Fuzzy Decision Tree using Soft Discretization and a Genetic Algorithm based Feature Selection Method

Min Chen and Simone A. Ludwig
Department of Computer Science
North Dakota State University
Fargo, ND, USA
min.chen@my.ndsu.edu, simone.ludwig@ndsu.edu

***Abstract*—In data mining, decision tree learning is an approach that uses a decision tree as a predictive model mapping observations to conclusions. The fuzzy extension of decision tree learning adopts the definition of soft discretization. Many studies have shown that decision tree learning can benefit from the soft discretization method leading to improved predictive accuracy. This paper implements a Fuzzy Decision Tree (FDT) classifier that is based on soft discretization by identifying the best "cut-point". The selection of important features of a data set is a very important preprocessing task in order to obtain higher accuracy of the classifier as well as to speed up the learning task. Therefore, we are applying a feature selection method that is based on the ideas of mutual information and genetic algorithms. The performance evaluation conducted has shown that our FDT classifier obtains in some cases higher values than other decision tree and fuzzy decision tree approaches based on measures such as true positive rate, false positive rate, precision and area under the curve.**

***Keywords*-Soft discretization, fuzzy decision tree, genetic algorithm**

I. Introduction

Data mining and knowledge discovery have been attracting a significant amount of researchers recently. Data is being collected and accumulated across all fields with dramatic speed. Efficient tools or techniques that can assist humans to extract useful information or knowledge from the rapidly expanding volume of data are required.

The data mining tasks can be classified as unsupervised or supervised learning. Unsupervised learning focuses on finding patterns describing the data that can be interpreted. Supervised learning involves using some features or fields of the data set to predict unknown or future values of interest. Discretization plays an important role in data mining and knowledge discovery. If attributes are continuous, discretization can be used to transform them into discrete features. Unsupervised and supervised discretization depends on whether it contains class information or not [1]. Unsupervised discretization does not consider the class information while supervised discretization does. Discrete values are intervals in a continuous spectrum of values. Discrete values are closer to a knowledge-level representation than continuous values, since they are more concise to represent and specify, easier to use and comprehend [1]. Due to the tremendous expansion in the volume of data being generated and stored, many studies show that induction tasks can benefit from discretization by leading to improved predictive accuracy in particular in the area of rule mining. Rules with discrete values are shorter and more comprehensible than continuous values.

Decision Tree (DT) mining is one of the frequently used classification methods that specify the sequences of decisions that need to be made accompanied by the resulting recommendation. DT mining typically uses a top-down strategy, and the measure of information gain was introduced as a "goodness" criterion. DTs are intrinsic multi-class learners that scale comparatively well, sometimes even outperforming other state-of-art methods especially when they are used as part of an ensemble method [2], [3]. DTs are comprehensible and interpretable and can handle different types of attribute (e.g., numerical and categorical) [4]. Popular methods of decision trees are ID3 [5], C4.5 [6] and CART [7], which generate a tree structure through recursively partitioning the attribute space until the whole decision space is completely partitioned into a set of non-overlapping subspaces [1], which is also called hard discretization. Soft discretization on the other hand is when the decision space is partitioned into a set of overlapping subspaces. The classical crisp discretization can cause low classification accuracy since it can not analyze noisy data using crisp cut points. Furthermore, crisp discretization can lead to misclassification of new objects, which are close to the separating boundary between decision classes [8].

Researchers have attempted to combine some elements of symbolic and sub-symbolic approaches to decision tree induction. The fuzzy approach is one of such extensions. Due to its ability of handling vagueness, ambiguity and reduction of complexity, fuzzy logic [9],[10] has been widely applied in dealing with problems of uncertainty, noise, and inexact data. A DT induction method using fuzzy set theory, in other words, Fuzzy Decision Tree (FDT), is becoming an increasingly popular method to solve classification problems.

FDT, like classical DT, uses the top-down strategy. In order to find the best so called "cut-point", FDT is based on soft discetization and follows the DT run recursively on each partition until the best cut point is found.

