types of models. By default, “GENFIS1” uses two “gbellmf” type membership functions for each input. Each rule generated by “GENFIS1” has one output membership function, which allows us to enforce the zero order by means of setting the constant output membership function type. Then we use the “ANFIS” command to perform the optimization of the consequents. By default, a hybrid

method of combining least-squares with back-propagation gradient descent algorithm is used for training FIS membership function parameters to model a given set of input data. During the learning process of the FIS, the consequent parameters are tuned until the desired response of the FIS is achieved.

### 4.2.3 Fuzzy C-means Clustering (FCM)

#### Model description

In hard (non-fuzzy) clustering algorithms, data is divided into crisp clusters, that is each data point belongs to exactly one cluster. In fuzzy clustering algorithms, the membership function follows a smoother line to indicate that each data point may belong to several clusters with different values of the membership grade from 0 to 1. Fuzzy C-means (FCM) clustering is such an unsupervised data clustering algorithm assigning each data point into a cluster to some degree according on specified membership grades. The original technique introduced by J.C. Bezdek [8] as an improvement on earlier clustering methods proposed by J.C. Dunn [21]. Now FCM has been applied extensively in a variety of researches [3][34].

FCM provides a method that shows how to group data points that populate some mono-dimensional space into a specific number of different clusters, and is carried out through an iterative optimization (minimization) of the basic c-means objective function:

$$J_m = \sum_{i=1}^N \sum_{j=1}^C \mu_{ij}^m \|x_i - c_j\|^2, 1 \leq m \leq \infty \quad (4.3)$$

where  $m$  is any real number greater than 1;  $\mu_{ij}$  is the degree of membership of  $x_i$  in the cluster  $j$ ;  $x_i$  is the  $i$ th of  $d$ -dimensional measured data;  $c_j$  is the  $d$ -dimension center of the cluster; and  $\|x_i - c_j\|$  is any norm expressing the similarity between any measured data and the center.

Fuzzy partitioning is carried out through an iterative optimization of this objective function, with the update of membership  $\mu_{ij}$  and  $c_j$  by:

$$\mu_{ij} = \frac{1}{\sum_{k=1}^C \left( \frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}} \quad (4.4)$$

$$c_i = \frac{\sum_{i=1}^N \mu_{ij}^m x_i}{\sum_{i=1}^N \mu_{ij}^m} \quad (4.5)$$

The FCM clustering algorithm is composed of the following iterative steps.

1. Initialize  $U = [\mu_{ij}]$  matrix,  $U^{(0)}$
2. At k-step, calculate the centers vectors  $C^{(k)} = [c - j]$  with  $U^{(k)}$  using Equation 4.5
3. Update  $U^{(k)}, U^{(k+1)}$  using Equation 4.4
4. IF  $\|U^{(k+1)} - U^{(k)}\| < \varepsilon$  then STOP; otherwise return to step 2.

The iteration will stop when  $\max_{ij} \|\mu_{ij}^{(k+1)} - \mu_{ij}^{(k)}\| < \varepsilon$ , where  $\varepsilon$  is a termination criterion between 0 and 1,  $k$  is the iteration steps. This procedure converges to a local minimum or a saddle point of  $J_m$ .

### Model configuration

In Matlab, function “GENFIS3” is used to generate a FIS using FCM clustering to search for spherically distributed clusters, given the separate sets of input and output data. GENFIS3 accomplishes the task by extracting a set of rules that models the data behavior. The rule extraction method first uses a Sugeno-type FIS as the defaulted FCM function to determine the number of rules and membership functions for the antecedents and consequents. After that, “EVALFIS” simulates the FCM System for the input data and returns the predicted output data.

# Chapter 5

## Experimental results

### 5.1 Dutch charity database

#### 5.1.1 Ranking TPR fuzzy intervals

1. The sets of ROC curves are derived from each of LR, FIS, and FCM models as captured in Figure 5.1.

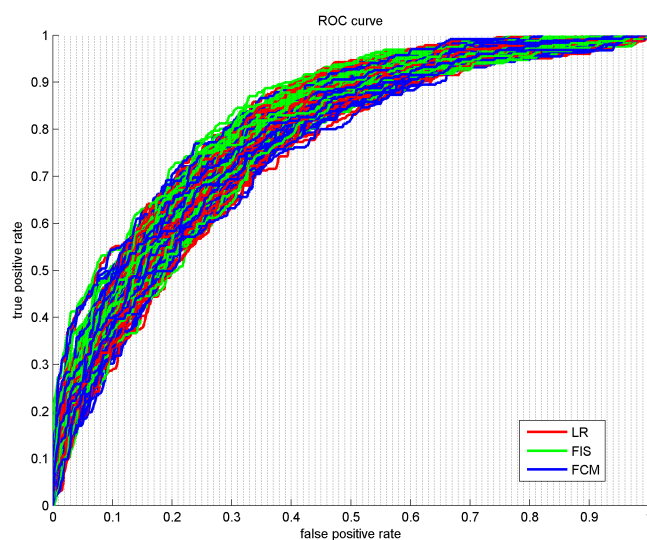


Figure 5.1: The ROC curves of LR, FIS, and FCM models

- The ROC curves are cut at each FPR from 0.01, 0.02, ... to 0.99. Figure 5.2 shows some examples of the distribution of TPRs at each FPR = 0.1 - 0.9.

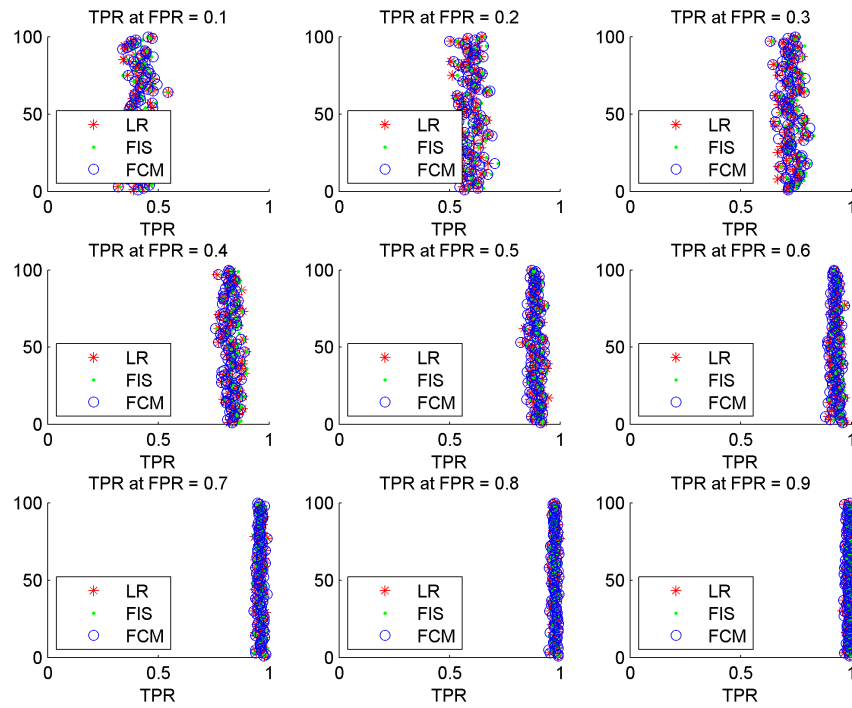


Figure 5.2: The distribution of TPRs at each selected FPR = 0.1 - 0.9

3. We map the TPRs at each  $FPR = 0.01 - 0.99$  into a fuzzy interval according to the five membership functions we described before. For example, the Figure 5.3 - Figure 5.11 illustrate the resulting fuzzy intervals derived from the five membership functions at each  $FPR = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8,$  and  $0.9$ , respectively.

