Summarizing Fuzzy Decision Forest by Subclass Discovery

Christophe Marsala* and Maria Rifqi*,†
* UPMC Univ Paris 06, UMR 7606, LIP6, F-75005, Paris, France
* CNRS, UMR 7606, LIP6, F-75005, Paris, France
† Université Panthéon-Assas – Paris 2, F-75005, Paris, France
Email: christophe.marsala@lip6.fr, maria.rifqi@lip6.fr

***Abstract*—Construction of forest of decision trees method is a popular tool in machine learning because of its good performances in terms of classification power as well as in computational cost. In this paper, we address two problems. The first one concerns the interpretability of a forest. Indeed, comparing to a single decision tree, a forest loose its ability to be easily understandable by an end-user. The second studied problem concerns the size of the forest and hence the memory size and classification time of a forest. We seek for a forest as small as possible that classify nearly as well as a larger forest. In order to solve these two problems, we propose to characterize a forest by discovering different classes of trees regarding their power of classification. These classes are discovered thanks to Forest's algorithm [1] of class segmentation, a variant of the hypersphere classifier [2].**

***Keywords*—Fuzzy decision forest, supervised clustering, pruning.**

I. Introduction

Fuzzy Decision Forest (FDF) or forest of fuzzy decision trees has been introduced in order to increase the classification capabilities of fuzzy decision trees.

FDF comes from classical Machine Learning where ensembles approaches have been developed in order to improve classical learning algorithms. Here, we cite the bagging (Bootstrap aggregating) approach [3], that are based on the bootstrap method [4] and the use of a machine learning algorithm, or the boosting approach [5], [6], [7] that proposes the construction of a *strong* learner as a set of *weak* learners to obtain an accurate prediction rule [8]. AdaBoost is a well-known boosting algorithm proposed by [9]. Also, (Random) Forests of decision trees have been introduced to lower the error rate of fuzzy decision trees when classifying new cases [10], [11], [12], [13].

In fuzzy machine learning, ensemble approaches based on fuzzy decision trees have been introduced to handle numerical or fuzzy data [14], [15], [16], [17], [18], [19]. The combination of a fuzzy learning algorithm with the ensemble approach takes advantages of the fuzzy decision of a fuzzy classifier when classifying new cases. Moreover, it is a fast and accurate machine learning tool that takes benefit of a fuzzy classification decision to increase its robustness and accuracy. For instance, these approaches offer good results in applications as for instance in video mining [20] where they have been used both to enhanced the accuracy of fuzzy decision trees, and to output a ranking of examples in classification.

There is three main difficulties to tackle with such approaches. First of all, a fast algorithm is needed to construct the ensemble of learners. Secondly, the number of fuzzy decision trees should be set in order to obtain the best performances with a limited number of trees. Finally, the understandability of a fuzzy decision forest is generally very poor due to the fact that the number of the fuzzy decision trees highly increases the complexity of the model. That last point is a very dreadful drawback in the sense that fuzzy decision trees are greatly famous due to their understandability. Used in a forest, they loose one of their main advantage.

In this paper, we propose an approach to summarize a forest of (fuzzy) decision trees by giving a set of significant accurate trees. A similar approach have been proposed in [21] by means of the use of a clustering algorithm to cluster the trees of the forest and kept only some representative trees. However, that approach did not use any knowledge to help the selection of a subset of the trees and need the setting of several parameters for the use of the clustering algorithms. We propose to find different subsets of accurate trees by Forest's algorithm [1]. In our approach, we try to keep the classification rate by selecting representative trees in the different subsets. We introduce the notion of quality of a tree based on the notion of its ability to correctly classify examples that are mainly badly classified by the trees of the forest.

Instead of in [21], the main aim of our approach is to take benefit of the knowledge on the fuzzy decision trees having a high accuracy when classifying "easy" examples but also of fuzzy decision trees that have a low accuracy on easy examples but a high classification rate for "difficult" examples. The combination of such different prototypes of fuzzy decision trees could thus enable to perform an increase of the global accuracy of the fuzzy decision forest by means of a small number of fuzzy decision trees, but also to provide a summarization of the FDF by this small sample of FDF.

