Fuzzy Decision Forest

Cezary Z. Janikow
Department of Math and Computer Science
University of Missouri – St. Louis
cjanikow@ola.cs.umsl.edu

Abstract

In the past, we have developed and presented a Fuzzy Decision Tree, more recently followed by an extension called a Fuzzy Decision Forest. The idea behind the forest is not only to represent multiple trees, but also to represent test alternatives at all levels of every tree. The resulting tree is in fact a 3-dimensional tree. A two-dimensional slice is equivalent to a single decision tree. The forest allows multiple choices of tests in some or all nodes of the decision tree. These alternative tests can be used to enhance the classification accuracy of the tree. However, the major advantage of having multiple test choices is to have alternative test decisions when features in test data are unreliable or just missing. In the paper, we overview the ideas behind Fuzzy Decision Forest, and we illustrate its enhanced capabilities with a number of experiments with missing features.

1. Introduction

In today's era of massive amounts of data, computer programs that are able to process and reason from data are of high importance. For classification tasks, decision trees proved to be one of the most successful methodologies [1][6][7]. The extracted knowledge, in the form of a decision tree along with inference procedures, has been praised for accuracy, efficiency, and comprehensibility.

Decision trees, originally proposed for symbolic domains and with a simple decision procedure [6], have enjoyed many methodological advancements, such as ability to produce binary trees and dealing with continuous data [1], new inference procedures, e.g., to compute probabilities of decisions [7], and finally incorporation of fuzzy sets and uncertain reasoning inferences to account for noisy and uncertain environments [2][8]. A decision tree is made up of two elements: a recursive top-down partitioning procedure, generating a decision tree, and then an inference rule from the resulting tree. The procedure starts with the training data, expressed by combinations of features according to the available variables and domains, and classified into some classes. The partitioning procedure selects one test at a time, usually based on one feature, and splits the data into subsets according to the tested features. The selected test is to maximize some objective, such as separation of examples of different classes [7]. The recursive procedure stops upon perfect class separation or based on some other objectives [7]. The subsequent inference rule uses the tree to assign new test data to some of the same classes.

Fuzzy sets and logic have been proposed to deal with language or data related uncertainties [9]. Combined with uncertain reasoning, fuzzy representation provides for greater stability and robustness. This representation has been incorporated into decision trees, resulting in trees still satisfying their standard advantages, yet also more robust and stable [2][8]. A Fuzzy Decision Tree (FID) is one such extension [2]. FID can deal with data described by a mixture of symbolic and continuous variables. FID originally required all domains to be pre-partitioned into fuzzy sets. It has later been extended to allow a mixture of pre-partitioned and un-partitioned variables [3][4]. However, FID still suffers from the same traditional disadvantage as all decision trees. The decision tree procedure attempts to minimize the number of tests needed to classify the training data. This greatly improves comprehensibility, but it also reduces the amount of learned characteristics about the data. Recognizing this as a potential problem, researchers have proposed extensions, such as extracting a number of diverse decision trees, which subsequently vote on or apply another decision procedure to classify new data.

A Fuzzy Decision Forest (FDF) [5] incorporates similar ideas into Fuzzy Decision Trees. The resulting knowledge is higher dimensional, and thus less comprehensible. Yet simple slices of the FDF forest reduce the representation to simple trees. Moreover, the resulting forest improves classification accuracy, especially when dealing with missing features in testing data. In this paper, we review the ideas behind Fuzzy Decision Forest, and then present some experimental results illustrating its enhanced capabilities.

2. Fuzzy Decision Forest

The procedure to build a decision tree selects a single test at every node of the tree, which maximizes some objectives on separation of data belonging to different classes. This single test produces minimal knowledge – the decision tree procedure is an example of a discriminant learning procedure, where the objective is to minimize class descriptions, and thus to minimize the set of tested variables and features. When two tests offer similar quality in a node, one of them needs to be tossed away, and the decision sometime is quite random. The test that is tossed away reduces the knowledge expressed in the tree.

Different tests performed in a node can lead to different decisions. Retaining those multiple tests, combined with a conflict resolution procedure, increases classification potentials. This is especially important in two situations:

1. The feature associated with the winning test may be noisy or inaccurate in a given test data. Retaining the alternative tests increases the predictive accuracy.
2. The feature associated with the winning test can be missing in a given test data Traditional way to deal with this is to test all possible cases of the feature and then resolve the resulting conflicts [7]. Again, retaining other relevant tests allows alternatives for more comprehensive and informed reasoning.

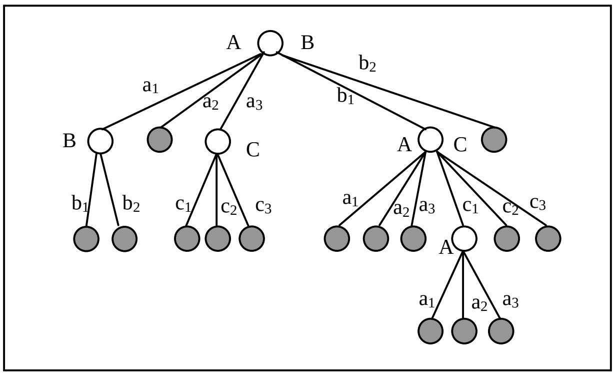

Figure 1. A Fuzzy Decision Forest.