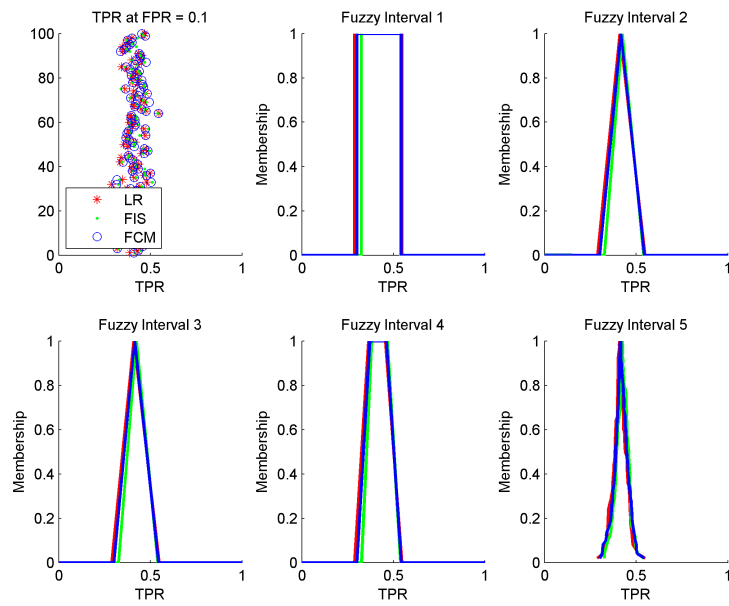


Figure 5.3: The fuzzy intervals of TPRs at  $FPR = 0.1$

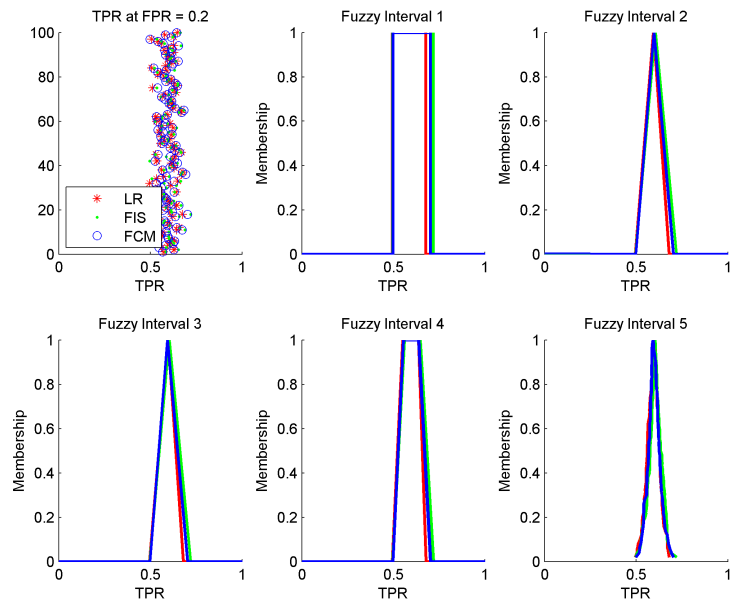


Figure 5.4: The fuzzy intervals of TPRs at  $FPR = 0.2$

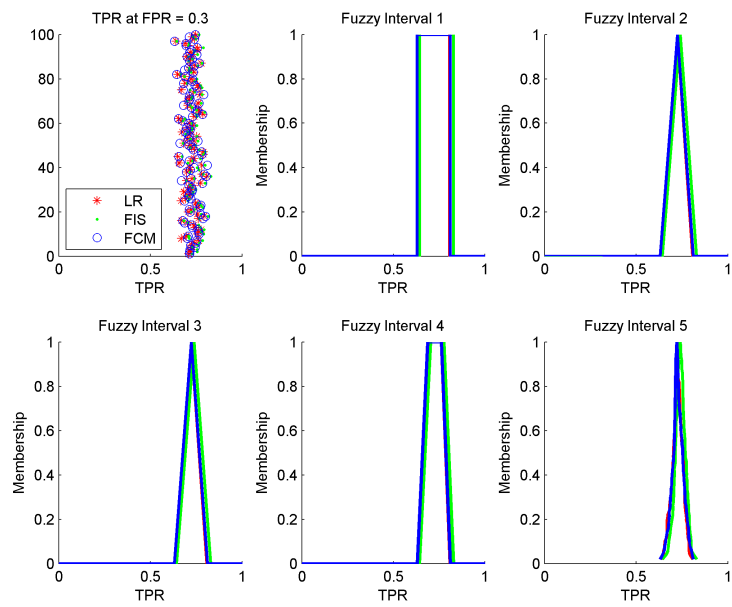


Figure 5.5: The fuzzy intervals of TPRs at  $FPR = 0.3$



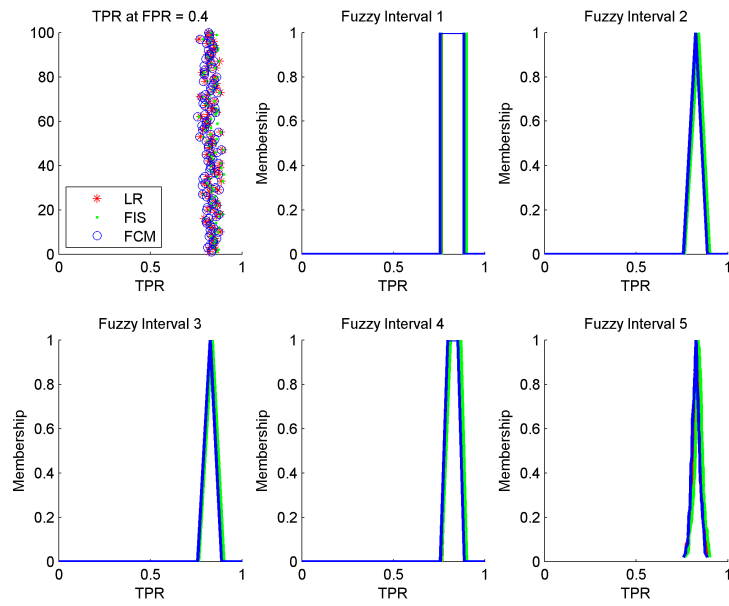


Figure 5.6: The fuzzy intervals of TPRs at  $FPR = 0.4$

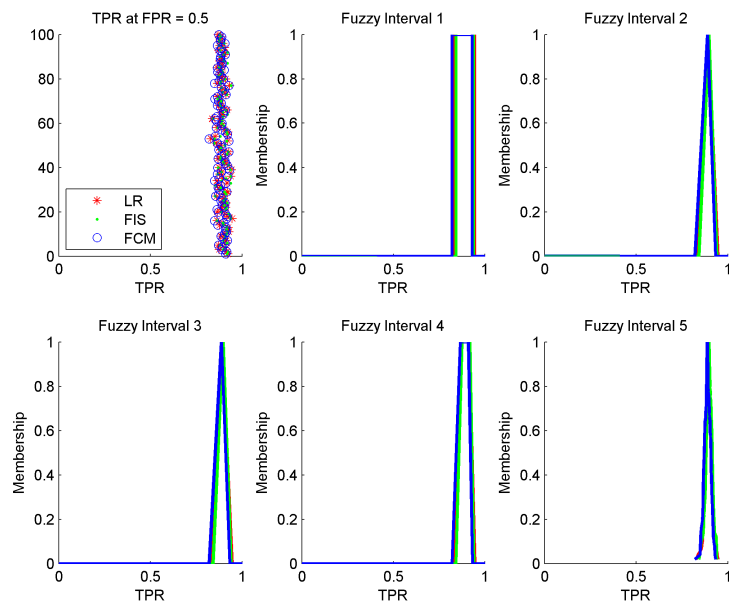


Figure 5.7: The fuzzy intervals of TPRs at  $FPR = 0.5$

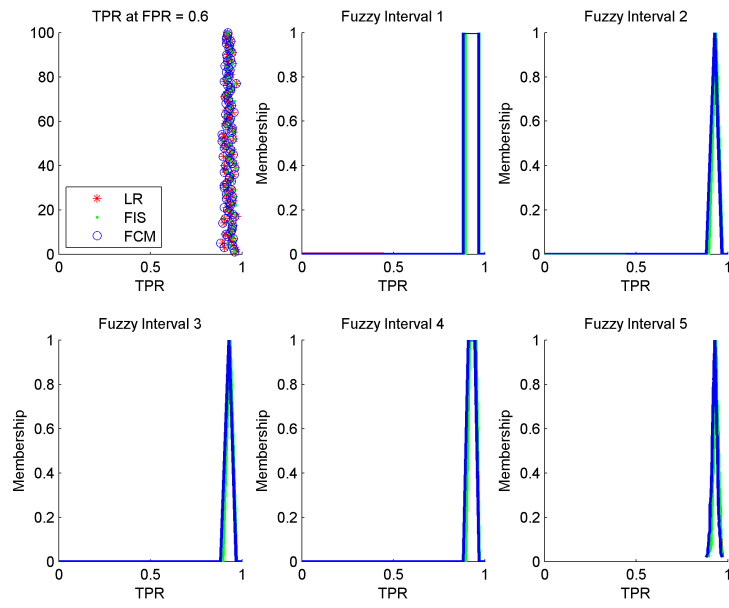


Figure 5.8: The fuzzy intervals of TPRs at  $FPR = 0.6$

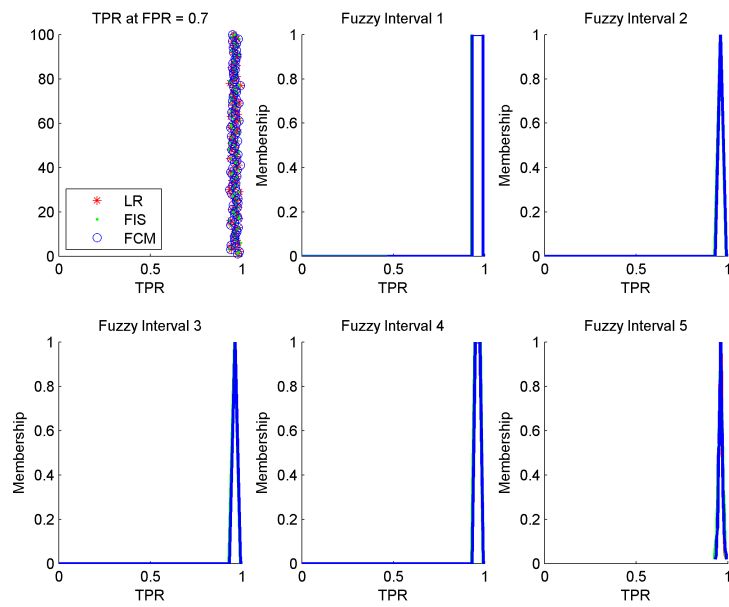


Figure 5.9: The fuzzy intervals of TPRs at  $FPR = 0.7$

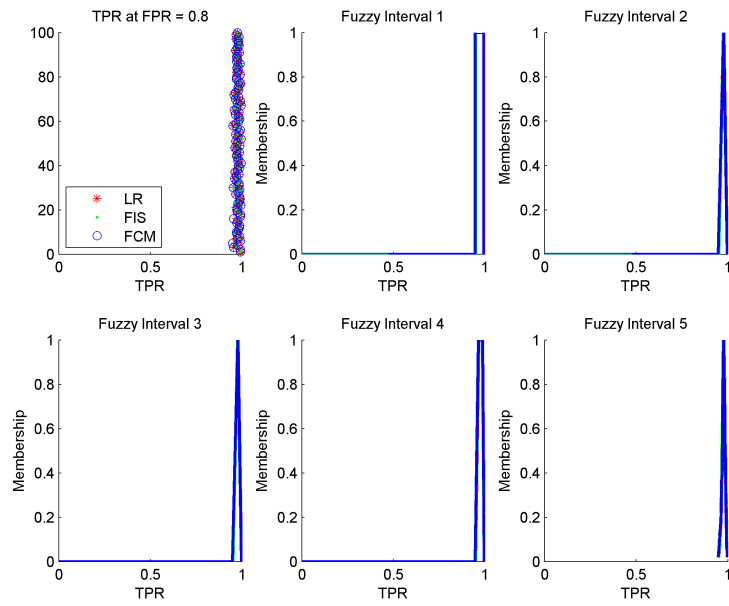


Figure 5.10: The fuzzy intervals of TPRs at FPR = 0.8

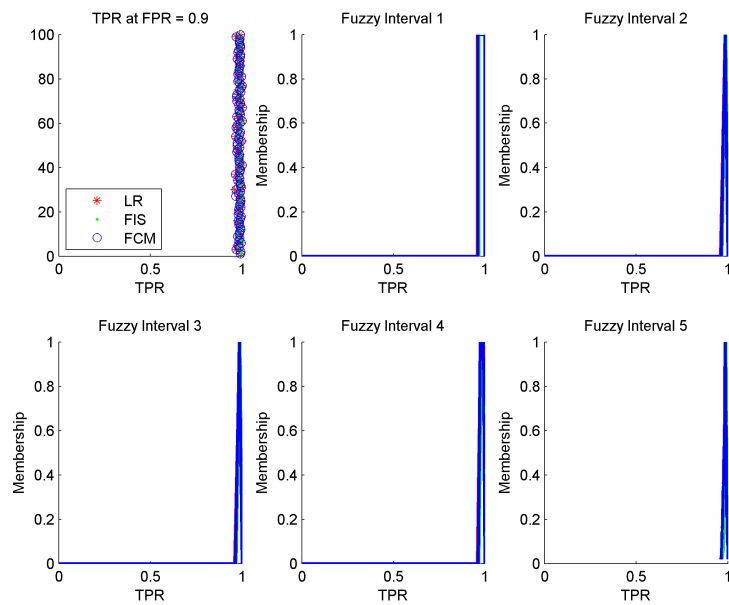


Figure 5.11: The fuzzy intervals of TPRs at FPR = 0.9

4. The index of fuzzy sets at each given FPR from 0.01 - 0.99 is calculated based on Yager's algorithms. Figure 5.12 and Figure 5.13 show some abstract results of the two Yager's indices at  $FPR = 0.1-0.9$  respectively.

The results drawn from the Yager's index of the center of gravity and the mean area are basically consistent, because both methods are based on obtaining a single real number from the average performance (general mean value) of fuzzy numbers.

In this model selection system with LR, FIS, and FCM models, it shows that FIS with the Sugeno-type fuzzy model performs best throughout the entire performance, except for slightly instabilities at the end of ROC curves.



### 5.1.2 Ranking AUC fuzzy intervals

- As the selected examples (TPR fuzzy sets at FPR 0.1 - 0.9 only) shown in Figure 5.14 - Figure 5.18, we analyze the five different shaped TPR fuzzy sets at each FPR from 0.01, 0.02, ... to 0.99, respectively. The  $\alpha$ -cut sets from  $\alpha = 0, 0.1, \dots$  to 1 are obtained, i.e. the crisp sets with the best element and the worst element of interval at each  $\alpha$ -cut.