However, the data contains many redundant or irrelevant features. These features provide no useful information in any context. In order to improve the model interpretability and enhance the generalization, a Genetic Algorithm (GA) based feature selector is applied in this paper. Mutual information is one suitable criterion for feature selection [11]. Mutual information can reduce the uncertainty about the class labels and minimize a lower bound on the Bayes classification error as investigated in [12]. Nevertheless, the estimation of mutual information is not an easy task. Mutual information is a nonlinear measure used to quantify not only linear and but also nonlinear correlations [13]. The challenge of using mutual information for feature selection is the estimation of this measure from the available data.

The contribution of this paper is arranged as follows. Section II describes related work. The proposed approach is introduced in Section III. The experimental setup and results are demonstrated in Section IV. Finally, conclusions and future work are discussed in Section V.

II. Related Work

Related work with regards to the classification task in the area of data mining include neural networks, naive Bayes classification, decision tree, genetic algorithm, etc. [14]. Neural networks have become equally popular to decision trees due to its relative ease of application and abilities to provide gradual improvements [15]. Neural networks are seen as data driven self-adaptive methods, which can adjust themselves to the data without any explicit specification of the underlying model [16]. However, neural networks lack similar levels of comprehensibility as decision trees, which is a problem when users want to understand or justify the decisions [15].

Naive Bayes learning is one particular strategy belonging to the category of learning methods. It is a statistical method for classification, which is based on applying the Bayes' theorem with the naive independence assumption [17]. Naive Bayes learning has been deployed in numerous classification tasks due to its simplicity, effectiveness and incremental training ability. Naive Bayes classifiers have widespread deployment in medical diagnosis [18], email filtering [19], and recommender systems [20], [21]. Due to the independence assumption of naive Bayes, a large amount of research has been conducted on relaxing the naive Bayes independence assumption in machine learning. However, learning the tree structure is not trivial especially in the area of text classification [22].

With respect to fuzzy decision trees applied to classification tasks, fuzzy decision trees have been applied in the medical and financial fields [15], and have been used for ranking tasks [4], etc. Fuzzy decision tree induction follows the same steps as that of when building a classical decision tree. [23] proposed a novel criterion on measurement of cognitive uncertainty, and [24] proposed an alternative criterion based on fuzzy mutual entropy in the possibility domain.

Related work related to feature selection has shown that many search approaches have been proposed. [25] aggressively reduce the document vocabulary in a naive Bayes model and a decision tree approach using an information measure. A Normalized Mutual Information Feature Selection (NMIFS) [26] is proposed as a measure of redundancy among features. Two feature evaluation metrics for the naive Bayes classifier have been applied on multi-class text data sets in [27]. Three new approaches to fuzzy-rough feature selection based on fuzzy similarity relation have been proposed in [28] to provide robust solutions and advanced tools for data analysis.

In general, feature selection can improve the scalability, efficiency and accuracy of classifiers. Therefore, the proposed FDT with GA feature selection is investigated. In addition, the combination of feature selection and FDT is of great interest. Based on the complementary nature of feature selection and FDT, it hybridizes the underlying concepts to deal with aspects of data imperfection and improve predictive classification accuracy.

III. Fuzzy Decision Tree Classifier

The main difference between classical DT and FDT is using crisp or soft discretization respectively. The classical DT uses crisp discretization while fuzzy decision tree is based on soft discretization. The decision space is partitioned into a set of non-overlapping subspaces using the crisp discretization method. For soft discretization, the decision space is partitioned into a set of overlapping subspaces. For both classical and fuzzy decision trees, each path from the root node to a leaf node represents a classification rule. In a more explicit form, the i^{th} branch has the following form:

$$\text{IF } x_{i1} \in A_1^m \text{ AND .. AND } x_{ij} \in A_j^n \text{ THEN } c_i \in C_i^k$$

where x_{ij} denotes the j^{th} attribute of the i^{th} branch. A_j^m denotes the m^{th} antecedent value of the j^{th} attribute. c_i is the consequent of the i^{th} rule.

