# Evaluation of Fuzzy Rules Extracted from Data

Adam Schenker[1], Mark Last[2], and Abraham Kandel[1]

**Abstract**

A general methodology for evaluation of fuzzy rules extracted from data is presented. Though the primary goal of most data mining systems is high classification or prediction accuracy, the user may be interested in rules which are not necessarily the most accurate. Our approach provides an alternative measure of rule validity, based on methods of fuzzy set theory. When the rules to be tested come from a human expert, the method can be viewed as a verification-based data mining method. If the rules are generated by another (discovery-based) data mining method (such as a decision-tree algorithm), the method can be seen as a post-processing step in the KDD process, aimed at evaluating the extracted rules. The method involves four major steps: hypothesis formulation, data selection, hypothesis testing, and decision. In the hypothesis formulation step, a set of fuzzy rules are created using conjunctive antecedents and consequent functions. In the data selection step, a subset of all data in the database is chosen as a sample set. This sample should contain enough records to be representative of the data to a certain degree of user satisfaction. In hypothesis testing, a fuzzy implication is applied to the selected data for each extracted rule and the results are combined using some aggregation function. These results are used in the final step to evaluate the validity of each rule. The presented technique is applied to the rules generated by the C4.5 algorithm from two sample databases. The experimental results demonstrate potential benefits of using validity-based evaluation of rules.

**Keywords.** Fuzzy hypothesis testing, fuzzy rule evaluation, data mining, approximate reasoning, sampling.

---

[1] Department of Computer Science and Engineering, University of South Florida, 4202 E. Fowler Avenue, ENB 118, Tampa, FL 33620, USA

[2] Department of Information Systems Engineering, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel

# 1 INTRODUCTION

In this paper, the concept of *fuzzy testing*, or more specifically, *fuzzy hypothesis testing* is presented as a method for evaluating rules that are extracted or induced by data. A fuzzy hypothesis test is used to determine the truth (or falsity) of a proposed hypothesis. The hypothesis may involve either crisp or fuzzy data; however, a fuzzy hypothesis test should produce a value in [0,1], which indicates the degree to which the hypothesis is valid for given sample data. This is an extension of the classical hypothesis test, which yields a value in {0,1}; i.e., the hypothesis under consideration (the null hypothesis $H_0$) is either accepted or rejected. The fuzzy hypothesis test will accept $H_0$ to some degree $\mu$ and $H_1$ to some degree $1-\mu$. Basic concepts of fuzzy theory can be found in [Klir95, Zadeh65].

We show how such hypothesis tests can be applied to the problem of data mining. *Data mining* refers to the automatic or semiautomatic extraction of useful information from a large set of data (i.e. a database [Elmasri94, Petry96]). Data mining techniques encompass several different methods for extracting information such as clustering, neural networks, and statistical methods [Mitchell97]. In this paper, we consider the case where the data mining method is *verification-based*; i.e., a hypothesis is presented and the data mining tool responds by saying whether or not the data in the database supports the hypothesis. This is different from *discovery-based* data mining methods, where the primary purpose is to discover previously hidden knowledge.

This type of verification can also be used as a post-processing step in the KDD process (the interpretation step in [Fayyad96b]). The rules to be verified can come from either a rule induction algorithm or a human expert. When our method is applied to a set of extracted rules, it can be used to evaluate the validity of each rule. This can lead to a reduction in the size of the rule base by eliminating the most irrelevant rules. When the method is used for interaction with a human expert, it can be used to perform exploratory hypothesis testing.

The reasoning employed in the "crisp" (statistical) hypothesis testing resembles the common judicial practice: a person is assumed innocent until proven guilty. According to the statistical theory (see [Mendenhall93]), *not rejecting* the null hypothesis should not be interpreted as accepting that hypothesis. "Not rejecting" just means that we do not have sufficient statistical evidence to refute the null hypothesis. On the other hand, rejecting the null hypothesis implies that there are an infinite number of alternative hypotheses, one of them being true.

Unlike statistical methods, which are aimed at testing one hypothesis at a time, machine learning methods are concerned with a search in a hypothesis space, defined by some underlying representation (e.g., decision trees or

artificial neural networks). However, most machine learning algorithms produce a single (possibly complex) hypothesis (e.g., a specific decision tree or a rule set), chosen by pre-determined criteria. In other words, the resulting hypothesis is always accepted with the degree of 1.00.

In this paper, a fuzzy hypothesis testing procedure based on fuzzy set theory, rather than strictly on probability theory, is developed. This is a departure from other fuzzy hypothesis testing methods, such as [Casals94], which are fuzzy extensions of classical statistical techniques (e.g. Bayes). As noted in [Glymour96], statistical-based hypothesis testing has certain limitations. Specifically, hypotheses that are good approximations can be rejected with sufficiently large samples. The fuzzy set theoretic-based hypothesis testing presented here overcomes this problem by allowing partial (approximate) matches to the hypothesis conditions. As discussed in [Rocke98], statistical methods and data mining methods have different characteristics, but similar goals. For statistics, data can be expensive to collect and the goal is to perform analysis of sparse data sets. With data mining, computational efficiency is more important and availability of data is not a problem. In his conclusion, Rocke states that statistical methods applied to data mining must not depend on any prior knowledge of the structure of the data. This is the case with our method, as we make no prior assumptions about probability distributions of the data.

There are many techniques described in the literature for discovering and evaluating fuzzy rules from raw data (without using the output of any other data mining method). In [Au99], the authors describe a genetic algorithm-based method for discovering fuzzy associations between linguistic terms. The fitness function (i.e. the "goodness" of a rule) used to evaluate the set of rules is based on a probabilistic approach. In [Slawinski99] a hybrid evolutionary approach is used. Here, the fitness of a rule is related to its relevance (a rule is considered relevant if the constrained probability of its conclusion exceeds the unconstrained probability). [Imamura99] use fuzzy neural networks trained on typical and exceptional data to extract fuzzy rules. For this method, the set of rules is considered good if, after training and tuning, the model produces a low prediction error. In [Kim99] a fuzzy decision tree approach is presented. Their method includes generation of fuzzy membership functions from histogram analysis. Like in "crisp" decision tree algorithms (e.g., C4.5), rules are extracted from paths connecting the root node to terminal nodes. Some more fuzzy-based methods for automating the human perception of raw data are described in [Last99a,b] and [Last01].