The Fuzzy Decision Forest extends FID trees by allowing alternative tests to be performed at all nodes. FDF builds the tree exactly like FID would [2], except for the following: in a given node, more than one test can be selected. When this happens, each of the tests results in growing different subtrees. The actual selected tests, and the number of them, are based on some heuristics and parameters. Alternatives offering similar level of class separation are maintained; however, the number of potential alternative tests diminishes at deeper levels. The resulting tree is in fact a forest if it has more than one test at the root. Moreover, alternative tests can also be explored at deeper levels, resulting in a 3-dim tree [5].

One may produce a *slice* of the forest by selecting a single test at each node. A slice of the FDF forest is indeed an FID tree. Selecting the best test at each node with alternative tests produces a forest equivalent to the FID tree that would have been built from the same data.

Of course one needs an inference procedure that is able to explore the extra information retained in the forest. FID provides a number of inference procedures [2]. Each inference procedure takes all leaves whose path-restrictions match the test data and combines the classifications of those leaves in some fashion. When the data is matched by more than one tree in FDF, the result is simply a greater number of leaves participating in the final vote. Each slice of the tree carries out its own vote. Then, another inference combines the classification presented by each slice. The inference can be

a) a sum of the individual votes for each class (a simple vote),
b) a weighted sum, weighted by the strength of each test used in the slice producing the result (the slice matching the test data better has a higher vote),
c) a weighted sum, weighted by the strength of each test used in the producing the result, additionally weighted by the number of training data matching the same tests in this slice (the slice matching the test data better but also having more training data has a higher vote).

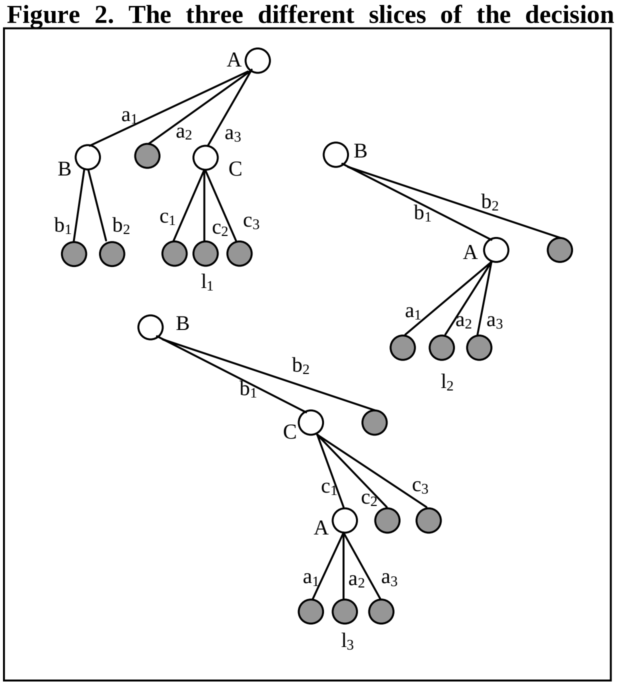

Figure 2. The three different slices of the decision forest from Figure 1.

An FDF forest is illustrated in Figure 1, where we assume three variables A, B, and C, with domains as indicated. The forest uses only two alternative tests in two nodes, and it is shown in 2-D.

The illustrated forest can be sliced three different ways, as illustrated in Figure 2. Suppose the first slice corresponds to the dominant FID tree that would have been constructed if not for the alternative tests. Now suppose we have test data with the following features: $A=a_2$, $B=b_1$, $C=c_1$. The first slice would classify the data as belonging to the class in its l_1 leaf, the second as according to its l_2 leaf, and the third as according to its l_3 leaf. When resolving a potential conflict between these three responses, one may weight higher the response from l_1 as coming from the dominant slice. Alternatively, if l_2 has many more training data, its response may be weighted higher. This illustrates the potential inferences from the forest.

3. Experiments

We have conducted two sets of experiments, one on actual data from the ML depository – Glass data, the other one with purposely-modified Glass data.

3.1 Glass data

Glass data is one of the standard data sets used in machine learning. It contains 214 data samples of 7 different glass classes. Each data sample is described by 9 continuous attributes without any missing values.

Table 1. Training and testing on the Glass data.

Method	Training success	Testing success w/o any missing features
FID decision tree	**67.9%**	**64.7%**
FDF decision forest	**72.7%**	**69.8%**

First, we have trained an FID decision tree and an FDF decision forest in a 10-fold cross validation setup, measuring the error on the training data while facing the same termination criteria (to avoid overspecialization with one of the experiments). The results are presented in Table 1. As seen, FDF forest trains to recognize the training data to a higher success rate.