The fuzzy decision tree has been extended in the possibility domain based on fuzzy set theory [26]. A fuzzy set F is characterized by a membership function $F(a) : U \rightarrow [0, 1]$. $F(a)$ is the membership degree of F taking a value $a \in U$. Let $V = \{F_1, F_2, ..., F_m\}$ be a family of fuzzy sets of U. Then

$$\sum_{i=1}^{m} F_i(a) = 1, \forall a \in U \tag{1}$$

The cut-point is determined by the fuzzy set pair A_1 and A_2 such that $A_1(a) + A_2(a) = 1$. The fuzzy class entropy

in a data set S is:

$$E(S) = \sum_{j=1}^{k} p(c_j, S) \log p(c_j, S) \quad (2)$$

where $p(c_j, S) = \sum_{a_i \in c_j} (A_1(a_i) + A_2(a_i))$ is the proportion of records in S that belongs to class c_j. After soft discretization, the set S is partitioned into two subsets S_1 and S_2 given a threshold value. The class information entropy is calculated by the probability of fuzzy partition as:

$$E(S) = \frac{N^{S_1}}{N^S} E(S_1) + \frac{N^{S_2}}{N^S} E(S_2) \quad (3)$$

$$E(S_i) = -\sum_{j=1}^{k} p(c_j, S_i) \log p(c_j, S_i), i = 1, 2 \quad (4)$$

$$p(c_j, S_i) = \frac{N^{S_i c_j}}{N^{S_i}}, i = 1, 2 \quad (5)$$

where $N^S = \sum_{n=1}^{|S|} \sum_{i=1}^{2} A_i(a_n)$, $N^{S_i} = \sum_{n=1}^{|S|} A_i(a_n)$, $i = 1, 2$.

A fuzzy discretization process mainly includes four phases (shown in Figure 1): sorting, evaluation, splitting and stopping. Since we are also considering feature selection as a preprocessing step, it is the step to be performed before the other four phases are started.

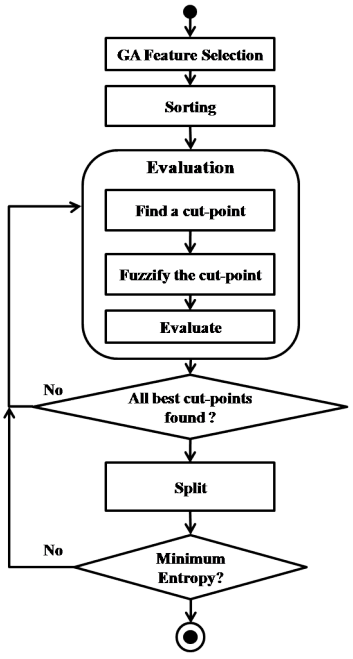

Figure 1. A fuzzy discretization process.

1) Preprocessing Phase: Feature selection is a common technique in data mining in order to reduce the overall feature set that is provided to the algorithm choosing the most important features to be used for the training of the classifier. However, not only does the reduction of features contribute to a faster learning process, but it usually also improves the classification accuracy. For the feature selection task, methods from information theory are frequently used. Feature selection involves the maximization of the mutual information between features and the class label. However, this procedure is very computationally expensive since the joint entropy has to be calculated requiring the estimation of the joint probability distributions. In order to reduce the computational complexity, a variable selection based on the principle of minimum-redundancy/maximum-relevance, which maximizes the mutual information indirectly was proposed in [11]. However, since all possible combinations of variables need to be checked, there is still a large computation involved, thus, a simple method of incremental search, that obtains sub-optimal solutions has been proposed by previous work [29]. The use of a genetic algorithm was proposed to address the combinatorial checking of the variables, which our FTD classifier has adopted.

Algorithm 1 shows the steps involved in the feature selection process. The inputs are the number of features of the data set, feature vector, and class vector. The output is a vector of selected features. The first steps of the algorithm are the calculation of the entropy of each feature vector and the class vector, as well as the calculation of the mutual information between the feature and class vectors and between the features. Once these values are calculated the GA process can start by setting up a population of randomly initialized chromosomes. The first generation can begin. While iterating over the population, the maximum relevance, minimum redundancy and fitness value are calculated for each chromosome. Afterwards the population is ranked, crossover is performed, and repeated features and features with an entropy of 0 are removed, and another generation is started. This process proceeds until the maximum number of generations is reached. The feature vector found is the one used for the next steps in our proposed FDT approach.