Most discovery-oriented methods mentioned above do not deal with interpreting rules extracted by another (possibly "crisp") rule induction algorithm (an application of fuzzy logic to post-processing the results of a data mining algorithm is presented in [Last01]). In this paper, we are applying a verification-based fuzzy approach to post-processing a set of "crisp" rules produced by a

"classical" decision-tree algorithm (C4.5). As noted in [Pedrycz98], fuzzy techniques have natural links to data mining. Specifically, it is desired that the user be able to interact with the data mining process in order to focus on certain areas of interest, since most of the patterns generated by the discovery methods are of no interest to the user [Silberschatz96]. This user interaction should be performed in a linguistic manner, and we can use fuzzy sets to represent the subjective "interestingness" of the results (validity, generality, usefulness, etc.).

Classification accuracy is the most common performance measure of data mining algorithms (decision trees, neural networks, etc.), but it is also used for measuring the interestingness of individual rules produced by these algorithms. However, the most accurate rule is not necessarily the most interesting to the user. Our method evaluates rules by their *validity*, rather than by their classification accuracy. The validity measure provides an alternative evaluation and scoring of the extracted rules *without modifying the underlying classification model*.

The difference between accuracy and the proposed validity measure is subtle, but important. Accuracy is a *statistical* measure. It measures the probability that a randomly chosen record satisfies the rule. For example, for a rule with 90% accuracy, 9 out of 10 records will match the rule. Our measure is a *fuzzy* measure. It measures the membership of a rule in the fuzzy set of valid rules. A rule with validity of 0.9 is a good rule, but there is no probabilistic information about the rule. Consider the following example to understand the difference between probability and fuzziness. A man is lost in the desert. In the distance, he sees two wells, but he only has the strength to walk to one of them. One well has a 90% probability of having drinkable (healthy) water. The other has water with a fuzzy membership of 0.9 in the fuzzy set of healthy water. The probabilistic information is of little use to this person: it just means that if he drinks water from 10 wells, one of them is expected to be poisonous and the man will die. However, if he goes to the second well, he knows that he will not die, but the water quality is not quite the best. As we see from this example, it is sometimes advantageous to rely on fuzzy measures, rather than on statistical measures, because, in certain cases, the fuzziness provides information that is more helpful for users.

The paper is organized as follows. The formal notation used in the paper is presented in Section 2. In Section 3, the fuzzy testing procedure is explained. Applications of the fuzzy testing procedure for evaluating discovered rules are described in Section 4. Conclusions are given in Section 5.

## 2 FORMAL NOTATION

Let us now define the formal notation that will be used in the paper. A set of collected data, i.e. a database, is defined:

$$X = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_m\}$$

where $\mathbf{x}_i$ is an *n*-dimensional vector in an *n*-dimensional feature space:

$$\mathbf{x}_i = \langle x_{i,1}, x_{i,2}, \ldots, x_{i,n} \rangle$$

A set $D \subseteq X$ is chosen, called a sample set, which will be used to test the hypothesis. Next, we have a set of hypotheses $H=\{H_0, H_1, \ldots H_f\}$ where $H_0$ is the null hypothesis to accept or reject and $H_1$ through $H_f$ are the alternative hypotheses we must accept if we reject $H_0$. A hypothesis can be thought of as an implication of the form:

if *condition*$_1$ and *condition*$_2$ and ... *condition*$_k$
then $\mathbf{x}_i$ is a member of *F* with membership $\mu(\mathbf{x}_i)$

In other words, a hypothesis is composed of a set C of *k* conjunctive antecedent conditions and a consequent classification (e.g. cluster, fuzzy set) *F*. A condition is a comparison of one of the components of $\mathbf{x}_i$ and a constant (possibly fuzzy) value. $\mu$ is defined as a mapping:

$$\mu(\mathbf{x}_i, H_a) \rightarrow [0,1]$$

We should note that $\mu$ above comes from the standard fuzzy terminology of a membership function; it is not the statistical notation of the mean. The value of $\mu$ determines whether the data collected agrees with the hypothesis. As a shorthand notation, we write $\mu(\mathbf{x}_i, H_a)$ as $\mu_a$. A value of $\mu_0=1$ means the data is in total agreement with the null hypothesis; a value of $\mu_0=0$ means the data totally contradicts the null hypothesis. Additionally, the value of $\mu$ for the alternative hypotheses should be the inverse of that of $H_0$, i.e. $\mu_1+\mu_2+\ldots\mu_f=1-\mu_0$.

Now that the formal terminology is in hand, several problems must be addressed to achieve the goal of fuzzy testing. First, it is not always possible to use the entire set of observations to test the hypothesis. If this is the case, a sample set D, which is a subset of the total observations X, must be chosen. However, the sample must be an accurate representation of the overall database. Second, there should be some criteria for selecting whether to use a crisp or fuzzy condition. Also, note that the hypotheses are restricted to having conjunctive (ANDed) conditions only. A hypothesis with disjunctive conditions can be tested by separating out the disjunctive parts and treating each as a separate

hypothesis; if one hypothesis is accepted, then all can be accepted. Third, a mapping function $\mu$ must be created for each hypothesis. This function should be maximal (i.e., equal to 1) when all conditions are met for that hypothesis. It should be minimal (i.e., equal to 0) when no conditions are met. Fourth, once $\mu_0$ and $\mu_1, \mu_2, \ldots, \mu_f$ have been computed, we must decide which hypothesis should prevail. In general, when $\mu_0 \leq \mu_i$, we should reject $H_0$ and accept $H_i$, where $1 \leq i \leq f$ and $\mu_i = \max(\mu)$. When $\mu_0$ is significantly greater than $\mu_i$ we should accept $H_0$ and reject $H_i$ ($1 \leq i \leq f$). However, we must decide what "significantly greater" means. We will address these issues and describe the fuzzy hypothesis testing procedure in more detail in the next section.