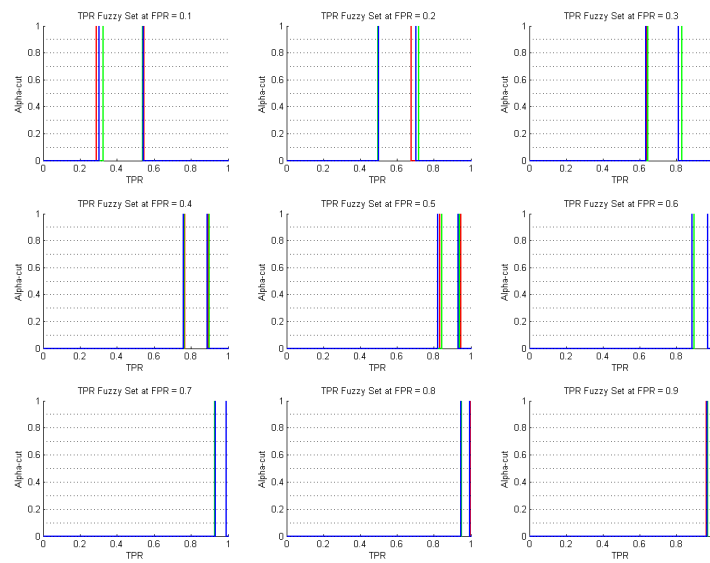


Figure 5.14: Sample crisp TPR fuzzy sets from FPR = 0.1 - 0.9

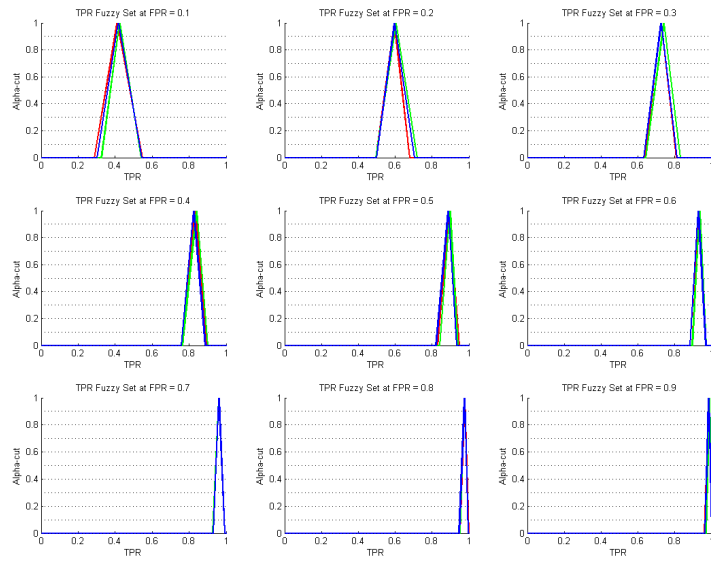


Figure 5.15: Sample triangular mean TPR fuzzy sets from FPR = 0.1 - 0.9

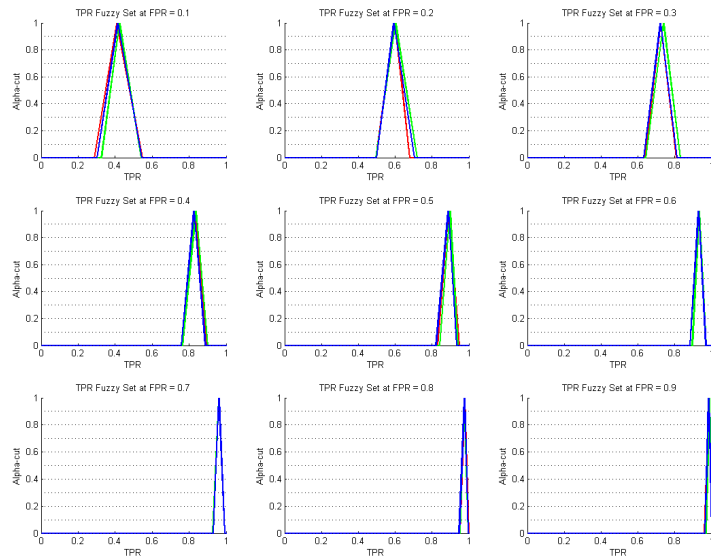


Figure 5.16: Sample triangular median TPR fuzzy sets from FPR = 0.1 - 0.9

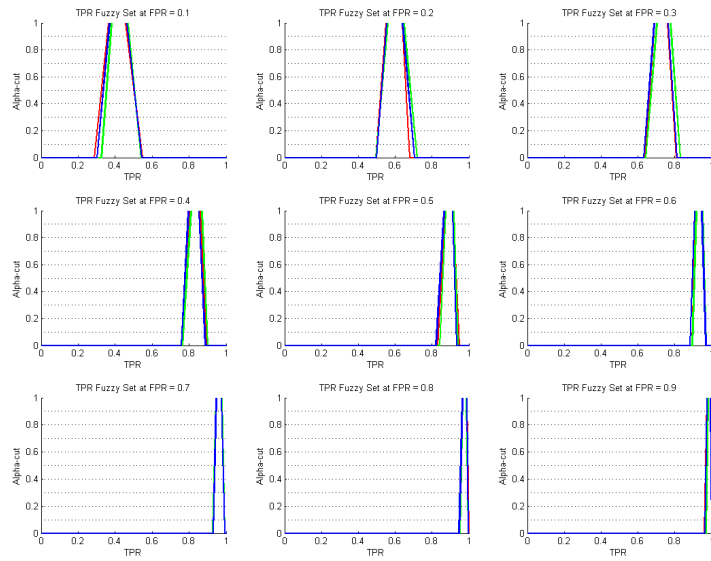


Figure 5.17: Sample trapezoidal TPR fuzzy sets from  $FPR = 0.1 - 0.9$

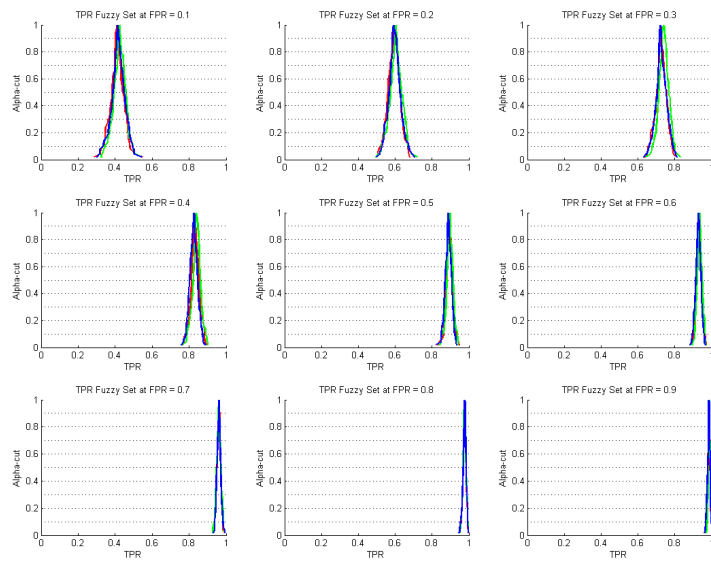


Figure 5.18: Sample enveloping TPR fuzzy sets from  $FPR = 0.1 - 0.9$



- As Figure 5.19 - Figure 5.23 illustrated, we use the best and worst TPR at each FPR from 0.01, 0.02, ...to 0.99 to form ROC curves for each  $\alpha$ -cut. Because ROC curves start from 0 and end at 1. Therefore, the best and worst ROC curve for each  $\alpha$ -cut are generated by connecting the best TPR and the worst TPR points of at FPR = 0, 0.01, ..., 0.99, and 1.

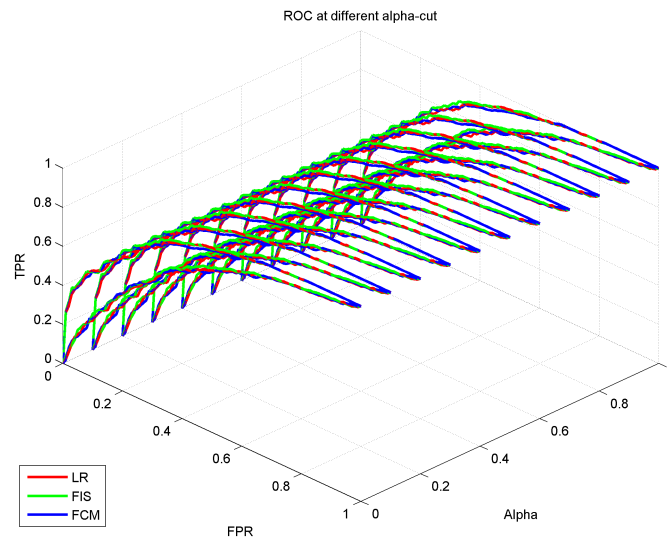


Figure 5.19: The best and worst ROC curves generated from the crisp TPR fuzzy sets

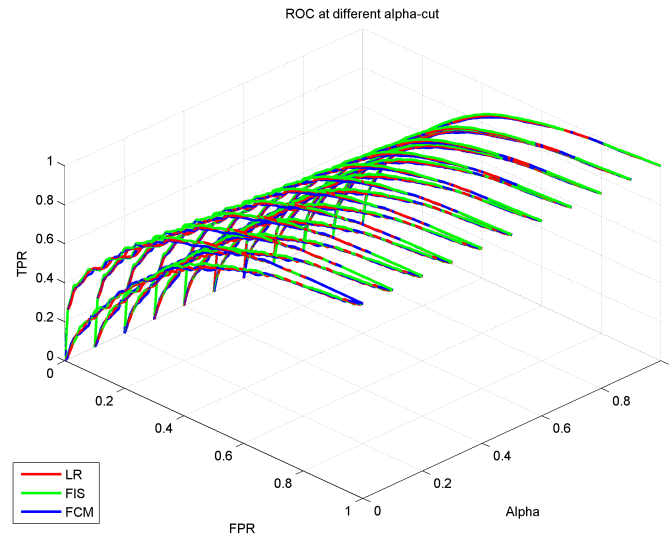


Figure 5.20: The best and worst ROC curves generated from the triangular mean TPR fuzzy sets

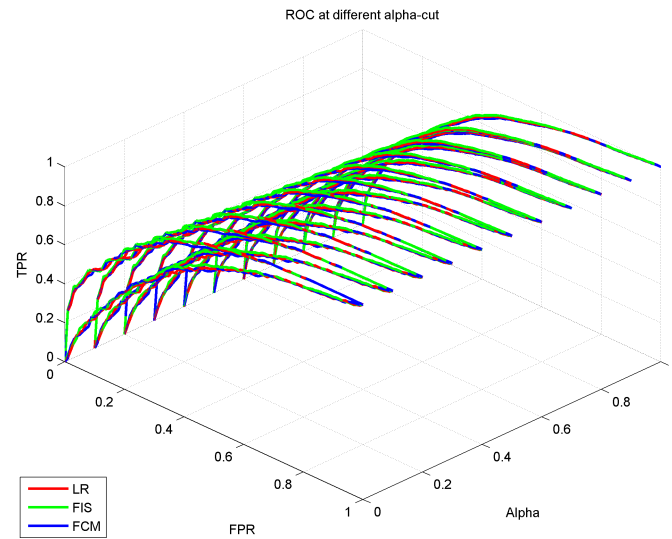


Figure 5.21: The best and worst ROC curves generated from the triangular median TPR fuzzy sets

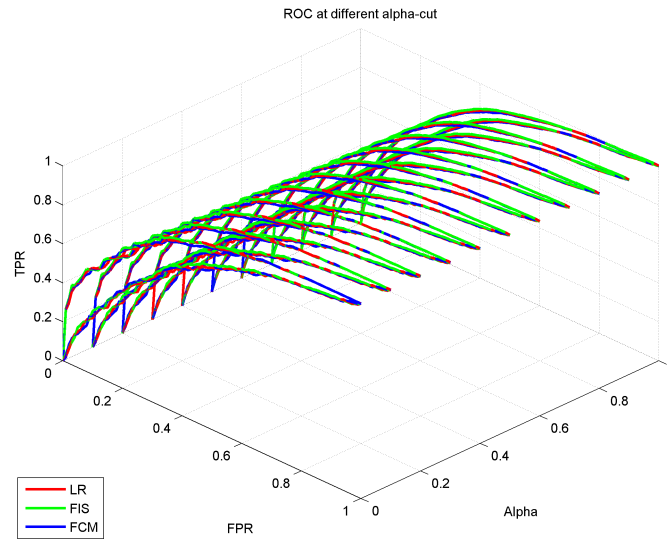


Figure 5.22: The best and worst ROC curves generated from the trapezoidal TPR fuzzy sets

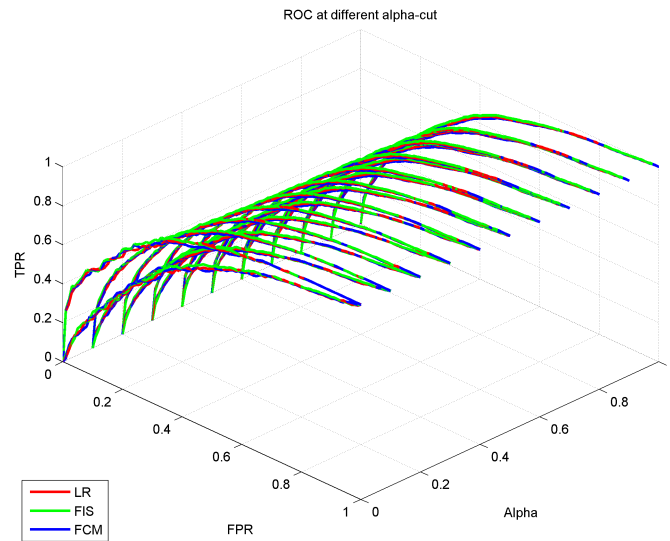


Figure 5.23: The best and worst ROC curves generated from the enveloping TPR fuzzy sets

- As Figure 5.24 shown, at each  $\alpha$ -cut, we calculate the best AUC value based on the best ROC curve and the worst AUC value based on the worst ROC curve. Accordingly, The AUC fuzzy interval is generated by collecting the crisp  $\alpha$ -cut set at each  $\alpha$  value.

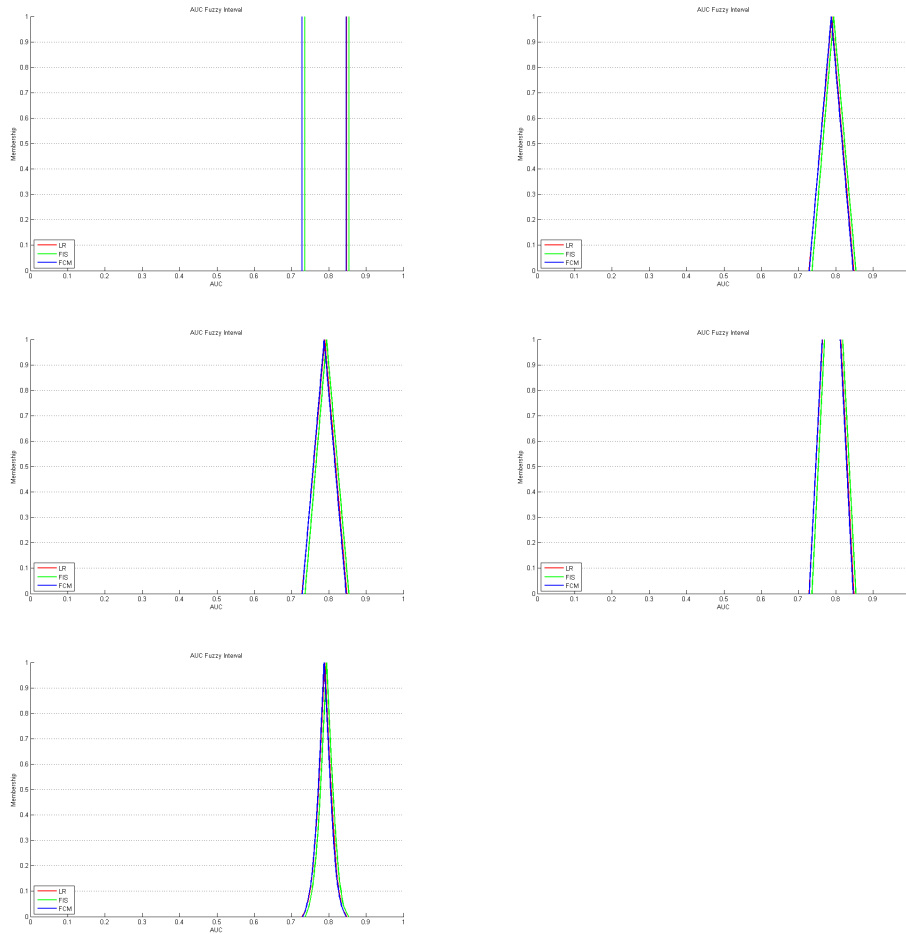


Figure 5.24: The AUC fuzzy set generated from 5 different-shaped TPR fuzzy sets

4. Ranking the AUC fuzzy intervals tells us the predictive performance of these three model at a macro level. Intuition favors the FIS model compared to the other two models, because it locates the right side in Figure 5.24, represented with green lines. Precisely, associating a crisp value with these intervals by using Yager's algorithms confirms that FIS model overall performs best with a highest value in Table 5.1. Table 5.1 shows the resulting Yager's index for each model with respect to different determined fuzzy intervals, as well as the traditional way to calculate AUC value by computing the average of the area under all ROC curves. The fuzzy analysis results in a compatible model selection with traditional AUC method.

The AUC fuzzy interval is considered as an global measure based on the combination of TPR fuzzy intervals at each FPR. As long as one classifier clearly dominates others over the entire performance, the resulting AUC value is basically consistent with the micro performance at each FPR.

Table 5.1: Defuzzified AUC value

		<i>Yager (1)</i>	<i>Yager (2)</i>	<i>Traditional average</i>
(1) crisp	<i>LR</i>	0.7796	0.7796	0.7894
	<i>FIS</i>	0.7850	0.7850	0.7946
	<i>FCM</i>	0.7784	0.7884	0.7883
(2) triangular mean	<i>LR</i>	0.7800	0.7807	0.7894
	<i>FIS</i>	0.7856	0.7866	0.7946
	<i>FCM</i>	0.7790	0.7802	0.7883
(3) triangular median	<i>LR</i>	0.7799	0.7806	0.7894
	<i>FIS</i>	0.7853	0.7858	0.7946
	<i>FCM</i>	0.7788	0.7894	0.7883
(4) trapezoid	<i>LR</i>	0.7802	0.7802	0.7894
	<i>FIS</i>	0.7858	0.7858	0.7946
	<i>FCM</i>	0.7793	0.7793	0.7883
(5) envelope	<i>LR</i>	0.7808	0.7806	0.7894
	<i>FIS</i>	0.7866	0.7858	0.7946
	<i>FCM</i>	0.7800	0.7794	0.7883

## 5.2 Haberman's survival database

Experiments with the databases described in Table 4.1 result in quite similar outcomes with those of the Dutch charity database. Briefly, the resulting TPR numbers drawn from the two Yager's indices are basically consistent. It also shows that the predictive performance at each FPR is in line with the predictive performance at a macro level interpreted by a defuzzified AUC value, because one of these three model is clearly dominate over the entire performance. Additional, the fuzzy analysis is comparable with the traditional AUC approach. Hence we do not show the details of the resulting model selection of these databases.