Algorithm 1 GA-based Feature Selection Method

```
Input: number of features
Input: feature vector
Input: class vector
Output: selected feature vector
gen_max: maximum number of generations
N_pop: population size
calculate entropy of each feature
calculate output entropy
calculate mutual information between feature and output
calculate mutual information between features
random initialization of population
for gen = 1 : gen_max do
    for index = 1 : N_pop do
        calculate maximum relevance
        calculate minimum redundancy
        calculate fitness by subtracting max. relevance from min. redundancy
    end for
    rank population according to their fitness
    perform crossover
    remove repeated features and features with entropy = 0
end for
```

2) Sorting Phase: The continuous values of a feature are sorted in either ascending or descending order. This task can be computationally expensive if care is not taken when considering the sorting algorithm. Quick-sort is one efficient sorting algorithm, which has a time complexity of $O(NlogN)$ [30]. Specifically, assume current nodes contain a data set S with N records, the records are sorted according to the value of A generating a sequence of ordered values $a_1, a_2, ..., a_N$.

3) Evaluation Phase: The next step after sorting is to find the best "cut-point", which can split a range of continuous values into two parts. A list of candidate cut points $T = (a_i + a_{i+1})/2$ are generated using the class boundary points. By using the fuzzy set pair, A_1 and A_2, the cut points can be fuzzified to generate candidate soft discretizations. In the proposed algorithm, the evaluation function used to evaluate each candidate soft discretization is as given by Equation 3.

4) Splitting Phase: The intervals are split in a top-down strategy, which requires to evaluate "cut-points". In order to choose the best one and split the range of continuous values into two partitions, the algorithm runs recursively for each part until a stopping criterion is satisfied.

5) Stopping Phase: A stopping criterion specifies when the discretization process is stopped. Specifically, a threshold value $\theta \in [0.1, 0.2]$ is predefined. If the truth level of a branch $\frac{N^{S_i}}{N^S}$ is greater than θ, then the truth level of the branch belonging to the j^{th} class is calculated as follows:

$$\delta_{i,j} = \frac{\sum_{a_k \in c_j} A_i(a_k)}{N^{S_i}}, i = 1, 2 \quad (6)$$

Otherwise, the corresponding branch is deleted. Another predefined maximum value of δ called $\mu \in [0.8, 0.9]$ is used as the stopping criterion. If the maximum δ value is greater than μ, the corresponding branch search is terminated as a leaf. This leaf is assigned as the class c_j. Otherwise, the data set S is partitioned into S_1 and S_2 until the above criterion (either $\frac{N^{S_i}}{N^S} \geq \theta$ or $\delta \geq \mu$) are satisfied.

Generally, a FDT classifier starts by sorting the continuous values of a feature. It then generates a possible candidate "cut-point", and fuzzifies the "cut-point". It uses an entropy evaluation function to check whether the candidate's "cut-point" is satisfied or not. It recursively keeps checking until the best "cut-point" is found, and repeats to generate the soft discretization for the other attributes. When all attributes have been soft discretized, the attribute of minimum value is selected to generate two child branches and nodes. This process repeats until the stopping criterion is met.

IV. Experiments and Results

In order to investigate the performance of our FDT approach, experiments are conducted comparing the effect of using all features of five chosen data sets, or using the preprocessing step that reduces the feature set with the GA-based feature selection method as described earlier. The experimental setup is described in the following subsection followed by the experimental results.

A. Experimental Setup

The experiments of all algorithms are conducted on a number of data sets taken from the UCI repository [31]. The experiments of FDT are run on an ASUS desktop (Intel(R) Dual Core I3 CPU @3.07 GHz, 3.07 GHz) with Java Version 1.6.0.25. A few data mining algorithms are used for comparison provided by the Weka software (version 3.7.8). All experiments use the 10-fold cross validation [16] technique. Each data set is divided into 10 partitions. Nine partitions of the data set are used as training data and one partition is selected as test data.

B. Experimental Results

In order to compare our FDT classifier, two DT classifiers J48 and REPTree, and a fuzzy rule classification algorithm FURIA were chosen. The algorithms are summarized as follows:

- **J48** is a decision tree implementation induced by the C4.5 algorithm, which is developed by Quinlan [6]. It learns decision trees for the given data by constructing them in a top-down way.
- **REPTree** stands for Reduced Error Pruning Tree [16], which is a fast decision tree implementation that builds a decision tree using information gain as the splitting criterion. It adopts a reduced-error pruning using top-down strategy. It uses the C4.5 method to deal with missing values and only sorts values of numeric attributes once.
- **FURIA** is short for Fuzzy Unordered Rule Induction Algorithm, which extends the well-known RIPPER algorithm [32]. FURIA learns unordered fuzzy rule sets instead of rule lists. It includes a number of modifications and extensions to deal with uncovered examples.