The paper is organized as follows: first, used algorithms are presented in Section II and Section III; Section IV explains our approach and Section V gives results on a UCI database and analyses them; finally the paper ends by a conclusion and some future work.

II. Fuzzy Decision Forest

Fuzzy Decision trees have been proposed to handle continuous or fuzzy attributes and to propose a way to construct a set of understandable fuzzy rules from a training set. Fuzzy set theory brings out more robustness and more interpretability when handling such kind of data.

A. Background

Fuzzy decision tree construction algorithms are generalization of classical algorithms [22], [23] that are very popular in data mining even if they encounter some problems when dealing with numerical attributes. In fuzzy decision trees, the use of fuzzy values has been introduced to enhance the ability to take into account numerical values, to allow smoother decisions and provide membership degrees instead of binary classifications [24], [25].

Given a training set $T = \{X_1, \ldots, X_n\}$, a fuzzy decision tree is built from its root to its leaves, by successively partitioning T into subsets. Each partition is done by means of a test on an attribute, which leads to the definition of a node of the tree.

Let us assume that each example from the training set is described by means of a set of attributes $\mathcal{A} = \{A_1, ..., A_m\}$. Each example is associated with a class C_k from the set $C = \{C_1, ..., C_K\}$. Here, each attribute A_j can take a numerical or fuzzy value.

In this paper, the fuzzy decision tree construction algorithm is not recalled. It can be found, for instance, in [25].

B. Construction of a fuzzy decision forest

A fuzzy decision forest is composed of a given number of fuzzy decision trees. A first approach to construct a fuzzy decision forest has been proposed in [14] to handle applications with more than two classes to predict. A forest of fuzzy decision trees is constructed by considering a n-classes problem as a set of n two-classes problems. Thus, each fuzzy decision trees of the forest can be constructed to predict a class against all the other classes. When classifying examples, several aggregation methods (normalized vote, unnormalized vote, possibilistic aggregation,...) have been proposed to aggregate the classification results of each fuzzy decision tree.

Various methods have been introduced later to construct fuzzy decision forests. First of all, some approaches modify the labels of the training examples to construct fuzzy decision trees [14]. Secondly, some methods used a set of samples of the training set to construct several fuzzy decision trees. For instance, [18] introduced an approach based on random forests. Finally, several approaches introduce the use of the given training set, without any sampling. They introduce some modification of the selection process of the test nodes, either by deleting some attributes from the list of attributes [16], or by choosing a set of attributes instead of a unique one [15], [17]. In [16], a set of fuzzy decision trees is first constructed classically and the fuzzification of the test nodes is done at the end of the construction of the whole set of trees. In [15], [17], the trees are constructed by considering at each node, not only the best attributes, but all the best ones if there are several attributes that can be convenient to split the training set.

In our approach [19], fuzzy decision forest is composed of N_F fuzzy decision trees. Each fuzzy decision tree DT_i is constructed from a training set T_i which is a random sample of the whole original training set. When the distributions of classes in the training set are highly unbalanced, the algorithm to construct (fuzzy) decision trees cannot be applied directly and should be adapted otherwise the majority class will cover the other classes. In that case, the sampling sets T_i are drawn in order to be composed of an equal number of examples from each class.

C. Classification by a fuzzy decision forest

A fuzzy decision forest is usually constructed to classify any forthcoming examples. A fuzzy decision tree provides a membership degree $f_{C_k}^{DT_i}$ to a predicted class C_k for any example it classifies.