However, the experiment of Haberman's survival database is the situation where ROC curves of the three classifiers obviously intersect. In Figure 5.25, the interaction of the ROC curves of FIS with other ROC curves occurs around  $FPR = 0.5$ , which shows FIS performs worst before  $FPR = 0.5$ , but outperforms after  $FPR = 0.5$ .

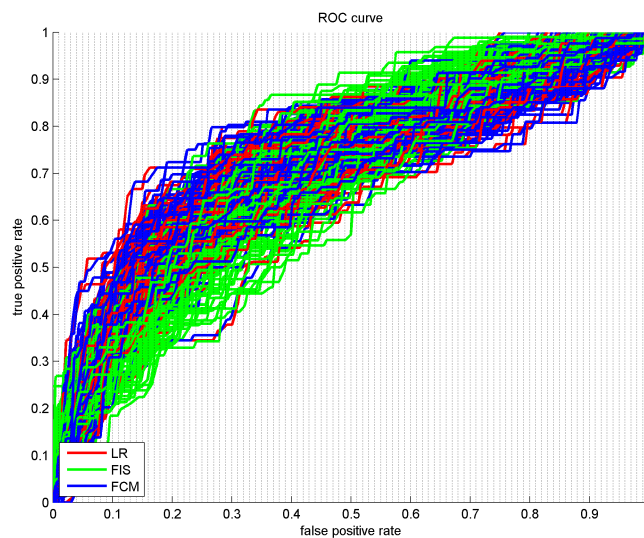


Figure 5.25: The ROC curves of LR, FIS, and FCM models

This can be easier detected by charting TPR points at each FPR from 0.01, 0.02, ... to 0.99. As the examples of the distribution of TPRs at each FPR = 0.1 - 0.9 shown in Figure 5.26, The TPR points of LR, FIS, and FCM are overlapped at FRP = 0.5. The green TPR points of FIS distributed slightly in the left side of those of LR (red) and FCM (blue) before FPR = 0.5, whereas these green point moves to the right side of red and blue points after FPR = 0.5.

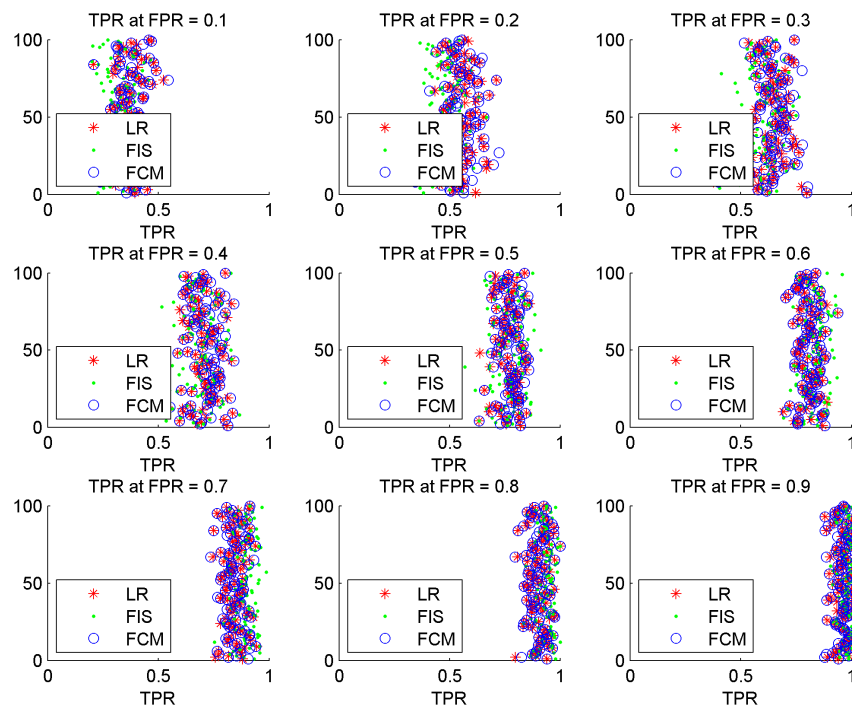


Figure 5.26: The distribution of TPRs at each selected FPR = 0.1 - 0.9

The analysis of TPR and AUC fuzzy intervals are illustrated as follows. Figure 5.27 - Figure 5.31 show the selected TPR fuzzy sets at FPR 0.1 - 0.9 in terms of five different determined membership functions, respectively.

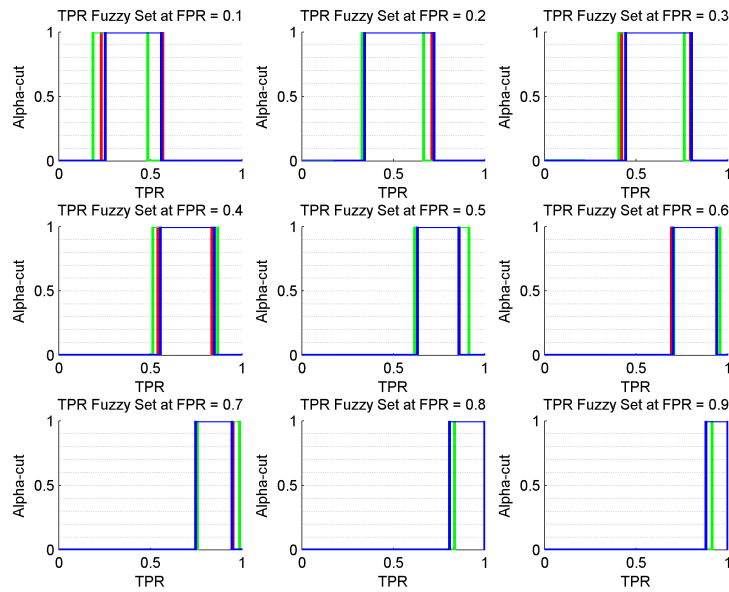


Figure 5.27: Sample crisp TPR fuzzy sets from FPR = 0.1 - 0.9



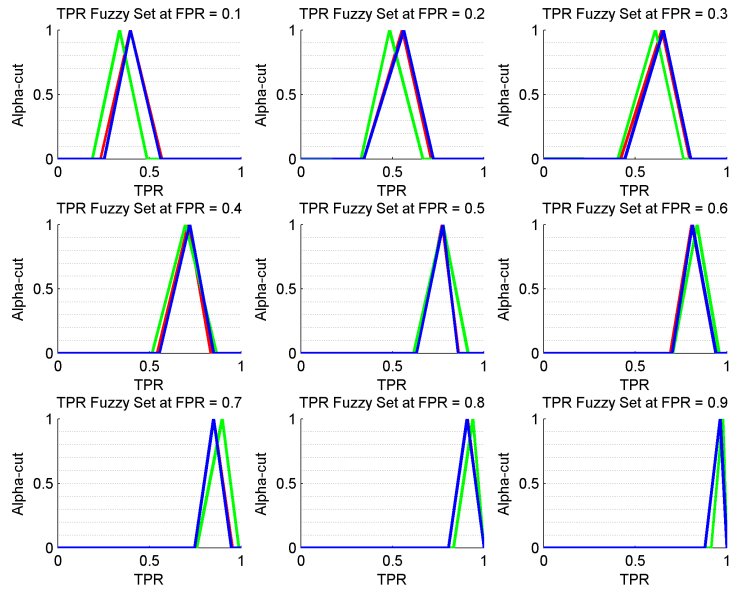


Figure 5.28: Sample triangular mean TPR fuzzy sets from FPR = 0.1 - 0.9

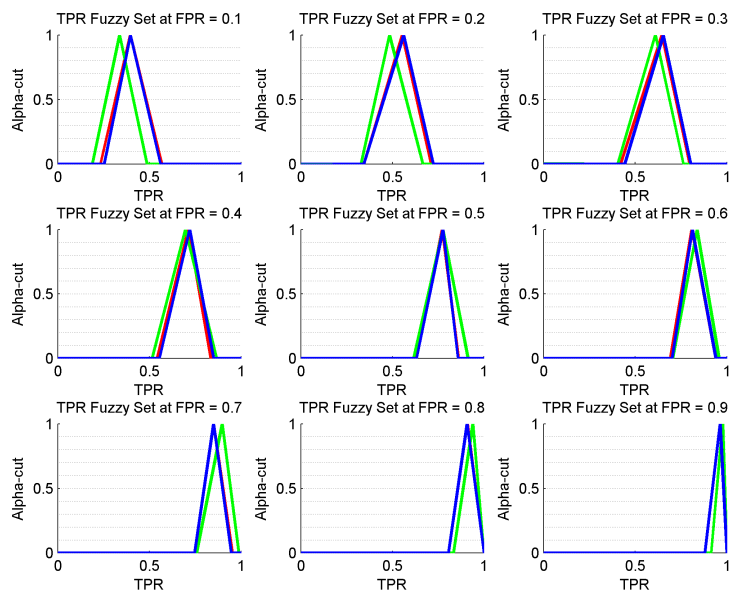


Figure 5.29: Sample triangular median TPR fuzzy sets from FPR = 0.1 - 0.9

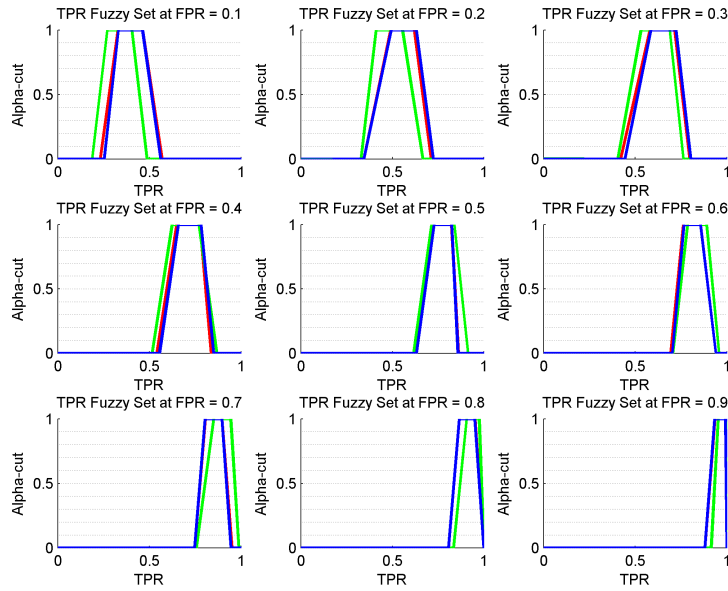


Figure 5.30: Sample trapezoidal TPR fuzzy sets from  $FPR = 0.1 - 0.9$

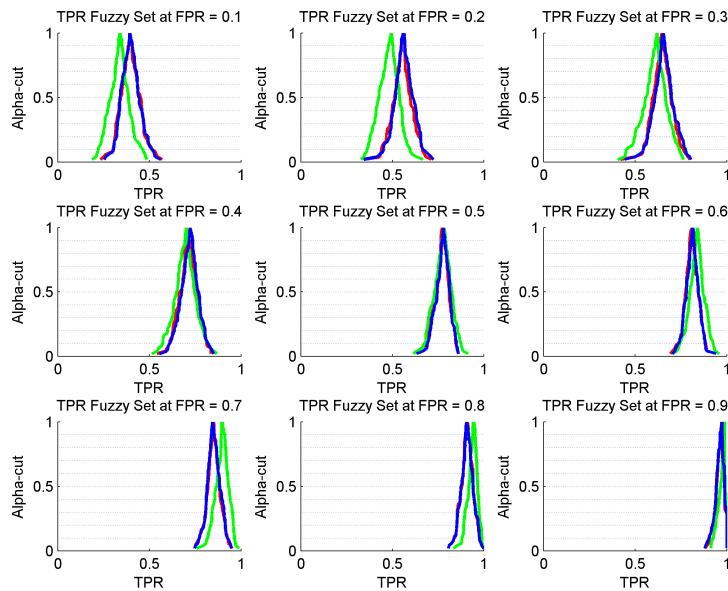


Figure 5.31: Sample enveloping TPR fuzzy sets from  $FPR = 0.1 - 0.9$

Figure 5.32 shows the corresponding AUC fuzzy interval generated by collecting the crisp  $\alpha$ -cut set at each  $\alpha$  value in terms of five different determined membership function, respectively. Table 5.2 contains the corresponding defuzzified AUC values.