Parameter	Values
Population size	$200 \times$ # of selected features
Maximum iteration	80
Selection	Elitism
Crossover rate	1

Table I
GA PARAMETERS OF GA-BASED FEATURE SELECTION METHOD.

Table I shows the parameters and their values used for FDT with the GA-based feature selection. For the proposed algorithm, the population size is chosen as the product of 200 and the number of selected features, and the maximum number of iterations is set to 80. An elitist selection strategy is selected and the crossover rate is set to 1.

The description of the selected data sets used are summarized in terms of number of attributes, number of instances

and number of classes as shown in Table II. The 5 data sets are listed alphabetically. The values in brackets under the column *Features* is the reduced number of features after the preprocessing process is applied.

Data Set	Features	Instances	Classes
Diabetes	8 (4)	768	2
Glass	10 (8)	214	7
Ionosphere	34 (18)	351	2
Pendigits	16 (9)	10992	10
Vehicle	18 (7)	946	4

Table II
DATASETS USED FOR EXPERIMENTS.

Measured are the weighted average True Positive Rate (FPR) and the False Positive Rate (TPR), as well as the precision. Experiments were run using the data sets as listed above on all algorithms, first without the feature selection stage meaning that all features were used, and the second time using the reduced feature set as determined by the GA-based feature selection method. All results reported in the tables are reported by a number indicating all features were used from the data sets, and the second value in brackets are results when the algorithms were run with the reduced feature set. The values in bold are the best values comparing the results for with/without the GA-based feature selection method.

In Table III, the average weighted true positive rates of all algorithms are measured. As shown in the table, FURIA and FDT using soft discretization always score better than the classical DT techniques, J48 and REPTree that use hard discretization. In addition, FDT with/without the GA-based feature selection method scores slightly better than FURIA on most data sets except for data set Pendigits.

Data Set	J48	REPTree	FURIA	FDT
Diabetes	73.8 (*74.9*)	75.3 (*72.8*)	74.5 (*75.1*)	**76.2** (***76.8***)
Glass	66.8 (*71.5*)	66.4 (*69.2*)	**70.6** (*74.8*)	**70.6** (***75.1***)
Ionosphere	91.5 (*91.7*)	89.5 (*90.6*)	91.2 (*89.5*)	**91.6** (***91.9***)
Pendigits	96.6 (*93.9*)	95.6 (*92.9*)	**98.0** (***94.5***)	96.8 (*93.4*)
Vehicle	72.5 (*68.3*)	72.3 (*66.7*)	70.6 (*70.2*)	**72.6** (***72.1***)

Table III
AVERAGE WEIGHTED TPR (%) OF ALL ALGORITHMS.

The average weighted false positive rate is tabulated in Table IV. FURIA and FDT achieve better results (smaller values) than J48 and REPTree. Furthermore, FDT scores slightly better than FURIA for most data sets except Pendigits.

With respect to the average weighted precision, FURIA and FDT obtain better results than J48 and REPTree. FDT scored slightly better than FURIA with the GA feature selection for 3 of 5 data sets. In addition, all the algorithms using the GA feature selection achieve a better value than without using the GA feature selection in most cases. But it is not necessary to be true in all cases. Sometimes, it is even worse since the GA feature selection process ignores some features which cause information loss.

Data Set	J48	REPTree	FURIA	FDT
Diabetes	32.7 (*32.3*)	32.8 (*34.8*)	35.7 (*36.2*)	**31.9** (***31.3***)
Glass	13.0 (*10.4*)	13.8 (*12.3*)	13.1 (*11.1*)	**12.4** (***10.2***)
Ionosphere	12.5 (*11.6*)	13.2 (*11.9*)	12.3 (*14.3*)	**11.9** (***11.2***)
Pendigits	0.4 (*0.7*)	0.5 (*0.9*)	**0.2** (***0.6***)	0.4 (*0.8*)
Vehicle	9.3 (*10.7*)	9.3 (*10.2*)	9.8 (*11.1*)	**9.2** (***10.1***)

Table IV
AVERAGE WEIGHTED FPR (%) OF ALL ALGORITHMS.