This by itself may not be relevant if FDF achieved higher training rate by overspecializing its trees. To verify that, we tested the generated FID tree and FDF forest with testing data, again in the same 10-fold cross-validation setup, starting with the actual testing data. The results are presented in Table 1 and indicate higher success rate from the FDF forest. Then, we repeated the same tests, but with various percentages of features removed from the testing sets. These results are presented in Figure 3. As seen, FDF presents higher robustness to missing features than FID alone.

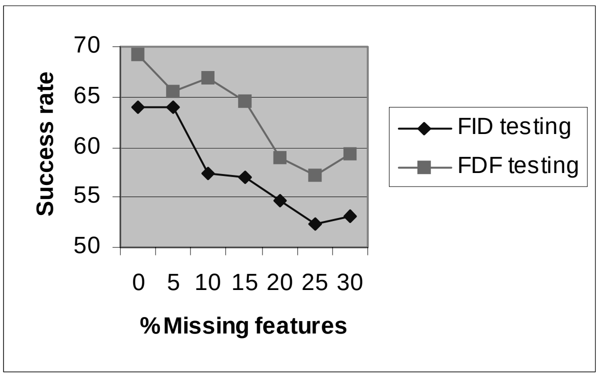

Figure 3. Testing with missing features on Glass data.

3.2 Modified Glass data

We have also modified the Glass data as follows: for each of three random attributes, we have added two more attributes with feature values generated at random but in such a way that the correlation of the new attributes to the original attribute is 0.75 and 0.5. This was intended to simulate the case where one or more attributes are correlated and can produce similar tests. Then we repeated the previous 10-fold cross-validation experiment: training FID tree and FDF forest and then testing them on data with missing features. The results presented in Figure 4 indicate that in this case FDF is able to take advantage of the correlations in attributes to build additional slices to increase its prediction rate (given our run parameters, the number of slices increased from 6 to 11).

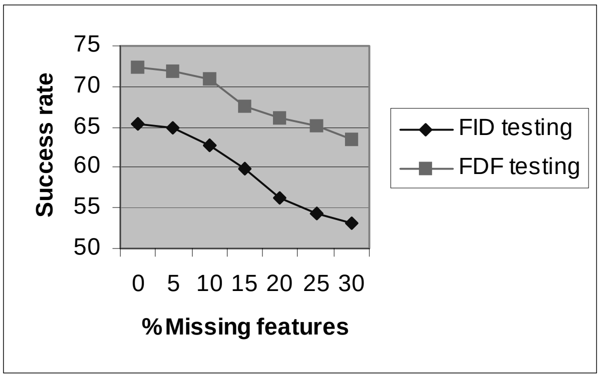

Figure 4. Testing with missing features on the modified Glass data.

4. Conclusions

We have presented the concept of a Fuzzy Decision Forest, which extends Fuzzy Decision Trees by allowing multiple tests to be retained at some nodes of the tree. The resulting tree is indeed a 3-dim forest. The forest can be sliced, producing single decision trees. However, a number of slices can be used in the inference procedure to classify test data. This procedure is especially helpful when some features in the test data are noisy, uncertain, or just missing. Experimental results do indeed verify that the resulting forest is more capable to reason under such unfavorable yet often encountered conditions.

The software is available from `http://www.cs.umsl.edu/~janikow/FID`.

10. References

[1] Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J., *Classification and Regression Trees*, Wadsworth and Brooks, 1984.

[2] C.Z. Janikow, "Fuzzy Decision Trees: Issues and Methods", *IEEE Transactions on Man, Systems, and Cybernetics*, Vol. 28, Issue 1, pp. 1-14, 1998.

[3] M. Fajfer, C.Z. Janikow, "Bottom-up Partitioning in Fuzzy Decision Trees", *Proceedings of the 19th International Conference of the North American Fuzzy Information Society*, Atlanta 2000, pp. 326-330.

[4] C.Z. Janikow, M. Fajfer. "Fuzzy Partitioning with FID3.1", *Proceedings of the 18th International Conference of the North American Fuzzy Information Society*, NY 1999, pp. 467-471.

[5] C.Z. Janikow, M. Fajfer, "Fuzzy Decision Forest", *Proceedings of the19th International Conference of the North American Fuzzy Information Society*, Atlanta 2000, pp. 218-221.

[6] J.R. Quinlan, "Induction on Decision Trees", *Machine Learning*, Vol. 1, 1986, pp. 81-106.

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

[8] M. Umano, H. Okamoto, I. Hatono, H. Tamura, F. Kawachi, S. Umedzu, J. Kinoschita, "Fuzzy Decision Trees by Fuzzy ID3 Algorithm and Its Application to Diagnostic Systems", *Proceedings of the Third IEEE Conference on Fuzzy Systems*, 1994, pp. 2113-2118.

[9] L.A. Zadeh, "Fuzzy Sets", *Information and Control* 8:338-353, 1965.