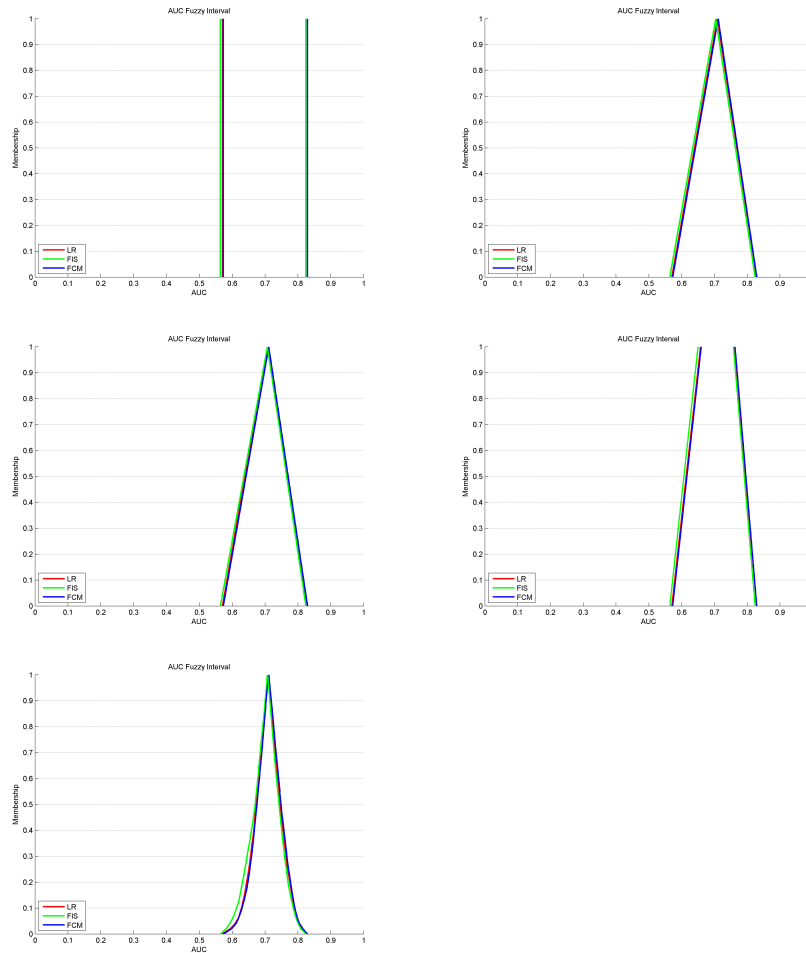


Figure 5.32: The AUC fuzzy set generated from 5 different-shaped TPR fuzzy sets

The macro performance presented by the AUC fuzzy intervals shows FCM model is best, while there is not a big distinction between these three models. The defuzzified AUC values are in line with the average AUC values calculated in a traditional way. However, the analysis TPR fuzzy intervals sheds

Table 5.2: Defuzzified AUC value

		<i>Yager (1)</i>	<i>Yager (2)</i>	<i>Traditional average</i>
(1) crisp	<i>LR</i>	0.6995	0.6995	0.7092
	<i>FIS</i>	0.6940	0.6940	0.7043
	<i>FCM</i>	0.7009	0.7009	0.7114
(2) triangular mean	<i>LR</i>	0.7027	0.7091	0.7092
	<i>FIS</i>	0.6974	0.7040	0.7043
	<i>FCM</i>	0.7044	0.7113	0.7114
(3) triangular median	<i>LR</i>	0.7027	0.7090	0.7092
	<i>FIS</i>	0.6984	0.7072	0.7043
	<i>FCM</i>	0.7043	0.7110	0.7114
(4) trapezoid	<i>LR</i>	0.7043	0.7043	0.7092
	<i>FIS</i>	0.6990	0.6990	0.7043
	<i>FCM</i>	0.7061	0.7061	0.7114
(5) envelope	<i>LR</i>	0.7091	0.7090	0.7092
	<i>FIS</i>	0.7040	0.7072	0.7043
	<i>FCM</i>	0.7114	0.7110	0.7114

light on an inconsistent performance at micro level. In this sense, decision maker should pay more attention when some special specificity is required.

# Chapter 6

## Conclusion and discussion

### 6.1 Main findings

In this study we have developed a small computer tool that can efficiently select best models among alternatives under sampling uncertainty for binary predictions (source code available in Appendix). The implementation of the bootstrap algorithm simulates the sampling uncertainty based on the distribution of the original database and reduces the computational complexity when dealing with extremely large data sets. The introduction of fuzzy numbers enables to analyze the variation from sampling uncertainty and deal with complicated scenarios when ROC curves intersect. The model selection can be based on either overall performance or at any required false positive cut-off threshold, which can be flexibly supplied by the user.

So far, we have included three classification models: “Logistic regression”, “Fuzzy inference system with the Sugeno model”, and “Fuzzy C-means clustering”, and used Yager’s index of the center of gravity and the mean area for ranking fuzzy numbers. We have tested our method on several real-world data sets. In general, the best model selected using our method is the one with the maximum defuzzified number.

- Among the tested data sets, the scenario that the selection based on a micro level disagrees with a global AUC based selection is rare. This is most likely explained by the fact that one classifier clearly outper-

forms compared with others, and these data sets have been well quality controlled before analysis and lack of hidden heterogeneity.

- The resulting numbers from Yager’s index of the center of gravity and the mean area are consistent, because both algorithms measure the general mean of fuzzy numbers.
- The five different shaped AUC fuzzy subsets: “the regular crisp interval function”, “the triangular mean function”, “the triangular median function”, “the trapezoidal function with mean and standard deviation”, and “the fitting function of envelope” represent the AUC value with the different level of information accessible. The results derived from the five different shaped AUC fuzzy subsets are compatible, as well as in line with the transitional average AUC value. It implicates that one can draw a relatively reliable selection even based on limited information available.

In conclusion, the main contributions of the paper are:

1. A fuzzy presentation of ROC curves has been provided. The algorithm addresses not only the predictive accuracy but also the degree of sampling uncertainty, based on which the confidence intervals of ROC curves can be efficiently and reliably estimated. The uncertainty-fuzziness model selection system enables to describe imprecise information. It is capable of representing dubious, incomplete, and fragmentary information in the fuzziness description. Therefore the proposed system benefits from the comparison of classifiers in imprecise environments or environments without rich information available.
2. The fuzzification transformation of TPR intervals into a AUC interval has been presented. It simplifies quantitative information to a lower information content and thus eases the procedure of uncertainty quantification. The compatible results with the traditional average AUC values indicate that our method is reliable. The major benefit is that the worst & best performance boundaries due to sampling uncertainty can be captured and estimated efficiently.

3. In addition to an overall accuracy evaluation, the implemented fuzzy algorithm also allows specific accuracy analysis based on the FPR thresholds required by users. This is particularly important in many real applications such as clinical medicine or economics. For example, if a certain operation on a cancer patient requires the diagnosis to be 99% sure (*specificity*), the model selection should depend on the predictive performance up to the FPR threshold of 0.01 ( $1 - \textit{specificity}$ ), instead of the full range integration of the ROC curves.

## 6.2 Future research

There remains a plenty of spaces where the developed tool can be extended. For example, the method now focuses on binary prediction where multi-class responses are dichotomized to one vs. the remaining classes. The approach can be easily extended to allow the prediction of multi-class responses, as well as quantitative outcomes. Also, more criteria of ranking fuzzy numbers can be included in case of some special scenarios. So far we have implemented three alternative prediction models. The system can be extended by including more prediction models in the future.

# Bibliography

- [1] H. Aikake. Information theory and an extension of the maximum likelihood principle. In *Proc, 2nd International Symp. on Inference Theory*, pages 267–281, 1973.
- [2] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.
- [3] M. Alata, M. Molhim, and A. Ramini. Optimizing of Fuzzy C-Means Clustering Algorithm Using GA. *Update*, 1:5.
- [4] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.
- [5] B.M. Ayyub. *Uncertainty modeling and analysis in engineering and the sciences*. Chapman & Hall/CRC, 2006.
- [6] P. Baldi, S. Brunak, Y. Chauvin, C.A.F. Andersen, and H. Nielsen. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, 16(5):412–424, 2000.
- [7] A. Ben-David. What’s Wrong with Hit Ratio? *IEEE Intelligent Systems*, pages 68–70, 2006.
- [8] J.C. Bezdek. *Pattern recognition with fuzzy objective function algorithms*. Kluwer Academic Publishers Norwell, MA, USA, 1981.
- [9] G. Bortolan and R. Degani. Review of some methods for ranking fuzzy subsets. *Fuzzy Sets and Systems*, 15(1):1–20, 1985.



- [10] A.P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.
- [11] ST Buckland, KP Burnham, and NH Augustin. Model selection: an integral part of inference. *Biometrics*, pages 603–618, 1997.
- [12] K.P. Burnham and D.R. Anderson. Kullback-Leibler information as a basis for strong inference in ecological studies. *Wildlife Research*, 28(2):111–120, 2001.
- [13] JM Cadenas and JL Verdegay. Using ranking functions in multiobjective fuzzy linear programming. *Fuzzy sets and systems*, 111(1):47–54, 2000.
- [14] C. Carlsson and R. Fuller. *Fuzzy reasoning in decision making and optimization*. Physica Verlag, 2002.
- [15] S.H. Chen. Ranking fuzzy numbers with maximizing set and minimizing set. *Fuzzy Sets and Systems*, 17(2):113–129, 1985.
- [16] S.J. Chen and S.M. Chen. A new method for handling multicriteria fuzzy decision-making problems using FN-IOWA operators. *Cybernetics and Systems*, 34(2):109–137, 2003.
- [17] C.H. Cheng. A new approach for ranking fuzzy numbers by distance method. *IFSA 97 Prague Proceedings Volume , 159~ 164 ( 6 )*, 1997.
- [18] F. Choobineh and H. Li. An index for ordering fuzzy numbers. *Fuzzy Sets and Systems*, 54(3):287–294, 1993.
- [19] M. Detyniecki and R.R. Yager. Ranking fuzzy numbers using alpha-weighted valuations. *International Journal of Uncertainty Fuzzyness and Knowledge Based Systems*, 8(5):573–592, 2000.
- [20] N. Dukic and Z. Avdagic. Fuzzy functional dependency and the resolution principle. *Informatika*, 16(1):45–60, 2005.

- [21] JC Dunn. Some recent investigations of a new fuzzy partitioning algorithm and its application to pattern classification problems. *Cybernetics and Systems*, 4(2):1–15, 1974.
- [22] B. Efron and R.J. Tibshirani. *An introduction to the bootstrap*. Chapman & Hall, 1997.
- [23] J.P. Egan. *Signal detection theory and ROC-analysis*. Academic Pr, 1975.
- [24] T. Fawcett. An introduction to ROC analysis. *Pattern recognition letters*, 27(8):861–874, 2006.
- [25] C. Ferri and J. Hernández-Orallo. Cautious classifiers. In *Proceedings of the ROC Analysis in Artificial Intelligence, 1st International Workshop*, pages 27–36, 2004.
- [26] C. Ferri, J. Hernández-Orallo, and R. Modroiu. An experimental comparison of performance measures for classification. *Pattern Recogn. Lett.*, 30(1):27–38, 2009.
- [27] P. Fortemps and M. Roubens. Ranking and defuzzification methods based on area compensation. *Fuzzy Sets and Systems*, 82(3):319–330, 1996.
- [28] C.P. Friedman and J.C. Wyatt. *Evaluation methods in medical informatics*. Springer, 2006.
- [29] R. Fuller and C. Carlsson. Fuzzy multiple criteria decision making: Recent developments. *Fuzzy sets and systems*, 78(2):139–153, 1996.
- [30] M. Goumas and V. Lygerou. An extension of the PROMETHEE method for decision making in fuzzy environment: ranking of alternative energy exploitation projects. *European Journal of Operational Research*, 123(3):606–613, 2000.

- [31] J.G. Greenhut, ML Greenhut, and Y. Mansur. Oligopoly and behavioral uncertainty: an application of fuzzy set theory. *Review of Industrial Organization*, 10(3):269–288, 1995.
- [32] K. Guney and N. Sarikaya. Comparison of Mamdani and Sugeno fuzzy inference system models for resonant frequency calculation of rectangular microstrip antennas.
- [33] D.J. Hand and R.J. Till. A simple generalisation of the area under the ROC curve for multiple class classification problems. *Machine Learning*, 45(2):171–186, 2001.
- [34] DB Hou, Z.K.Z.D. Yu, and ZK Zhou. Research on the method of neural network modeling based on fcm algorithm and its application on vision-based sensors. In *Progress In Electromagnetics Research Symposium 2005, Hangzhou, China, August 22*, volume 26.
- [35] R. John. Fuzzy Sets and Knowledge Representation. *Fuzzy systems in medicine*, page 78, 2000.
- [36] George Kapetanios. Model selection uncertainty and dynamic models. NIESR Discussion Papers 165, National Institute of Economic and Social Research, May 2000.
- [37] F.O. Karray and C.W. De Silva. *Soft computing and intelligent systems design: theory, tools, and applications*. Addison-Wesley, 2004.
- [38] M. Kearns, Y. Mansour, A.Y. Ng, and D. Ron. An experimental and theoretical comparison of model selection methods. *Machine Learning*, 27(1):7–50, 1997.
- [39] S. Kikuchi. A method to defuzzify the fuzzy number: transportation problem application. *Fuzzy Sets and Systems*, 116(1):3–10, 2000.
- [40] R. Klinkenberg and T. Joachims. Detecting concept drift with support vector machines. In *Proceedings of the Seventeenth International Conference on Machine Learning table of contents*, pages 487–494. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 2000.

- [41] G.J. Klir and B. Yuan. *Fuzzy sets and fuzzy logic: theory and applications*. Prentice Hall Upper Saddle River, NJ, 1995.
- [42] S. Mabuchi. An approach to the comparison of fuzzy subsets with an  $\alpha$ -cutdependent index. *IEEE Transactions on Systems, Man and Cybernetics*, 18(2):264–272, 1988.
- [43] B. Moller, M. Beer, and M. Liebscher. Fuzzy analysis as alternative to stochastic methods—theoretical aspects. In *Proceedings of the 4th German LS-DYNA Forum*, 2005.
- [44] M.S. Pepe, T. Cai, and Z. Zhang. Combining predictors for classification using the area under the ROC curve. *Biometrics*, 62:221–229, 2005.
- [45] T. Pietraszek. On the use of ROC analysis for the optimization of abstaining classifiers. *Machine Learning*, 68(2):137–169, 2007.
- [46] W. Pijls and R. Potharst. Classification and target group selection based upon frequent patterns. In *Proceedings of the Twelfth Belgium-Netherlands Artificial Intelligence Conference (BNAIC 00)*, pages 125–132, 2000.
- [47] R. Potharst, U. Kaymak, and W. Pijls. Neural networks for target selection in direct marketing.
- [48] DMW Powers. Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation, School of Informatics and Engineering, Flinders University Adelaide Australia. Technical report, Technical Report SIE-07-001, 2007.
- [49] F. Provost and T. Fawcett. Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, pages 43–48. AAAI Press, 1997.
- [50] F. Provost and T. Fawcett. Robust classification for imprecise environments. *Machine Learning*, 42(3):203–231, 2001.