Data Set	J48	REPTree	FURIA	FDT
Diabetes	73.5 (*74.4*)	74.7 (*72.3*)	73.7 (*74.4*)	**75.7** (***76.5***)
Glass	67.0 (*71.5*)	65.8 (*69.0*)	**70.5** (***72.1***)	68.5 (*70.3*)
Ionosphere	91.5 (***91.8***)	89.4 (*90.6*)	91.2 (*89.4*)	91.6 (***91.8***)
Pendigits	96.6 (*93.9*)	95.6 (*92.3*)	**98.0** (***94.8***)	96.8 (*94.1*)
Vehicle	72.2 (*67.9*)	71.1 (*68.3*)	68.8 (*63.9*)	**73.8** (***71.2***)

Table V
AVERAGE WEIGHTED PRECISION (%) OF ALL ALGORITHMS.

AUC is the area under the ROC curve. ROC stands for "Receiver Operation Characteristic", which is part of a field called "Signal Detection Theory" developed during World War II for the analysis of radar images [33]. Two different methods are used to calculate the AUC. J48, REPTree and FURIA use a parametric method using a maximum likelihood estimation to fit a smooth curve to the data points since these algorithms are part of the WEKA software. Our FDT classifier uses a non-parametric method based on the construction of a trapezoid under the curve as an approximation of the area as follows:

$$AUC = \frac{1 - FPR + TPR}{2} \tag{7}$$

In order to compare the performance of using the GA-based feature selection method, the AUC of all the algorithms with/without the feature selection are measured. J48-P, REPTree-P, FURIA-P and FDT-P are abbreviations for the algorithms using the feature selection preprocessing method.

In Figure 2, J48 when the GA-based feature selection method was applied achieved the same or even better AUC values for 4 out of 5 data sets except for Pendigits.

The AUC values when using REPTree with the reduced feature set are either the same or less than using REPTree when all features are used. Only 2 out of 5 data sets achieve better results using REPTree with the reduced feature set.

Evaluating AUC on the FURIA classifier shows that 4 out of 5 data sets have less AUC values when the GA-based feature selection method is used. It seems that FURIA suffers from over-fitting when using the GA-based preprocessing method. The results are shown in Figure 4.

As shown in Figure 5, the AUC values of 3 of 5 data sets show slight improvement when using FDT with the GA-

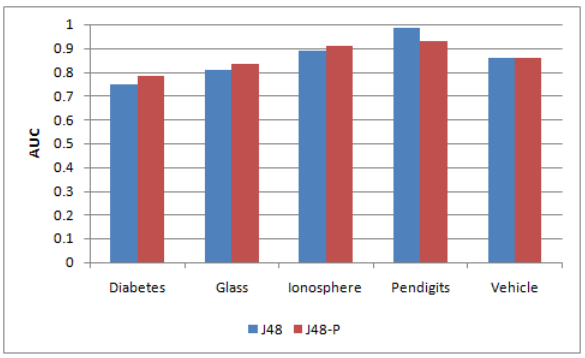

Figure 2. AUC of J48 and J48-P.

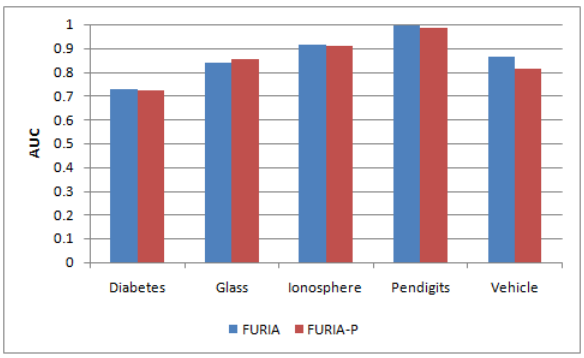

Figure 4. AUC of FURIA and FURIA-P.