## 3 FUZZY HYPOTHESIS TESTING

In this section, the major problem areas described in the previous section are addressed, and the general fuzzy hypothesis test procedure is given.

### 3.1 Obtaining a "good" sample

Since it may not always be practical or possible to use all collected data (i.e. the entire database), a sampling of data, called a *sample set*, is used to verify the hypotheses. The sample set D is usually chosen at random from among the set X (the database). This random sampling must be large enough to make sure that the set D is "good"; i.e. that D reflects the contents of X. If D = X it must be accepted; the sample is the entire database. If D = $\varnothing$, it must be rejected; the sample contains no data. Otherwise, the number of records in D, denoted $d = |D|$, will determine if a sample is rejected or accepted. Using some basic heuristics, we can define a DoS (*Degree of Satisfaction*) function (in general, a non-linear function) that can tell us how good the sample is. In fact, we are actually creating a fuzzy set with the conceptual meaning "good sample." Given the sample size $d$, a membership function $f(d)$ of the fuzzy set "good sample" can reflect how "good" the sample is if it satisfies the following conditions:

1. boundary conditions: $f(0)=0$; $f(|X|)=1$
2. monotonically increasing: for all $0 \leq a < b \leq |X|$, $f(a) \leq f(b)$
3. monotonically decreasing derivatives: for all $0 \leq a < b \leq |X|$, $f'(a) \geq f'(b)$

The first condition (boundary condition) tells us that a sample with no items is totally unacceptable and a sample that comprises the entire database is totally acceptable. We can also define a non-zero number of records as the minimum

acceptable sample size (e.g., ten records). The second condition states that adding a new item to the sample always increases the "goodness" of the sample. The final condition is related to the heuristic concept of a sample; adding the same number of items to a small sample increases the acceptability of the sample more than adding the same number of items to a sample with a lot of items. For example, if we raise the size of a sample set from 20 to 40 items, it should cause a greater increase in the quality of the sample than adding 20 items to a sample set of 1000 items. Clearly, many functions can satisfy these conditions; the selection can be tuned to the specific application based on expert knowledge. In this paper, we use a function based on the logarithm, called the Degree of Satisfaction, to represent the belief that D is a good sample of X based on *d* (the sample set size) and *m* (the size of the entire database). Here the value *m* is used to scale *d* so the input is in the range [0,1]. We choose the logarithm function since it easily satisfies the conditions above; recall that *log(x)* is strictly monotonically increasing and its derivative, $x^{-1}$, is strictly monotonically decreasing. With a minor modification, we can ensure the proper boundary conditions. The function is shown below:

$$f_b(d) = \begin{cases} \dfrac{\log(\frac{d}{m})}{\log(b)} + 1 & \text{when } d > \dfrac{m}{b} \\ 0 & \text{otherwise} \end{cases}$$

where $b>0$ is a constant that controls the behavior (as discussed below) and x-intercept of the function (the minimum acceptable sample size). Larger values of *b* make the x-intercept closer to 0. For example, when *b=10*, the x-intercept is at 10% of *m*; for *b=100*, the x-intercept is 1% of *m*. In general, the value of *b* will be determined according to the user perception of the minimum sample size required for providing meaningful results.

As can be seen from the above expression, the fuzzy approach to calculating the sample size is not based on any assumptions about the data itself. This approach closely agrees with the common process of knowledge discovery in databases [Brachman96, Fayyad96a-e], where the target data set is selected before the data mining stage. It also agrees with the conclusions of [Rocke98] that data mining should not depend on any prior knowledge of the data structure.

The procedure for selecting an appropriate sample size, suggested by the statisticians [Mendenhall93], is based on setting the desired power of test and choosing in advance a hypothesis about the underlying distribution of the data (including the parameters of that hypothesis). The power is defined as the probability of a test to detect an *important* effect size $\delta$. However, statistical techniques do not provide any objective way of measuring importance under the assumption that the effect size can be determined from expert knowledge. This

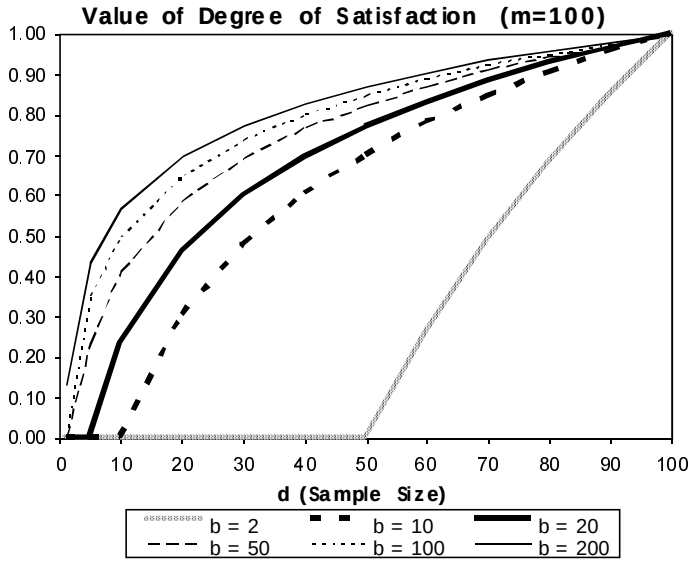


Figure 1. The Degree of Satisfaction as a Function of the Sample Size for Various Values of *b*

is similar to the fuzzy membership function, shown above, which represents the user attitude towards the importance of having a minimum number of examples.