- [51] M. Setnes and U. Kaymak. Fuzzy modeling of client preference from large data sets: an application to target selection in direct marketing. *IEEE Transactions on Fuzzy Systems*, 9(1):153–163, 2001.
- [52] K.A. Smith and J.N.D. Gupta. *Neural networks in business: techniques and applications*. Irm Press, 2003.
- [53] J.M. Sousa, U. Kaymak, and S. Madeira. A comparative study of fuzzy target selection methods in direct marketing. In *Proceedings of the 11th IEEE International Conference on Fuzzy Systems*, 2002.
- [54] L. Tran and L. Duckstein. Comparison of fuzzy numbers using a fuzzy distance measure. *Fuzzy Sets and Systems*, 130(3):331–342, 2002.
- [55] S. Vanderlooy, I.G. Sprinkhuizen-Kuyper, and E.N. Smirnov. Reliable classifiers in ROC space. In *Proceedings of the 15th BENELEARN Machine Learning Conference*, pages 27–36. Citeseer, 2006.
- [56] R.R. Yager. Ranking fuzzy subsets over the unit interval. In *1978 IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes*, volume 17, 1978.
- [57] R.R. Yager. On the measure of fuzziness and negation part I: Membership in the unit interval. *International Journal of General Systems*, 5(4):221–229, 1979.
- [58] R.R. Yager. A procedure for ordering fuzzy subsets of the unit interval. *Information Sciences*, 24(2):143–161, 1981.
- [59] D. Yong and L. Qi. A TOPSIS-based centroid-index ranking method of fuzzy numbers and its application in decision-making. *Cybernetics and Systems*, 36(6):581–595, 2005.
- [60] L. A. Zadeh. Knowledge representation in fuzzy logic. *IEEE Trans. on Knowl. and Data Eng.*, 1(1):89–100, 1989.
- [61] LA Zadeh. Fuzzy Sets, Information and Control, Vol. 8. No, 3:338–353, 1965.

- [62] LA Zadeh. Calculus of fuzzy restrictions. *World Scientific Series In Advances In Fuzzy Systems*, pages 210–237, 1996.
- [63] LA Zadeh. Some reflections on soft computing, granular computing and their roles in the conception, design and utilization of information/intelligent systems. *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, 2(1):23–25, 1998.
- [64] J. Zahavi and N. Levin. Issues and problems in applying neural computing to target marketing. *Journal of Direct Marketing*, 9(3), 1995.
- [65] H.J. Zimmermann. *Fuzzy sets, decision making, and expert systems*. Kluwer Academic Pub, 1987.

# Appendix A

## Matlab scripts

### A.1 Main script

## Contents

- [Database loading](#)
- [Data preparation - removing highly correlated predictors](#)
- [Bootstrap-sampling](#)
- [TPR and FPR derived from bootstrap samples modeling](#)
- [ROC curves plotting](#)
- [TPRx100 with respect to each FPR](#)
- [TPR distribution plotting:](#)
- [TPR mapping into FI and plotting](#)
- [TPR fuzzy numbers defuzzifying](#)
- [TPR fuzzying intervals transferring into AUC fuzzy intervals \(1\)](#)
- [TPR fuzzying intervals transferring into AUC fuzzy intervals \(2\)](#)
- [TPR fuzzying intervals transferring into AUC fuzzy intervals \(3\)](#)
- [TPR fuzzying intervals transferring into AUC fuzzy intervals \(4\)](#)
- [TPR fuzzying intervals transferring into AUC fuzzy intervals \(5\)](#)

## Database loading

different with the different format of databases

```
clear
load tarseldata.mat

data1=[ytest,Xtest];
data2=[ytrain,Xtrain];
data=[data1;data2];
data(any(isnan(data),2),:) = [];

clear var_list ytest Xtest ytrain Xtrain data1 data2
```

## Data preparation - removing highly correlated predictors

```
y=data(:,1);
if (min(y)==0),y=y+1;end
X=data(:,2:end);
n=size(X,2);
D = zeros(1,n);           % model deviance
for i=1:n                 % over m predictors
    x=X(:,i);
    [B,d] = mnrfit(x,y);
    D(i)=d;
end

cor=corr(X);
th = 0.9;                 % correlation threshold
ex = zeros(n,1);         % exclude index
for i=1:n
    for j=(i+1):n
        if (cor(i,j)>th)   % if i and j are highly correlated
            if(D(i)>=D(j)) % exclude the one with higher deviance
                ex(i)=ex(i)+1;
            else
                ex(j)=ex(j)+1;
            end
        end
    end
end

disp(ex);
X(:,ex>0)=[];           % remove variables indexed by ex
data=[y,X];             % new data
```

## Bootstrap-sampling

[BOOTSTAT,BOOTSAM] = BOOTSTRP(NBOOT,BOOTFUN,D1,...) draws NBOOT bootstrap data samples, returns BOOTSAM, a matrix of indices into the rows of the extra arguments.



To get the output samples BOOTSAM without applying a function, set BOOTFUN to empty ([]).

```
% initialize
nrep = 100;
per = .1;

% bootstrap
[a,b]=bootstrp(nrep,[],data);
n=floor(size(data,1)*per);
sample=zeros(n,size(data,2),nrep);

for i=(1:nrep)
    % a book (3-dimensional matrix) with 100 pages
    sample(:, :, i) = data(b(1:n,i),:);
    if(var(sample(:,1,i))==0)
        error('Variable Y has no variance')
    end
end

clear a b i n per nrep
save resamples
```

## TPR and FPR derived from bootstrap samples modeling

```
clear
load resamples.mat

num=100;
auc1=zeros(num,1);
auc2=zeros(num,1);
auc3=zeros(num,1);
tfpr1=zeros(1001,2);
tfpr2=zeros(1001,2);
tfpr3=zeros(1001,2);
for i = 1:num
    X= sample(:,2:end,i);
    Y= sample(:,1,i);
    % for multiclass Y=1/2/3: Y(Y==3)=2;
    [auc1(i),t1]=mymtfpr(Y,X,1);
    [auc2(i),t2]=mymtfpr(Y,X,2);
    [auc3(i),t3]=mymtfpr(Y,X,3);
    if i==1
        tfpr1=t1;
        tfpr2=t2;
        tfpr3=t3;
    else
        tfpr1=cat(3,tfpr1,t1);
        tfpr2=cat(3,tfpr2,t2);
        tfpr3=cat(3,tfpr3,t3);
    end
end

AUC1=mean(auc1); % traditional calculate the average of auc
AUC2=mean(auc2); % traditional calculate the average of auc
AUC3=mean(auc3); % traditional calculate the average of auc

clear i a b c X Y num
save all
```

## ROC curves plotting

```
clear
load all.mat

figure;
hold on;
num=100;
x=0:.01:1;
for i = 1:num
    plot(tfpr1(:,1,i),tfpr1(:,2,i),'LineWidth',2, 'linestyle','-', 'color','r');
    plot(tfpr2(:,1,i),tfpr2(:,2,i),'LineWidth',2, 'linestyle','-', 'color','g');
    plot(tfpr3(:,1,i),tfpr3(:,2,i),'LineWidth',2, 'linestyle','-', 'color','b');
end
for j=1:length(x)
```

```

    % plot xgrids
    line(x(j)*[1 1],ylim,'color','k','LineStyle',':', 'LineWidth',.5);
end
title 'ROC curve';
xlabel('false positive rate');
ylabel('true positive rate');
xlim([0 1]);
ylim([0 1]);
h = legend('LR','FIS','FCM',3);
set(h,'Interpreter','none')
hold off;
fnam = 'ROC';
print ('-dtiff', '-r300', fnam);

```

## TPRx100 with respect to each FPR

```

clear
load all.mat

% FPR = 0.1, 0.2, ..., to 0.9
th=.1:.1:.9;
dim=size(tfpr1);
num=dim(3);
tpr=zeros(num,3,length(th));

for i=1:length(th)
    fpr=th(i);
    ix=round((dim(1)-1)*fpr+1);
    tpr1=reshape(tfpr1(ix,2,:),num,1);
    tpr2=reshape(tfpr2(ix,2,:),num,1);
    tpr3=reshape(tfpr3(ix,2,:),num,1);
    tpr(:, :, i)=[tpr1,tpr2,tpr3];
end

save 'TPR.mat' tpr;

% FPR = 0.01, 0.02, ..., to 0.99
th=.01:.01:.99;
dim=size(tfpr1);
num=dim(3);
tpr99=zeros(num,3,length(th));

for i=1:length(th)
    fpr=th(i);
    ix=round((dim(1)-1)*fpr+1);
    tpr1=reshape(tfpr1(ix,2,:),num,1);
    tpr2=reshape(tfpr2(ix,2,:),num,1);
    tpr3=reshape(tfpr3(ix,2,:),num,1);
    tpr99(:, :, i)=[tpr1,tpr2,tpr3];
end

save 'TPR99.mat' tpr99;

```

## TPR distribution plotting:

```

clear;
load TPR.mat

dim=size(tpr);
th=.1:.1:.9;
figure;
for i=1:dim(3)
    SUBPLOT(3,3,i)
    xlim([0 1]);
    xlabel('TPR');
    hold on
    scatter(tpr(:,1,i), 1:dim(1),'*r')
    scatter(tpr(:,2,i), 1:dim(1),'*g')
    scatter(tpr(:,3,i), 1:dim(1),'*ob')
    title(['TPR at FPR = ',num2str(th(i))]);
    h = legend('LR','FIS','FCM',3);
    set(h,'Interpreter','none')
    hold off
end
fnam = 'AllTPR';
print ('-dtiff', '-r300', fnam);

```

## TPR mapping into FI and plotting

```
clear;
load TPR.mat

dim=size(tpr);
n=dim(1);
th=.1:.1:.9;
nfig=6;
stpr=sort(tpr);

for i=1:dim(3)
    figure(i)
    for j=1:nfig
        SUBPLOT(2,3,j)
        xlim([0 1]);
        xlabel('TPR');
        if (j==1)
            hold on
            scatter(tpr(:,1,i), 1:n, '*r')
            scatter(tpr(:,2,i), 1:n, '.g')
            scatter(tpr(:,3,i), 1:n, 'ob')
            title(['TPR at FPR = ', num2str(th(i))]);
            h = legend('LR', 'FIS', 'FCM', 3);
            set(h, 'Interpreter', 'none')
            hold off
        elseif (j==2) % j==2: Trapezoidal membership function with min & max
            x=[0,0,0;min(tpr(:, :, i));min(tpr(:, :, i));max(tpr(:, :, i));max(tpr(:, :, i));1,1,1];
            y=[0;0;1;1;0;0];
            plotit(x,y,j-1);
        elseif (j==3) % j==3: Triangular membership function with mean
            x=[0,0,0;min(tpr(:, :, i));mean(tpr(:, :, i));max(tpr(:, :, i));1,1,1];
            y=[0;0;1;0;0];
            plotit(x,y,j-1);
        elseif (j==4) % j==4: Triangular membership function with median
            x=[0,0,0;min(tpr(:, :, i));median(tpr(:, :, i));max(tpr(:, :, i));1,1,1];
            y=[0;0;1;0;0];
            plotit(x,y,j-1);
        elseif (j==5) % j==5: Trapezoidal membership function with mean & standard
            deviation
            x=[0,0,0;min(tpr(:, :, i));...
                mean(tpr(:, :, i)) - std(tpr(:, :, i));...
                mean(tpr(:, :, i)) + std(tpr(:, :, i));...
                max(tpr(:, :, i));1,1,1];
            y=[0;0;1;1;0;0];
            plotit(x,y,j-1);
        elseif (j==6) % j==6: Fitting function to envelope
            x=stpr(:, :, i);
            y=[(1:50)*2, (50:-1:1)*2]/100;
            plotit(x,y,j-1);
        end
    end
    fnam = ['fig', num2str(i)];
    print ('-dtiff', '-r300', fnam);
end
close all;
```

## TPR fuzzy numbers defuzzifying

```
clear;
load TPR99.mat;

tpr=tpr99;
R=zeros(size(tpr,2),size(tpr,3),5*2); % Results

% 1. Crisp membership function with min & max
% 1) Center of gravity - Yager's index:
R(:, :, 1) = shiftdim(mean([min(tpr);max(tpr)]),1);
% 2) Mean area - Yager's index 2:
R(:, :, 2) = shiftdim(mean([min(tpr);max(tpr)]*1,1));

% 2. Triangular membership function with mean
% 1) Center of gravity - Yager's index:
R(:, :, 3) = shiftdim(mean([mean(tpr);min(tpr);max(tpr)]),1);
% 2) Mean area - Yager's index 2:
R(:, :, 4) = shiftdim(mean(tpr).*1,1);
```

```

% 3. Triangular membership function with median
% 1) Center of gravity - Yager's index:
R(:, :, 5) = shiftdim(mean([median(tpr);min(tpr);max(tpr)]),1);
% 2) Mean area - Yager's index 2:
R(:, :, 6) = shiftdim(median(tpr).*1,1);

% 4. Trapezoidal membership function with mean & standard deviation
% 1) Center of gravity - Yager's index:
R(:, :, 7) = shiftdim(mean([mean(tpr)+std(tpr);mean(tpr)-std(tpr);min(tpr);max(tpr)]),1);
% 2) Mean area - Yager's index 2:
R(:, :, 8) = shiftdim((mean([mean(tpr)+std(tpr);mean(tpr)-std(tpr)])+
mean([min(tpr);max(tpr)]))).*1/2,1);

% 5. Fitting function to envelope:
% 1) Center of gravity - Yager's index:
R(:, :, 9) = shiftdim(mean(tpr),1);
% 2) Mean area - Yager's index 2:
R(:, :, 10) = shiftdim(median(tpr),1);

save 'Result99.mat' R;

% check the result
clear;
load Result99.mat;
R1=zeros(99,3,10);
for i=1:10
    R1(:, :, i)=R(:, :, i)';
end