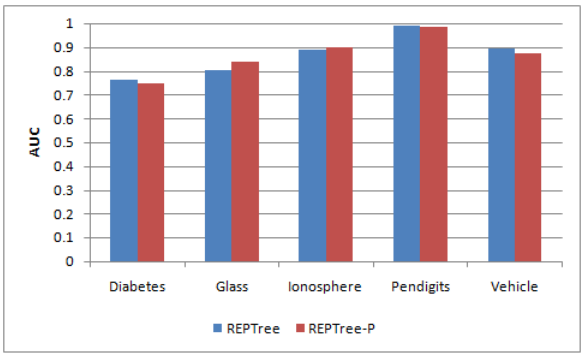

Figure 3. AUC of REPTree and REPTree-P.

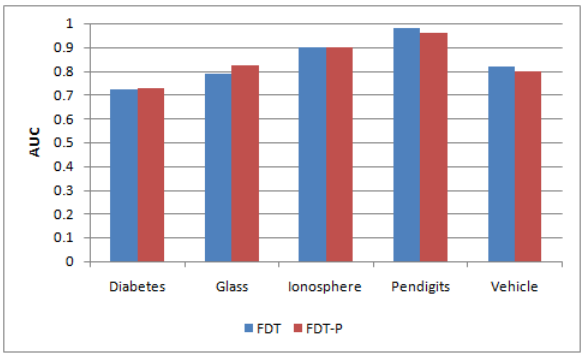

Figure 5. AUC of FDT and FDT-P.

based feature selection method.

Generally, J48 and FDT, except REPTree and FURIA, achieve slightly higher AUC values when the GA-based feature selection is used as the data preprocessing method.

V. Conclusion

In this paper, we presented a fuzzy decision tree (FDT) approach using a GA-based feature selection method. The FDT approach uses soft-discretization searching for the best cut-point in order to improve the predictive accuracy. The soft-discretization works by partitioning the decision space into a set of overlapping subspaces instead of using crisp discretization partitioning. Futhermore, since the reduction of the feature space has shown to improve the accuracy of classifiers in general, we investigated a GA-based feature selection method combined with our FDT approach. In terms of the value of AUC, the results show that J48 and FDT can score better by applying the GA feature selection on most data sets. However, REPTree and FURIA can not achieve as good values as without using GA feature selection.

Our FDT classifier was compared to J48, REPTree, and FURIA both with and without the GA-based feature selection method. Five continuous-valued data sets taken from the UCI repository were used. Overall, the results revealed that the approaches using soft discretization rather than hard discretization, such as FURIA and our FDT classifier, obtained better predictive classification accuracy in terms of TPR, FPR, precision and AUC. Furthermore, our proposed classifier achieved slightly better results than FURIA in most cases.

As for future work, we will investigate and compare other feature selection techniques available in terms of improvements in accuracy but also in terms of execution time.

References

[1] L. Huan, F. Hussain, C.L. Tan, and M. Dash. “Discretization: An enabling technique.” Data mining and knowledge discovery 6, no. 4 (2002): 393-423.

[2] F. Provost and V. Kolluri. “A survey of methods for scaling up inductive algorithms.” Data Mining and Knowledge Discovery, 3(2):131169, 1999.

[3] P. Geurts, D. Ernst, and L. Wehenkel. “Extremely randomized trees.” Machine Learning, 36(1):342, 2006.

[4] E. Hullermeier, and S. Vanderlooy. “Why fuzzy decision trees are good rankers.” Fuzzy Systems, IEEE Transactions on 17, no. 6 (2009): 1233-1244.

[5] J.R. Quinlan. “Induction of Decision Trees.” Mach. Learn. 1, 1 (Mar. 1986), 81-106, 1986.

[6] J.R. Quinlan. “C4. 5: programs for machine learning.” Vol. 1. Morgan kaufmann, 1993.

[7] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone (1984). "Classification and regression trees." Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software. ISBN 978-0-412-04841-8.

[8] H.S. Nguyen. "On exploring soft discretization of continuous attributes." In Rough-Neural Computing, pp. 333-350. Springer Berlin Heidelberg, 2004.

[9] L.A. Zadeh, "Fuzzy sets." Information and Control, Vol. 8, pp. 338-353, 1965.