After showing that both measures of the sample quality (the statistical and the fuzzy one) depend on some subjective decisions made by users, we are comparing the behavior of these measures as a function of the sample size and the user-dependent parameters ($b$ or $\delta$) in Figure 1 and Figure 2. As can be seen from Figure 2, the power of test increases rapidly for small sample sizes (the first 10%) and then the function becomes almost flat. Thus, there is almost no difference between the power of test for 30% sample and 100% sample, which means that the power of test seems to favor smaller samples, regardless of the value of $\delta$. With the fuzzy function, the change is gradual throughout, but still more pronounced for lower sample sizes. In addition, for small values of $\delta$ and finite sample sizes, the maximum achievable power of test is much less than one. In the fuzzy model, we fix the high end of the function at 1.0 for 100% sample size. The $b$ parameter allows us to adjust the linear behavior of the function. For a small values of $b$, it behaves almost as a straight line. For larger values ($b >> m$), the function becomes more non-linear.

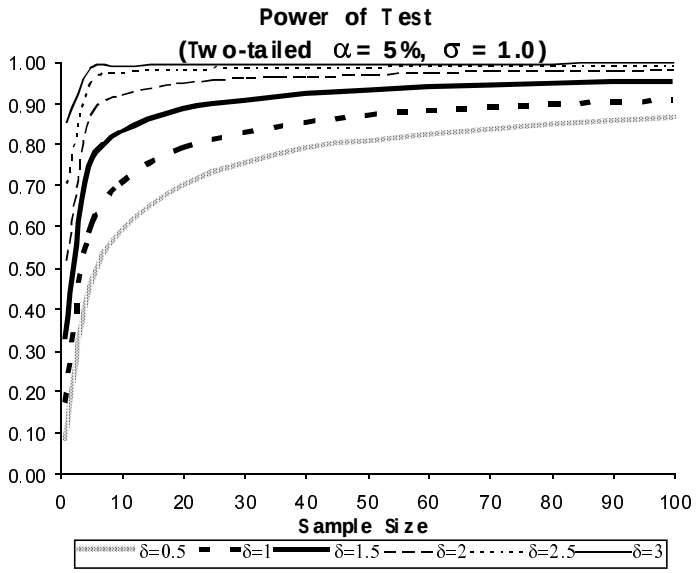


**Figure 2. The Power of Test as a Function of the Sample Size for Various Values of the Effect (δ)**

The overall conclusion is that the fuzzy DoS function is more intuitive and practical than the power of test function for the following reasons:

1. The function is always normalized to [0,1] range (equal to 0 for an empty sample and 1.0 for a complete sample).
2. The function can be set to zero for insufficient, small samples (by changing the value of *b*).
3. The "goodness" of sample size varies gradually.
4. The function is easily adjusted to linear and non-linear behavior.

### 3.2 Creating the mapping function

The mapping function $\mu_i$ maps each vector (record) in D, for a given hypothesis $H_i$, to a value in [0,1]. This number represents the degree to which each vector agrees with the hypothesis. In order to determine the agreement, the membership function of the consequent $F_i$ must be known. If the data described by the vector **x** lies within $F_i$, (viewing $F$ as a fuzzy cluster) then $\mu_i$

should equal the degree of membership of **x** in $F_i$. Usually $F_i$ will be some geometric function on [0,1], such as a triangular or trapezoidal shaped function.

The vectors in D are compared with the conjunctive conditions in the antecedent of the hypothesis. For crisp conditions, any condition(s) which are false cause **x** to be excluded from consideration since they do not lend any support to the null hypothesis or alternative hypotheses. For fuzzy conditions, it may be necessary to use some threshold value to determine if the vector **x** should be excluded. For example, for a fuzzy value of 0.5 or less, the vector **x** may be closer to some other fuzzy set. Similarly, we may wish to use a threshold when examining the consequent. Each fuzzy condition in the antecedent will have a value on [0,1] for each **x**, and these values must be combined using a t-norm operation, such as *min*. The resulting value indicates the degree to which **x** supports the antecedent conditions of H. A fuzzy implication is then performed for the combined antecedent values and the consequent value:

$$\mu_I = \max(1\text{-}P_k f_l)$$

where P is the value of the combined antecedents and *f* is a function describing the fuzzy membership of the consequent. Here the subscript *l* denotes to which hypothesis each variable belongs; it will range from 0 (the null hypothesis) to *k*, for *k* alternative hypotheses. We are using here the Kleene-Diennes implication, but other implications are possible. The selection of a fuzzy implication will depend on the application. [Klir95] The Kleene-Diennes implication has the feature that when a record strongly matches the antecedent of a rule but the consequent does not strongly match (incorrect classification), the rule is penalized. If a record strongly matches the consequent, it is rewarded regardless of the matching of the antecedent. We use threshold values (see above) to ignore data which do not match the rule strongly enough, thus eliminating the case where the matching of the antecedent is weak and the consequent is strong. Records that partially match both antecedent and consequent are partially rewarded.

Once the membership $\mu_0$ for each record **x** in the sample D is determined, the values must be aggregated to determine if the values in D, taken as a whole, support $H_0$. This can be done in a variety of ways including arithmetic mean (each point contributes to the decision), minimum (pessimistic – if any **x** fail $H_0$, then $H_0$ is rejected), or maximum (optimistic – if any **x** pass $H_0$, then $H_0$ is accepted). For arithmetic mean, we denote the overall validity $M_k$ for hypothesis *k*:

$$M_k(D) = \frac{\sum_{i \in D} \mu_k(x_i)}{d}$$

When we perform fuzzy hypothesis testing on crisp rules (recall that crisp sets are just a special subset of fuzzy sets) and use the arithmetic average to generate the overall matching, $M_k(D)$ becomes identical to the accuracy of the rule in the traditional sense.

The method presented above is similar to that of approximate reasoning or the use of fuzzy inference in control systems [Klir95]. In those methods, we compute the degree to which each rule antecedent in the rule base matches the given data and then aggregate the corresponding rule consequents. Here we also add thresholds for both the antecedent and consequent. The threshold on the antecedent allows us to in effect remove weakly matching rules from consideration. The threshold on the consequent penalizes rules whose antecedents match strongly but whose consequent matching is weak. Thus, we penalize rules that have strong evidence but which support the wrong conclusion. In control systems, we usually defuzzify the result of the aggregation to a single value and select a control action based on the defuzzification. With hypothesis testing, we have two actions we can take: accept or reject. The decision of which action to take is discussed in the next sub-section.