```

## TPR fuzzying intervals transferring into AUC fuzzy intervals (1)

### 1. Crisp membership function with min & max

```

close all;
clear;
load TPR99.mat
tpr=tpr99;
dim=size(tpr);
th=0:.01:1;
c1=zeros(length(th),dim(2),dim(3));
c2=zeros(length(th),dim(2),dim(3));

for i=1:dim(3)
    x1=[min(tpr(:, :, i));min(tpr(:, :, i))];
    y1=[0;1];
    x2=[max(tpr(:, :, i));max(tpr(:, :, i))];
    y2=[1;0];
    c1(:, :, i) = interp1(y1,x1,th);
    c2(:, :, i) = interp1(y2,x2,th);
end

d=size(c1);
w=[zeros(d(2),1,d(1)),shiftdim(c1,1),ones(d(2),1,d(1))];
b=[zeros(d(2),1,d(1)),shiftdim(c2,1),ones(d(2),1,d(1))];

% Plot ROC generated by different Alpha-cut in 3D from FPR 0, 0.1, - 0.99,
% 1

figure;
title ('ROC at different alpha-cut');
xlim([0 1]);
ylim([0 1]);
zlim([0 1]);
xlabel('FPR');
ylabel('Alpha');
zlabel('TPR');
grid on
hold on

x=0:.01:1;
s=1:10:101;
for i=s
    y=ones(1,length(x)).*x(i);
    plot3(x,y,w(1,:,i), 'r', 'LineWidth', 2);
    plot3(x,y,w(2,:,i), 'g', 'LineWidth', 2);
    plot3(x,y,w(3,:,i), 'b', 'LineWidth', 2);
end

```

```

plot3(x,y,b(1,:,i),'r','LineWidth',2);
plot3(x,y,b(2,:,i),'g','LineWidth',2);
plot3(x,y,b(3,:,i),'b','LineWidth',2);
%%
for j=1:length(x)
    % plot xgrids
    line(x(j)*[1 1],ylim,'color','k','LineStyle',':','LineWidth',.5);
end
h = legend('LR','FIS','FCM',3);
set(h,'Interpreter','none')
end
hold off
view([45 45]);
print '-dtiff' '-r300' 3DROC(1)

% Plot 9 TPR fuzzy intervals with alpha-cut from 0, 0.1, - 1
figure;

z=0:.1:1;
j=0;
s=11:10:91;
for i=s
    j=j+1;
    SUBPLOT(3,3,j)
    hold on
    x=[0,0,0;min(tpr(:, :, i));min(tpr(:, :, i));max(tpr(:, :, i));max(tpr(:, :, i));1,1,1];
    y=[0;0;1;1;0;0];
    plotit(x,y,i-1);
    title(['TPR Fuzzy Set at FPR = ',num2str(th(i))]);
    xlim([0 1]);
    ylim([0 1]);
    xlabel('TPR');
    ylabel('Alpha-cut');
    % h = legend('LR','FIS','FCM',3);
    % set(h,'Interpreter','none')
    for k=1:length(z)
        % plot ygrids
        line(xlim, z(k)*[1 1],'color','k','LineStyle',':','LineWidth',.2);
    end
    hold off
end
print '-dtiff' '-r300' allTPR(1)

% -----
% Plot AUC fuzzy interval from alpha-cut 0, 0.1, - 1

x=0:.01:1;
wauc=zeros(d(2),d(1));
bauc=zeros(d(2),d(1));
for j=1:d(2);
    for i=1:d(1);
        wauc(j,i)=trapz(x,w(j,:,i));
        bauc(j,i)=trapz(x,b(j,:,i));
    end
end

figure;
y=0:0.01:1;
z=0:0.1:1;
xlim([0 1]);
ylim([0 1]);
title('AUC Fuzzy Interval');
xlabel('AUC');
ylabel('Membership');
hold on
plot(wauc(1,:),y,'r','LineWidth',2);
plot(wauc(2,:),y,'g','LineWidth',2);
plot(wauc(3,:),y,'b','LineWidth',2);
h = legend('LR','FIS','FCM',3);
set(h,'Interpreter','none')
plot(bauc(1,:),y,'r','LineWidth',2);
plot(bauc(2,:),y,'g','LineWidth',2);
plot(bauc(3,:),y,'b','LineWidth',2);
for j=1:length(z)
    % plot ygrids
    line(xlim, z(j)*[1 1],'color','k','LineStyle',':','LineWidth',.5);
end
hold off
print '-dtiff' '-r300' AUCFI(1)

% Calculat AUC fuzzy numbers: Crisp membership function with min & max
a=min(wauc,[],2);

```

```

b=max(bauc,[],2);
% 1) Center of gravity - Yager's index 1:
Yi1 = mean([a,b],2);
% 2) Mean area - Yager's index 2:
Yi2 = mean([a,b],2)*1;

save AUCFI1

```

## TPR fuzzifying intervals transferring into AUC fuzzy intervals (2)

### 2. Triangular membership function with mean

```

close all;
clear;
load TPR99.mat
tpr=tpr99;
dim=size(tpr);
th=0:.01:1;
c1=zeros(length(th),dim(2),dim(3));
c2=zeros(length(th),dim(2),dim(3));

for i=1:dim(3)
x1=[min(tpr(:,:,i));mean(tpr(:,:,i))];
y1=[0;1];
x2=[mean(tpr(:,:,i));max(tpr(:,:,i))];
y2=[1;0];
c1(:,:,i) = interp1(y1,x1,th);
c2(:,:,i) = interp1(y2,x2,th);
end

d=size(c1);
w=[zeros(d(2),1,d(1)),shiftdim(c1,1),ones(d(2),1,d(1))];
b=[zeros(d(2),1,d(1)),shiftdim(c2,1),ones(d(2),1,d(1))];

% Plot ROC generated by different Alpha-cut in 3D from FPR 0, 0.1, - 0.99,
% 1

figure;
title('ROC at different alpha-cut');
xlim([0 1]);
ylim([0 1]);
zlim([0 1]);
xlabel('FPR');
ylabel('Alpha');
zlabel('TPR');
grid on
hold on

x=0:.01:1;
s=1:10:101;
for i=s
y=ones(1,length(x)).*x(i);
plot3(x,y,w(1,:,i),'r','LineWidth',2);
plot3(x,y,w(2,:,i),'g','LineWidth',2);
plot3(x,y,w(3,:,i),'b','LineWidth',2);
plot3(x,y,b(1,:,i),'r','LineWidth',2);
plot3(x,y,b(2,:,i),'g','LineWidth',2);
plot3(x,y,b(3,:,i),'b','LineWidth',2);
% for j=1:length(x)
% % plot xgrids
% line(x(j)*[1 1],ylim,'color','k','LineStyle',':','LineWidth',.5);
% end
h = legend('LR','FIS','FCM',3);
set(h,'Interpreter','none')
end
hold off
view([45 45]);
print '-dtiff' '-r300' '3DROC(2)

% Plot 9 TPR fuzzy intervals with alpha-cut from 0, 0.1, - 1
figure;

z=0:.1:1;
j=0;
s=11:10:91;
for i=s
j=j+1;

```

```

SUBPLOT(3,3,j)
hold on
x=[0,0,0;min(tpr(:, :, i));mean(tpr(:, :, i));max(tpr(:, :, i));1,1,1];
y=[0;0;1;0;0];
plotit(x,y,i-1);
title(['TPR Fuzzy Set at FPR = ', num2str(th(i))]);
xlim([0 1]);
ylim([0 1]);
xlabel('TPR');
ylabel('Alpha-cut');
% h = legend('LR','FIS','FCM',3);
% set(h,'Interpreter','none')
for k=1:length(z)
% plot ygrids
line(xlim, z(k)*[1 1], 'color','k', 'LineStyle',':', 'LineWidth',.2);
end
hold off
end
print '-dtiff' '-r300' allTPR(2)

% -----
% Plot AUC fuzzy interval from alpha-cut 0, 0.1, - 1

x=0:.01:1;
wauc=zeros(d(2),d(1));
bauc=zeros(d(2),d(1));
for j=1:d(2);
for i=1:d(1);
wauc(j,i)=trapz(x,w(j,:,i));
bauc(j,i)=trapz(x,b(j,:,i));
end
end

figure;
y=0:0.01:1;
z=0:0.1:1;
xlim([0 1]);
ylim([0 1]);
title('AUC Fuzzy Interval');
xlabel('AUC');
ylabel('Membership');
hold on
plot(wauc(1,:),y,'r','LineWidth',2);
plot(wauc(2,:),y,'g','LineWidth',2);
plot(wauc(3,:),y,'b','LineWidth',2);
h = legend('LR','FIS','FCM',3);
set(h,'Interpreter','none')
plot(bauc(1,:),y,'r','LineWidth',2);
plot(bauc(2,:),y,'g','LineWidth',2);
plot(bauc(3,:),y,'b','LineWidth',2);
for j=1:length(z)
% plot ygrids
line(xlim, z(j)*[1 1], 'color','k', 'LineStyle',':', 'LineWidth',.5);
end
hold off
print '-dtiff' '-r300' AUCFI(2)

% Calculat AUC fuzzy numbers: Triangular membership function with mean
a=min(wauc,[],2);
b=max(wauc,[],2);
c=max(bauc,[],2);
% 1) Center of gravity - Yager's index 1:
Yi1 = mean([a,b,c],2);
% 2) Mean area - Yager's index 2:
Yi2 = b.*1;

save AUCFI2

```

## TPR fuzzifying intervals transferring into AUC fuzzy intervals (3)

### 3. Triangular membership function with median

```

close all;
clear;
load TPR99.mat
tpr=tpr99;
dim=size(tpr);

```

```

th=0:.01:1;
c1=zeros(length(th),dim(2),dim(3));
c2=zeros(length(th),dim(2),dim(3));

for i=1:dim(3)
x1=[min(tpr(:,:,i));median(tpr(:,:,i))];
y1=[0;1];
x2=[median(tpr(:,:,i));max(tpr(:,:,i))];
y2=[1;0];
c1(:,:,i) = interp1(y1,x1,th);
c2(:,:,i) = interp1(y2,x2,th);
end

d=size(c1);
w=[zeros(d(2),1,d(1)),shiftdim(c1,1),ones(d(2),1,d(1))];
b=[zeros(d(2),1,d(1)),shiftdim(c2,1),ones(d(2),1,d(1))];

% Plot ROC generated by different Alpha-cut in 3D from FPR 0, 0.1, - 0.99,
% 1

figure;
title ('ROC at different alpha-cut');
xlim([0 1]);
ylim([0 1]);
zlim([0 1]);
xlabel('FPR');
ylabel('Alpha');
zlabel('TPR');
grid on
hold on

x=0:.01:1;
s=1:10:101;
for i=s
    y=ones(1,length(x)).*x(i);
    plot3(x,y,w(1,:,i),'r','LineWidth',2);
    plot3(x,y,w(2,:,i),'g','LineWidth',2);
    plot3(x,y,w(3,:,i),'b','LineWidth',2);
    plot3(x,y,b(1,:,i),'r','LineWidth',2);
    plot3(x,y,b(2,:,i),'g','LineWidth',2);
    plot3(x,y,b(3,:,i),'b','LineWidth',2);
    %     for j=1:length(x)
    %         % plot xgrids
    %         line(x(j)*[1 1],ylim,'color','k','LineStyle',':','LineWidth',.5);
    %     end
    h = legend('LR','FIS','FCM',3);
    set(h,'Interpreter','none')
end
hold off
view([45 45]);
print '-dtiff' '-r300' 3DROC(3)

% Plot 9 TPR fuzzy intervals with alpha-cut from 0, 0.1, - 1
figure;

z=0:.1:1;
j=0;
s=11:10:91;
for i=s
    j=j+1;
    SUBPLOT(3,3,j)
    hold on
    x=[0,0,0;min(tpr(:,:,i));mean(tpr(:,:,i));max(tpr(:,:,i));1,1,1];
    y=[0;0;1;0;0];
    plotit(x,y,i-1);
    title (['TPR Fuzzy Set at FPR = ',num2str(th(i))]);
    xlim([0 1]);
    ylim([0 1]);
    xlabel('TPR');
    ylabel('Alpha-cut');
    %     h = legend('LR','FIS','FCM',3);
    %     set(h,'Interpreter','none')
    for k=1:length(z)
        % plot ygrids
        line(xlim, z(k)*[1 1],'color','k','LineStyle',':','LineWidth',.2);
    end
    hold off
end
print '-dtiff' '-r300' allTPR(3)

% -----