[10] L.A. Zadeh, "The concept of linguistic variable and its application to approximate reasoning." Inf Sci, 8 (1975).

[11] O. Ludwig, and U. Nunes. "Novel maximum-margin training algorithms for supervised neural networks." Neural Networks, IEEE Transactions on 21, no. 6 (2010): 972-984.

[12] I. Guyon, and A. Elissee (2003). "An introduction to variable and feature selection." In Journal of Machine Learning Research, volume 3, pages 1157-1182.

[13] J. Walters-Williams and Y. Li, " Estimation of mutual information: a survey". In: RSKT 2009: 4th International Conference on Rough Set and Knowledge Technology , 14-16 Jul 2009, Gold Coast, Australia.

[14] M. Kantardzic. "Data Mining: Concepts, Models, Methods, and Algorithms." IEEE Press & John Wiley, November 2002.

[15] C.Z. Janikow, "Fuzzy decision trees: issues and methods." IEEE Trans. Systems Man, CyberneticsPart B: Cybernetics 28 (1) (1998) 114.

[16] I.H. Witten and E. Frank. "Data Mining: Practical Machine Learning Tools and Techniques." Morgan Kaufmann, San Francisco, Calif, USA, 2nd edition, 2005.

[17] P. Domingos and M.J. Pazzan. "On the Optimality of the Simple Bayesian Classifier under Zero-One Loss." Machine Learning 29(2-3): 103-130 (1997).

[18] I. Kononenko, "Machine learning for medical diagnosis: history, state of the art and perspective." Articial Intelligence in Medicine, 23(1), 89109.

[19] P. Langley, "Induction of recursive Bayesian classiers." In Proceedings of the European conference on machine learning (pp. 153164), 1993.

[20] N. Lavrac, "Data mining in medicine: selected techniques and applications." In Proceedings of the 2nd international conference on the practical applications of knowledge discovery and data mining (pp.1131), 1998.

[21] D.D. Lewis, " Naive (Bayes) at forty: the independence assumption in information retrieval." In Proceedings of the 10th European conference on machine learning (pp. 415), 1998.

[22] J. Chen, H. Huang, S. Tian, and Y. Qu. "Feature selection for text classification with Nave Bayes." Expert Systems with Applications 36, no. 3 (2009): 5432-5435.

[23] G.J. Klir and T.A. Folger, "Fuzzy Sets, Uncertainty and Information." Prentice Hall 1988.

[24] Y.H. Peng, T.J. Chen, and P.W. Tse, "Fuzzy Inductive Learning in Data Mining and its Application to Machine Fault Diagnosis." 2000AMSMA international conference, pp.158-163, Guangzhou, China, (2000) 158-163.

[25] D.D. Lewis and M. Ringuette. "A comparison of two learning algorithms for text categorization." In Third annual symposium on document analysis and information retrieval, vol. 33, pp. 81-93. 1994.

[26] P.A. Estevez, M. Tesmer, C.A. Perez, and J.M. Zurada. "Normalized mutual information feature selection." Neural Networks, IEEE Transactions on 20, no. 2 (2009): 189-201.

[27] J. Chen, H. Huang, S. Tian, and Y. Qu. "Feature selection for text classification with Nave Bayes." Expert Systems with Applications 36, no. 3 (2009): 5432-5435.

[28] J. Richard, and Q. Shen. "New approaches to fuzzy-rough feature selection." Fuzzy Systems, IEEE Transactions on 17, no. 4 (2009): 824-838.

[29] H. Yang, J. Moody, "Feature selection based on joint mutual information, Proceedings of Advances in Intelligent Data Analysis (AIDA), Computational intelligence methods and applications (CIMA), Rochester, New York, 1999.

[30] S.S. Skiena (27 April 2011). "The Algorithm Design Manual." Springer. p. 129. ISBN 978-1-84800-069-8.

[31] A. Frank, and A. Asuncion, (2010). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[32] J.C. Huhn, E. Hullermeier, "FURIA: an algorithm for unordered fuzzy rule induction." Data Min. Knowl. Discov. 19(3): 293-319 (2009).

[33] T. Hastie, R. Tibshirani, J.H. Friedman (2009). "The elements of statistical learning: data mining, inference, and prediction (2nd ed.)".