### 3.3 Making the decision to accept or reject $H_0$

Once the validity $M_k(D)$ is computed, the decision must be made to accept or reject $H_0$. For $I$ alternative hypotheses, numbered 1 to $I$, a decision to accept $H_0$ requires that:

1. $M_0(D) \geq M_i(D)$ for all $I > 0$ and
2. $M_0(D)$ must be "significantly greater than" $max(M_i(D))$

"Significantly greater than" is a fuzzy term which may be defined in a number of ways. It must satisfy certain conditions, which are similar to the Degree of Satisfaction function:

1. boundary conditions: if $a \leq b$, $\mu_{sg}(a,b)=0$; $\mu_{sg}(1,0)=1$
2. monotonicity: let $c>0$ be a constant, then $\mu_{sg}(a,c) \leq \mu_{sg}(a',c)$ for all $a'>a$

For example, it can be defined as:

$$\mu_{sg}(\mu_0, \mu_i) = \begin{cases} \sqrt{\mu_0 - \mu_i} & \text{when } \mu_0 > \mu_i \\ 0 & \text{otherwise} \end{cases}$$

In the case where there is only one alternative hypothesis $\mu_1 = 1-\mu_0$, $\mu_{sg}$ can be rewritten as:

$$\mu_{sg}(\mu_0, \mu_1) = \begin{cases} \sqrt{2\mu_0 - 1} & \text{when } \mu_0 > \mu_1 \\ 0 & \text{otherwise} \end{cases}$$

## 4 APPLICATIONS OF FUZZY TESTING

In this section, we present examples of using the fuzzy hypothesis testing as a post-processing step in the KDD process. In order to examine the performance of the fuzzy hypothesis test on a given data set, we are performing the following procedure. First, Quinlan's C4.5 algorithm [Quinlan93] is applied to the data. This algorithm creates a set of crisp classification rules based on the decision tree approach (including the estimated accuracy of each rule). The crisp nature of C4.5 rules requires discretization of continuous features. Consequently, each record either supports a given rule (if its continuous features lie in the appropriate intervals), or not.

For the rules discovered by C4.5, we define appropriate fuzzy sets, based on the distribution of data, and formulate fuzzy versions of the rules. Then the fuzzy hypothesis test procedure is applied to calculate the validity of these rules. We should emphasize that the fuzzy hypothesis test itself is not being used for classification or prediction. The goal of fuzzy hypothesis testing is to provide an alternative evaluation of the rules, which come from a data mining algorithm (in this case C4.5). The set of rules produced by C4.5 can be used for classification or prediction disregarding the interestingness of each individual rule.

In sub-sections 4.1 and 4.2 below, we describe the results of applying C4.5 to two data sets and then evaluating the generated rules with the fuzzy hypothesis test. In Section 4.3, we present the discussion of the results.

### 4.1 CPU performance data set

The CPU performance data set, available from [Merz96], contains 209 records relating to various computer systems, such as their cache size and memory capacity. This data was initially presented and analyzed in [Ein-Dor87] and the actual numbers reflect the state of the computing technology in the early 1980's. The goal of this data set is to predict the performance of each system based on the other attributes. Since C4.5 requires a discrete target attribute, we discretized the continuous performance attribute used by [Ein-Dor87] to the following intervals, which have approximately equal numbers of cases:

**Table 1 – Results of C4.5 and fuzzy hypothesis testing for CPU database**

| Rule# | C4.5 Rule | Accuracy | Fuzzy Rule | Validity | $\mu_{sg}$ |
|---|---|---|---|---|---|
| **13** | if vendor = formation then performance poor | 75.8% | if vendor is formation, then performance is poor | 0.97 | 0.97 |
| **29** | if vendor = bti then performance poor | 50.0% | if vendor is bti, then performance is poor | 0.81 | 0.79 |
| **9** | if max main memory <= 6300 and cache size <= 30 then performance poor | 71.5% | if max main memory is small and cache is small, then performance is poor | 0.74 | 0.69 |
| **10** | if vendor = dec and max main memory > 6300 then performance medium | 70.7% | if vendor is dec and max main memory is not small, then performance is medium | 0.74 | 0.69 |
| 15 | if vendor = harris then performance medium | 82.0% | if vendor is harris, then performance is medium | 0.59 | 0.42 |
| 24 | if vendor = magnuson and max main memory <= 10480 then performance poor | 63.0% | if vendor is magnuson and max memory is not large, then performance is poor | 0.41 | 0.0 |
| 35 | if cache size > 30 and max channel > 7 then performance good | 87.3% | if cache is large and max channel is not small, then performance is good | 0.40 | 0.0 |
| 34 | if cache size > 30 and max channel <= 7 then performance medium | 75.8% | if cache is large and max channel is not large, then performance is medium | 0.30 | 0.0 |
| 33 | if max main memory > 16000 then performance good | 95.9% | if max main memory is large then performance is good | 0.29 | 0.0 |
| 1 | if vendor = apollo, then performance medium | 50.0% | if vendor is apollo, then performance is medium | 0.0 | 0.0 |

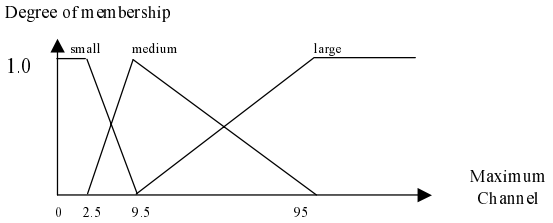


**Figure 3. Fuzzy Sets for the CPU Performance Data Set**

poor = (0, 35]
medium = (35, 90]
good = (90, 1150]

C4.5 produced a decision tree with 10 rules for this data set. For each of these rules, we created fuzzy rules, which interpret the semantic meaning of the

C4.5 rules. The fuzzy sets used for the rules were determined solely from the distribution of the data, to give each fuzzy set approximately equal coverage of the universe of discourse (similar to the approach of [Kim99]). The fuzzy sets (membership functions) for this database are shown in Figure 3.