```



```

% Plot AUC fuzzy interval from alpha-cut 0, 0.1, - 1
x=0:.01:1;
wauc=zeros(d(2),d(1));
bauc=zeros(d(2),d(1));
for j=1:d(2);
    for i=1:d(1);
        wauc(j,i)=trapz(x,w(j,:,i));
        bauc(j,i)=trapz(x,b(j,:,i));
    end
end

figure;
y=0:0.01:1;
z=0:0.1:1;
xlim([0 1]);
ylim([0 1]);
title('AUC Fuzzy Interval');
xlabel('AUC');
ylabel('Membership');
hold on
plot(wauc(1,:),y,'r','LineWidth',2);
plot(wauc(2,:),y,'g','LineWidth',2);
plot(wauc(3,:),y,'b','LineWidth',2);
h = legend('LR','FIS','FCM',3);
set(h,'Interpreter','none');
plot(bauc(1,:),y,'r','LineWidth',2);
plot(bauc(2,:),y,'g','LineWidth',2);
plot(bauc(3,:),y,'b','LineWidth',2);
for j=1:length(z)
    % plot ygrids
    line(xlim, z(j)*[1 1], 'color','k','LineStyle',':','LineWidth',.5);
end
hold off
print '-dtiff' '-r300' AUCFI(3)

% Calculat AUC fuzzy numbers: Triangular membership function with median
a=min(wauc,[ ],2);
b=max(wauc,[ ],2);
c=max(bauc,[ ],2);
% 1) Center of gravity - Yager's index 1:
Yi1 = mean([a,b,c],2);
% 2) Mean area - Yager's index 2:
Yi2 = b.*1;

save AUCFI3

```

## TPR fuzzifying intervals transferring into AUC fuzzy intervals (4)

### 4. Trapezoidal membership function with mean & standard deviation

```

close all;
clear;
load TPR99.mat
tpr=tpr99;
dim=size(tpr);
th=0:.01:1;
c1=zeros(length(th),dim(2),dim(3));
c2=zeros(length(th),dim(2),dim(3));

for i=1:dim(3)
    x1=[min(tpr(:, :, i));mean(tpr(:, :, i))-std(tpr(:, :, i))];
    y1=[0;1];
    x2=[mean(tpr(:, :, i))+std(tpr(:, :, i));max(tpr(:, :, i))];
    y2=[1;0];
    c1(:, :, i) = interp1(y1,x1,th);
    c2(:, :, i) = interp1(y2,x2,th);
end

d=size(c1);
w=[zeros(d(2),1,d(1)),shiftdim(c1,1),ones(d(2),1,d(1))];
b=[zeros(d(2),1,d(1)),shiftdim(c2,1),ones(d(2),1,d(1))];

% Plot ROC generated by different Alpha-cut in 3D from FPR 0, 0.1, - 0.99,
% 1

figure;

```

```

title ('ROC at different alpha-cut');
xlim([0 1]);
ylim([0 1]);
zlim([0 1]);
xlabel('FPR');
ylabel('Alpha');
zlabel('TPR');
grid on
hold on

x=0:.01:1;
s=1:10:101;
for i=s
    y=ones(1,length(x)).*x(i);
    plot3(x,y,w(1,:,i),'r','LineWidth',2);
    plot3(x,y,w(2,:,i),'g','LineWidth',2);
    plot3(x,y,w(3,:,i),'b','LineWidth',2);
    plot3(x,y,b(1,:,i),'r','LineWidth',2);
    plot3(x,y,b(2,:,i),'g','LineWidth',2);
    plot3(x,y,b(3,:,i),'b','LineWidth',2);
    for j=1:length(x)
        % plot xgrids
        line(x(j)*[1 1],ylim,'color','k','LineStyle',':','LineWidth',.5);
    end
    h = legend('LR','FIS','FCM',3);
    set(h,'Interpreter','none')
end
hold off
view([45 45]);
print '-dtiff' '-r300' 3DROC(4)

% Plot 9 TPR fuzzy intervals with alpha-cut from 0, 0.1, - 1
figure;

z=0:.1:1;
j=0;
s=11:10:91;
for i=s
    j=j+1;
    SUBPLOT(3,3,j)
    hold on
    x=[0,0,0;min(tpr(:, :, i));mean(tpr(:, :, i))- std(tpr(:, :, i));
        mean(tpr(:, :, i))+ std(tpr(:, :, i));max(tpr(:, :, i));1,1,1];
    y=[0;0;1;1;0;0];
    plotit(x,y,i-1);
    title (['TPR Fuzzy Set at FPR = ',num2str(th(i))]);
    xlim([0 1]);
    ylim([0 1]);
    xlabel('TPR');
    ylabel('Alpha-cut');
    % h = legend('LR','FIS','FCM',3);
    % set(h,'Interpreter','none')
    for k=1:length(z)
        % plot ygrids
        line(xlim, z(k)*[1 1],'color','k','LineStyle',':','LineWidth',.2);
    end
    hold off
end
print '-dtiff' '-r300' allTPR(4)

% -----
% Plot AUC fuzzy interval from alpha-cut 0, 0.1, - 1

x=0:.01:1;
wauc=zeros(d(2),d(1));
bauc=zeros(d(2),d(1));
for j=1:d(2);
    for i=1:d(1);
        wauc(j,i)=trapz(x,w(j,:,i));
        bauc(j,i)=trapz(x,b(j,:,i));
    end
end

figure;
y=0:0.01:1;
z=0:0.1:1;
xlim([0 1]);
ylim([0 1]);
title ('AUC Fuzzy Interval');
xlabel('AUC');
ylabel('Membership');

```

```

hold on
plot(wauc(1,:),y,'r','LineWidth',2);
plot(wauc(2,:),y,'g','LineWidth',2);
plot(wauc(3,:),y,'b','LineWidth',2);
h = legend('LR','FIS','FCM',3);
set(h,'Interpreter','none')
plot(bauc(1,:),y,'r','LineWidth',2);
plot(bauc(2,:),y,'g','LineWidth',2);
plot(bauc(3,:),y,'b','LineWidth',2);
for j=1:length(z)
    % plot ygrids
    line(xlim, z(j)*[1 1], 'color','k','LineStyle',':','LineWidth',.5);
end
hold off
print '-dtiff' '-r300' AUCFI(4)

% Calculat AUC fuzzy numbers: Trapezoidal membership function with mean &
% standard deviation
a=min(wauc,[ ],2);
b=max(wauc,[ ],2);
c=min(bauc,[ ],2);
e=max(bauc,[ ],2);
% 1) Center of gravity - Yager's index 1:
Yi1 = mean([a,b,c,e],2);
% 2) Mean area - Yager's index 2:
Yi2 = (mean([a,c],2)+mean([b,e],2))*1/2;

save AUCFI4

```

## TPR fuzzifying intervals transferring into AUC fuzzy intervals (5)

### 5. Fitting function to envelope:

```

close all;
clear;
load TPR99.mat
tpr=tpr99;
dim=size(tpr);
th=0:.01:1;
c1=zeros(length(th),dim(2),dim(3));
c2=zeros(length(th),dim(2),dim(3));

stpr=sort(tpr);
hl=round(length(stpr)/2); % half length
for i=1:dim(3)
    x1=stpr(1:hl,:,i);
    y1=interp1([1;hl],[0;1],1:hl)';
    x2=stpr((hl+1):2*hl,:,i);
    y2=interp1([hl+1;2*hl],[1;0],(hl+1):2*hl)';
    c1(:, :, i) = interp1(y1,x1,th);
    c2(:, :, i) = interp1(y2,x2,th);
end

d=size(c1);
w=[zeros(d(2),1,d(1)),shiftdim(c1,1),ones(d(2),1,d(1))];
b=[zeros(d(2),1,d(1)),shiftdim(c2,1),ones(d(2),1,d(1))];

% Plot ROC generated by different Alpha-cut in 3D from FPR 0, 0.1, - 0.99,
% 1

figure;
title('ROC at different alpha-cut');
xlim([0 1]);
ylim([0 1]);
zlim([0 1]);
xlabel('FPR');
ylabel('Alpha');
zlabel('TPR');
grid on
hold on

x=0:.01:1;
s=1:10:101;
for i=s
    y=ones(1,length(x)).*x(i);
    plot3(x,y,w(1,:,i),'r','LineWidth',2);
    plot3(x,y,w(2,:,i),'g','LineWidth',2);
end

```

```

plot3(x,y,w(3,:,i), 'b', 'LineWidth', 2);
plot3(x,y,b(1,:,i), 'r', 'LineWidth', 2);
plot3(x,y,b(2,:,i), 'g', 'LineWidth', 2);
plot3(x,y,b(3,:,i), 'b', 'LineWidth', 2);
%%
for j=1:length(x)
    % plot xgrids
    line(x(j)*[1 1],ylim,'color','k','LineStyle',':','LineWidth',.5);
end
h = legend('LR','FIS','FCM',3);
set(h,'Interpreter','none')
end
hold off
view([45 45]);
print '-dtiff' '-r300' 3DROC(5)

% Plot 9 TPR fuzzy intervals with alpha-cut from 0, 0.1, - 1
figure;

z=0:.1:1;
j=0;
s=11:10:91;
for i=s
    j=j+1;
    SUBPLOT(3,3,j)
    hold on
    x=stpr(:,i);
    y=[(1:50)*2,(50:-1:1)*2]/100;
    plotit(x,y,i-1);
    title (['TPR Fuzzy Set at FPR = ',num2str(th(i))]);
    xlim([0 1]);
    ylim([0 1]);
    xlabel('TPR');
    ylabel('Alpha-cut');
    % h = legend('LR','FIS','FCM',3);
    % set(h,'Interpreter','none')
    for k=1:length(z)
        % plot ygrids
        line(xlim, z(k)*[1 1],'color','k','LineStyle',':','LineWidth',.2);
    end
    hold off
end
print '-dtiff' '-r300' allTPR(5)

% -----
% Plot AUC fuzzy interval from alpha-cut 0, 0.1, - 1

x=0:.01:1;
wauc=zeros(d(2),d(1));
bauc=zeros(d(2),d(1));
for j=1:d(2);
    for i=1:d(1);
        wauc(j,i)=trapz(x,w(j,:,i));
        bauc(j,i)=trapz(x,b(j,:,i));
    end
end
end

figure;
y=0:0.01:1;
z=0:0.1:1;
xlim([0 1]);
ylim([0 1]);
title ('AUC Fuzzy Interval');
xlabel('AUC');
ylabel('Membership');
hold on
plot(wauc(1,:),y,'r','LineWidth',2);
plot(wauc(2,:),y,'g','LineWidth',2);
plot(wauc(3,:),y,'b','LineWidth',2);
h = legend('LR','FIS','FCM',3);
set(h,'Interpreter','none')
plot(bauc(1,:),y,'r','LineWidth',2);
plot(bauc(2,:),y,'g','LineWidth',2);
plot(bauc(3,:),y,'b','LineWidth',2);
for j=1:length(z)
    % plot ygrids
    line(xlim, z(j)*[1 1],'color','k','LineStyle',':','LineWidth',.5);
end
hold off
print '-dtiff' '-r300' AUCFI(5)

% Calculat AUC fuzzy numbers: Fitting function to envelope

```

```
a=max(wauc,[],2);  
b=min(bauc,[],2);  
% 1) Center of gravity - Yager's index 1:  
Yi1 = mean([wauc,bauc],2);  
% 2) Mean area - Yager's index 2:  
Yi2 = mean([a,b],2)*1;  
  
save AUCFI5
```

*Published with MATLAB® 7.6*

## **A.2 The function of obtaining TPRs and FPRs for different classifiers**

```

function [auc,TFPR] = mymtfpr(Y,X,Method)
% Function mymtfpr to be added to the MATLAB vocabulary returns:
% the AUC value, true positive rates and false positive rates.
%
% Y: a column vector of N subjects belonging to 0/1 binary classes.
% X: N by M of M predicotrs.
%
% Method is an optional integer, where
% 1: Multinomial logistic regression (default)
% 2: FIS - Single-output Sugeno-type FIS
% 3: FIS - Fuzzy C-means clustering
%
% (c) Bei Wen, May 2009, Erasmus University

% Check arguments
disp(nargchk(2, 3, nargin));           % Allow 2 to 3 inputs
disp(nargoutchk(0, 2, nargout));      % Allow 0 to 2 outputs
mthd=1;                                % Default
if nargin == 3
    mthd=Method;
end

% Remove missings
dt=[Y,X];
dt(any(isnan(dt),2),:)=[];
Y=dt(:,1);
X=dt(:,2:end);

% preprocess
if(min(Y)==0)
    Y=Y+1;
end
if(min(X)==0)
    X=X+1;
end

n = size(Y);
y = accumarray((1:n)' Y,1);
Y = y(:,end);
if mthd==1                                % Method #1: LR
    B = mnrfit(X,y);
    PHAT = mnrvl(B,X);
    PHAT = PHAT(:,2);
elseif mthd==2                            % Method #2: FIS
    fis0 = genfis1([X,Y],2,'gbellmf','constant');
    fis1 = anfis([X,Y],fis0,[10 0 0.01 0.9 1.1], [1 1 1 1],[X,Y] ,1);
    PHAT = evalfis(X,fis1);
    PHAT(PHAT>1) = 1;
    PHAT(PHAT<0) = 0;
elseif mthd==3                            % Method #3: FCM
    fis0 = genfis3(X,Y);
    PHAT = evalfis(X,fis0);
    PHAT(PHAT>1) = 1;
    PHAT(PHAT<0) = 0;
end

[tpr1,fpr1] = roc(y(:,2)',PHAT');
[B,I] = unique(fpr1);
fpr2 = 0:0.001:1;
tpr2 = interp1(fpr1(I),tpr1(I),fpr2);
TFPR = [fpr2',tpr2'];
TFPR(isnan(TFPR)) = 1;
auc = trapz(TFPR(:,1),TFPR(:,2));

```

## **A.3 The function of plotting figures**



```
function plotit(x,y,n)
xlim([0 1]);
ylim([0 1]);
hold on;
line(x(:,1),y,'Color','r','LineWidth',2,'LineStyle','-');
line(x(:,2),y,'Color','g','LineWidth',2,'LineStyle','-');
line(x(:,3),y,'Color','b','LineWidth',2,'LineStyle','-');
hold off;
title(['Fuzzy Interval ', num2str(n)]);
xlabel('TPR');
ylabel('Membership');
```

*Published with MATLAB® 7.6*