Next, we perform the fuzzy hypothesis testing on the rules using the given data with Degree of Satisfaction set to 0.95 and $b$=10. This yields a sample size of 187 records. We use thresholds of 0.5 for both antecedent and consequent (see Section 3.2). In Table 1, we show the crisp rules generated by C4.5, their accuracy, the fuzzified versions of these rules, and the results of performing the fuzzy hypothesis test on the fuzzy rules. The rules are sorted in descending order of validity (result of fuzzy hypothesis testing). $\mu_{sg}$ is the degree to which the hypothesis is significantly greater than the alternative hypothesis, as defined in Section 3.3. Rule numbers in bold indicate those hypotheses that we accept (rules 9, 10, 13 and 29).

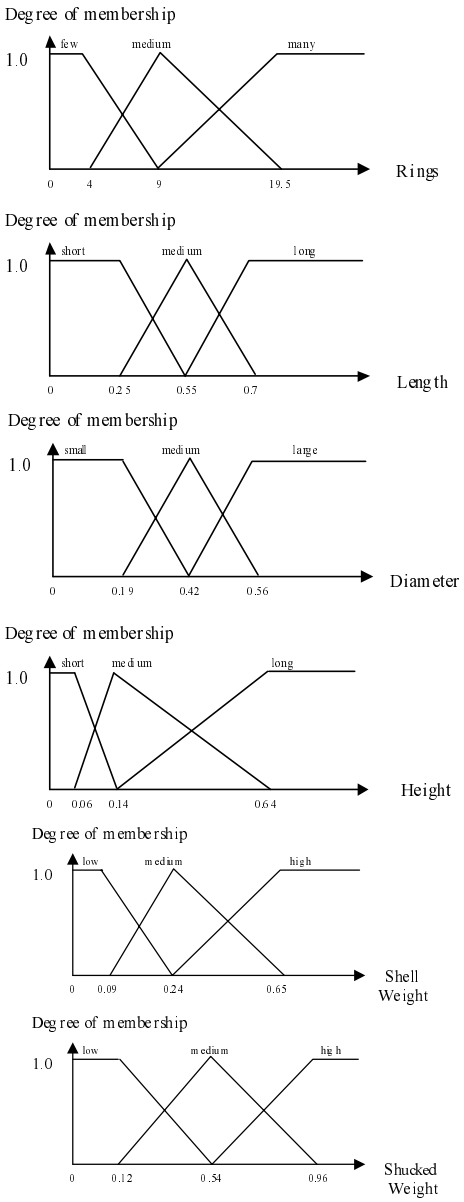


**Figure 4. Fuzzy Sets for the Abalone Data Set**

**Table 2 – Results of C4.5 and fuzzy hypothesis testing for abalone database**

| Rule # | C4.5 Rule | Accuracy | Fuzzy Rule | Validity | $\mu_{sg}$ |
|---|---|---|---|---|---|
| **18** | if shucked weight > 0.4455 and shell weight > 0.177 and shell weight <= 0.3595 then medium rings | 55.8% | if shucked weight is high and shell weight is medium, then medium rings | 0.83 | 0.81 |
| **2** | if sex = M and height > 0.12 and shell weight <= 0.177 then medium rings | 38.7% | if sex is M and height is not short and shell weight is low, then medium rings | 0.77 | 0.73 |
| **9** | if shucked weight > 0.306 and shell weight > 0.177 and shell weight <= 0.29 then medium rings | 49.1% | if shucked weight is not low and shell weight is medium, then medium rings | 0.75 | 0.71 |
| 11 | if length <= 0.565 and shucked weight <= 0.447 and shell weight > 0.2485 then many rings | 67.0% | if length is not long and shucked weight is not high and shell weight is high, then many rings | 0.45 | 0.0 |
| 14 | if shucked weight <= 0.4455 and shell weight > 0.29 then many rings | 75.4% | if shucked weight is not high and shell weight is high, then many rings | 0.43 | 0.0 |
| 1 | if height <= 0.12 and shell weight <= 0.177 then few rings | 77.2% | if shell weight is low and height is short, then few rings | 0.37 | 0.0 |
| 5 | if sex = I and shell weight <= 0.177 then few rings | 83.4% | if sex is I and shell weight is low, then few rings | 0.30 | 0.0 |
| 19 | if shell weight > 0.3595 then many rings | 71.4% | if shell weight is high, then many rings | 0.23 | 0.0 |
| 7 | if sex = F and shucked weight <= 0.306 and shell weight > 0.177 then many rings | 66.8% | if sex is F and shucked weight is not high and shell weight is not low, then many rings | 0.12 | 0.0 |
| 6 | if sex = M and shucked weight <= 0.306 and shell weight > 0.177 then many rings | 67.0% | if sex is M and shucked weight is not high and shell weight is not low, then many rings | 0.10 | 0.0 |
| 20 | if sex = I and shucked weight > 0.4455 and shell weight > 0.29 then many rings | 66.7% | if sex is I and shell weight is high and shucked weight is high, then many rings | 0.0 | 0.0 |
| 17 | if diameter > 0.48 and height > 0.17 and shell weight > 0.177 then many rings | 68.4% | if diameter is large and height is long and shell weight is not low, then many rings | 0.0 | 0.0 |

### 4.2 Abalone data set

The abalone data set, which is also available from [Merz96], contains 4177 records about abalone. The goal of this data set is to predict the number of rings an abalone has, which is related to its age, using attributes such as length, diameter and various weights. We discretized the continuous rings attribute by the following intervals of approximately equal frequency:

few = (0, 8]
medium = (8, 10]
many = (10, 29]

C4.5 produced 12 rules for this data set. To reduce the number and complexity of the rules, we require 2 branches with 100 or more cases and used a pruning confidence level of 35%. Like the CPU database, we created fuzzy rules which interpret the meaning of the C4.5 rules. The fuzzy sets (membership functions) for this database are shown in Figure 4.

Next, we perform the fuzzy hypothesis testing on the rules with Degree of Satisfaction set to 0.95 and $b$=10 (the same as the CPU database). This yields a sample size of 3723 records. We use thresholds of 0.5 for both antecedent and consequent. In Table 2 we show the rules generated by C4.5, their accuracy, the corresponding fuzzy rules and the results of performing the fuzzy hypothesis test on the fuzzy rules. The rules are sorted in descending order of validity. Rule numbers in bold indicate hypotheses we accept (rules 2, 9 and 18).

### 4.3 Discussion of results

It is important to understand what the results presented mean and what the benefits of using fuzzy hypothesis testing are. The accuracy of C4.5 for each rule represents the estimated ("pessimistic") accuracy of the rule when applied to unseen (validation) cases [Quinlan93]. With fuzzy hypothesis testing, $M_0(D)$ represents the overall validity of the rule, taking into account partial matching of fuzzy antecedents and consequents.

We see from the results that the accuracy (a statistical concept) of the crisp rule is not necessarily related to the validity (a fuzzy concept) of the fuzzy rule ($M_0(D)$): the most valid rules may be not the most accurate ones and vice versa. The difference comes from how the fuzzy sets representing the antecedents and consequent are defined. Each fuzzy condition can differ from its crisp counterpart in two ways. First, we could be considering either fewer or more records than the crisp case (a difference in the support of the set). Second, with fuzziness, there will be records that partially match several fuzzy rules, rather than the strict matching of a single crisp rule (a difference in degree of membership).

As we discussed in the introduction, accuracy, which is a statistical measure, and validity, which is a fuzzy measure, involve different kinds of information. Consider, for example, a rule from the CPU database (No. 33) with high accuracy but low validity:

if maximum main memory is large, then performance is good

The high accuracy of the C4.5 rule tells us that, of the computers in the database with a maximum memory capacity above 16 MB, most have performance between 90 and 1,150. The accuracy measure is "crisp": it completely ignores the computers with memory below 16 MB and the distribution of the performance index within the interval (90, 1150]. These "crisp" intervals have been determined to optimize the classification performance of the rule, but people tend to think in the linguistic "fuzzy" terms and most of them will interpret the same rule as "if maximum main memory is large, then performance is good." However, the low validity measure indicates that the above linguistic rule may not be so valid. This is possible for two reasons. First, the original rule applies to computers with more than 16MB of maximum memory capacity, but there is no distinction between different amounts of memory within that range. If there is truly a relation between memory size and performance, computers with more memory should have a correspondingly higher performance than those with less. However, there are instances where computers with 32MB of memory have worse performance than those with 24MB. Our method reflects this fact, but the crisp case does not. Second, even if most large memory computers do have performance between 90 and 1,150, their performance still may be close to the low boundary of the range (90). Both these facts will be completely ignored by the statistical approach of C4.5. While the data mining algorithm may be able to find crisp thresholds that "optimize" the model classification accuracy of the model, there is generally no single value where an abrupt change actually occurs in the real world system; the change is usually gradual. With fuzzy rules, we can capture the idea of a gradual change. Thus, rule No. 33 may be misused, based on its accuracy, leading to costly mistakes by potential computer buyers.

For the abalone database, we have the following rule (No. 9) with low accuracy but high validity:

if shucked weight is not low and shell weight is medium, then there are a medium number of rings

This rule states that if the weight of the abalone meat is not low and the weight of its shell is medium, then it is of medium age (a medium number of rings). This provides the user with an important observation that there is a correlation between the abalone's weight, and its age, with younger abalone being smaller. However, the accuracy of this rule is low, meaning that for the "crisp" intervals of the predicting attributes (shucked and shell weight), only about 50% of abalones have 9 or 10 rings. In this case, some information about

the impact of abalone's weight on its age may be lost, if the accuracy is the only measure used. With the fuzzy validity measure, we take into account the fact that the linguistic terms, such as "medium shell weight" or "medium number of rings," do not have sharp boundaries. Using crisp ranges, a slight change in the value of weight such as 0.05 can cause a record to match (or not match) a condition of a rule. With fuzziness, we gradually reduce the membership of a value as it moves away from the typical values and thus can take into account more records from the database. For the rule above, the crisp version was applicable to only 471 records, which matched the conditions exactly. For the fuzzy hypothesis test, we were able to use 2082 records. Thus, the fuzzy hypothesis test considered more records which were also relevant to the rule, but which had lower membership grades (less than the 1.0 of the crisp case).

As mentioned in [Pedrycz98], for each rule, we would like to evaluate various (fuzzy) performance measures, such as generality, usefulness, validity, novelty, and simplicity. As we have seen in the above examples, high accuracy is not necessarily related to the user interpretation of the rule and we should develop alternate forms of rule evaluation in order to account for these differences. Our method can be viewed as a general methodology of defining a fuzzy validity measure for the rules produced by a KDD process.

## 5 CONCLUSION

In this paper, a general procedure for using the fuzzy hypothesis test as a post processing method for rule inducing data mining algorithms was presented. This method provides an alternative measure (validity) of rules produced by a KDD process. In the first step of the method, the size of the sample set is determined by defining a function, which we introduced, called the Degree of Satisfaction; in other words, a fuzzy set with the meaning "good sample." This function does not depend on any assumptions about the structure of the data, as in statistical methods. The fuzzy hypothesis test determines the validity (a value on [0,1]) of the hypothesis (i.e. rule) on a sample set of data. This value is then used to determine if the hypothesis can be accepted or if it must be rejected and an alternative hypothesis accepted. This gives us a general method for testing the "goodness" of fuzzy rules; the procedure can be tailored to an application by selection of aggregation and fuzzy implication operations. When using the arithmetic mean to combine results from individual rules and using only crisp rules (recall that crisp sets are a subset of fuzzy sets) the usual classification accuracy percentage results.

We have selected two databases and used the C4.5 algorithm to generate crisp rules from them. These rules have been fuzzified by selecting

appropriate fuzzy sets based on attribute distributions to create fuzzy rules with a similar semantic meaning. We then applied the fuzzy hypothesis testing to each fuzzy rule and identified the most valid rules in the ruleset. The results show that the rule scoring, based on fuzzy validity, is significantly different from the scoring based on prediction accuracy. The main reason for that is that the "crisp" approach ignores certain information, which is hidden in the raw data. In other words, depending on the preferences of the user and the application details, the most accurate rules are not necessarily the most valid and interesting ones.

The knowledge discovery process is based on many subjective decisions and its results are evaluated by humans, who are rarely objective in their judgment. Data mining algorithms are often compared by their classification accuracy, and one can always find the most accurate method for a given problem. However, it is much more difficult to find an algorithm which sorts out the most *important* rules for a given user. Consequently, there is no objective way to compare the quality of various evaluation measures (e.g., accuracy vs. validity). Still, the availability of alternative measures, such as the one presented in this paper, is important for increasing the usefulness of the discovered knowledge.

Future work includes applying the fuzzy approach to other problems of hypothesis testing, like regression and correlation, estimation of probability distributions, analysis of variance, etc. Fuzzy hypothesis testing on time series data is another interesting topic. One can also consider developing similar methodologies for other fuzzy measures, related to generality and interestingness of rules.

## ACKNOWLEDGEMENTS

This work was partially supported by the USF Center for Software Testing under grant no. 2108-004-00.

## REFERENCES

W.-H. Au and K. C. C. Chan. [1999]. "FARM: A Data Mining System for Discovering Fuzzy Association Rules". *IEEE International Fuzzy System Conference Proceedings*.

R. J. Brachman *et. al.* [1996]. "Mining Business Databases". *Communications of the ACM*, Vol. 39, No. 11.

M. Casals and P. Gil. [1994]. “Bayesian Sequential Test for Fuzzy Parametric Hypotheses from Fuzzy Information”. *Information Sciences*. Vol. 80, No. 3-4.

P. Ein-Dor and J. Feldmesser. [1987]. “Attributes of the Performance of Central Processing Units: a Relative Performance Prediction Model”. *Communications ACM*. Vol. 30.

R. Elmasri and S. B. Navathe. [1994]. *Fundamentals of Database Systems*. Redwood City, CA: Benjamin/Cummings Publishing Company.

U. Fayyad. [1996a]. “Data Mining and Knowledge Discovery: Making Sense Out of Data”. *IEEE Expert*.

U. Fayyad and R. Uthurusamy. [1996b]. “Data Mining and Knowledge Discovery in Databases”. *Communications of the ACM*, Vol. 39, No. 11.

U. Fayyad *et. al.* [1996c]. “The KDD Process for Extracting Useful Knowledge from Volumes of Data”. *Communications of the ACM*. Vol. 39, No. 11.

U. Fayyad *et. al.* [1996d]. “Mining Scientific Data”. *Communications of the ACM*. Vol. 39, No. 11.

U. Fayyad *et. al.* (Eds.). [1996e]. *Advances in Knowledge Discovery and Data Mining*. Cambridge, MA: AAAI Press/MIT Press.

C. Glymour *et. al.* [1996]. “Statistical Inference and Data Mining”. *Communications of the ACM*. Vol. 39, No. 11.

K. Imamura, *et. al.*. [1999]. “Extraction of Typical and Exceptional Fuzzy Rules from Data Including Qualitative and Quantitative Attributes”. *IEEE International Fuzzy System Conference Proceedings*.

M. W. Kim, *et. al.*. [1999]. “Efficient Fuzzy Rule Generation Based on Fuzzy Decision Tree for Data Mining”. *IEEE International Fuzzy System Conference Proceedings*.

G. J. Klir and B. Yuan. [1995]. *Fuzzy Sets and Fuzzy Logic*. Upper Saddle River, NJ: Prentice Hall.

M. Last and A. Kandel. [1999a]. “Automated Perceptions in Data Mining”. 1999 *IEEE International Fuzzy Systems Conference Proceedings, Part I*. Seoul, Korea.

M. Last, A. Schenker, and A. Kandel. [1999b]. “Applying Fuzzy Hypothesis Testing to Medical Data”. *New Directions in Rough Sets, Data Mining, and Granular-Soft Computing 7th International Workshop ’99*.

M. Last and A. Kandel. [2001]. “Fuzzification and Reduction of Information-Theoretic Rule Sets”. *Data Mining and Computational Intelligence*. A. Kandel, H. Bunke, and M. Last (Eds), Physica-Verlag, Studies in Fuzziness and Soft Computing. Vol. 68.

W. Mendenhall, *et al.* [1993]. *Statistics for Management and Economics*. Belmont, CA: Duxbury Press.

C. J. Merz *et. al.* [1996]. “UCI Repository of Machine Learning Databases”. http://www.ics.uci.edu/~mlearn/MLRepository.html. Irvine, CA: University of California, Department of Information and Computer Science.

T. M. Mitchell. [1997]. *Machine Learning*. WCB/McGraw-Hill.

W. Pedrycz. [1998]. “Fuzzy Set Technology In Knowledge Discovery”. *Fuzzy Sets and Systems*. Vol. 98.

F. E. Petry. [1996] *Fuzzy Databases: Principles and Applications*. Boston, MA: Kluwer Academic Publishers.

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

D. M. Rocke. [1998]. “A Perspective on Statistical Tools for Data Mining Applications”. *Proceedings of the Second International Conference on the Practical Application of Knowledge Discovery and Data Mining.*

A. Silberschatz and A. Tuzhilin. [1996]. “What Makes Patterns Interesting in Knowledge Discovery Systems.” *IEEE Transactions on Knowledge and Data Engineering*. Vol. 8, No. 6.

T. Slawinski, *et. al.* [1999]. “A Hybrid Evolutionary Search Concept for Data-based Generation of Relevant Fuzzy Rules in High Dimensional Spaces”. *IEEE International Fuzzy System Conference Proceedings*.

L. A. Zadeh. [1965]. “Fuzzy Sets.” *Information and Control*. Vol. 8.