With a fuzzy decision forest composed of N_F fuzzy decision trees, the classification of an example X is performed in two steps. First of all, X is classified by each of the N_F fuzzy decision trees DT_i of the forest in order to obtain a degree $f_{C_k}^{DT_i}(X)$ of X to have the class C_k. Secondly, the aggregation $\bigoplus$ (sum or average for instance) of the $f_{C_k}^{DT_i}(X),\ i = 1 \ldots N_F$ degrees for each X in order to obtain a single value:

$$f_{C_k}(X) = \bigoplus_{i=1}^{n} f_{C_k}^{DT_i}(X)$$

which corresponds to the degree of membership of X to C_k provided by the fuzzy decision forest.

III. Subclass Discovery

A. Related work

Among supervised learning methods, some are dedicated to represent each class by a set of prototype regions. We are interested in those that partition a class (or category) in hyperspheres like in [2], the Restricted Coulomb Energy (RCE) network [26] or the set covering machine (SCM) [27].

The final goal of all these methods is the classification of a new example: the class assigned to it is the class of the closest prototype. It is not our aim to classify the fuzzy decision trees but to find subclasses of trees in order to provide a set of trees ie the prototypes of the subclasses. This is the reason why we choose the algorithm proposed by Forest [1] close to the one proposed by Wang et al. [28].

The principle is more or less the same for all these methods. They differ on the necessary parameters. The Forest's algorithm is detailed below.

B. Forest's algorithm

Given a database of n examples: $D = \{X_1, \ldots, X_n\}$, where X_i are described in R^d and belongs to a class $C_k \in C$ where $k = 1, \ldots, K$.

Input : $D = \{X_1, \ldots, X_n\}$, $C = \{C_1, \ldots, C_K\}$, $S = \emptyset$
for each $X_i \in D$ of class C_k **do**
– create a **sphere** $S_{X_i,C_k,r} \in S$ centered on X_i and with radius $r = \min_{X_j \notin C_k} d(X_i, X_j)$ associated with class C_k
– create the **graph of direct friends** of X_i, $F_{X_i} = \{(X_i, X)/X \in S_{X_i,r}\}$
end for

– create the **global graph of friends** as the union of the direct graphs $F = \bigcup_i F_{X_i}$
– create the **subclasses** $SC \subseteq C_k$ as the set of connected components of F.

Forest's algorithm can be decomposed in 4 steps. The first one consists in growing a sphere centered on an example X_i until an example of a different class is reached. In the second step, this sphere provides the graph of direct friends of X_i, ie a vertex links X_i and each example in its sphere (see figures 1a and 1b). The third step aggregates all the direct graphs by union in order to obtain the global graph. Lastly, the class of X_i is partitioned in subclasses where a subclass is defined as a connected component of the global graph (see figure 1c).

It has to be noted that a sphere covers only examples of a same class but this constraint can be soften by introducing a parameter of tolerance t which allows to grow a sphere until t examples of a different class are reached.

IV. Summarizing a fuzzy decision forest

Machine learning meta-algorithms lie on the hypothesis that a set of weak learning models forms a strong learning model. In the case of decision forests, some decision trees can be weak considering their classification performance, but associated to other less weaker trees, they constitute a good learning model. It can be explained on the fact that some tree are specialized on a sparse region of the learning space, difficult to learn and hence with possibly bad classification performance, whereas other trees are specialized on a dense region, easier to learn, and hence with possibly good classification performance.

Our goal is to summarize a decision forest by keeping the trees that cover the best the learning space. We need to find a representative tree for each region covered by the learned trees of the forest. Hence, the trees can not be separated on weak trees and strong trees: their performance should be balanced by the level of difficulty they have to face to.

A. Main notions

Our idea is to score each example of the test database by its ease to be learned, ie by its ease to be correctly classified by a forest. The more an example is correctly classified by a forest, the more it is easy to learn it. This score is used to qualify the quality of a fuzzy decision tree. A tree can be of high quality even with a medium error rate because it classifies correctly examples that are mainly incorrectly classified by the fuzzy decision forest.

Let us consider a forest of N_F fuzzy decision trees DT_j. Each DT_j provides a membership degree to a class C_k for an example X_i: $f_{C_k}^{DT_j}(X_i)$.

Definition 1: The degree of ease ε to be learned of an example X_i is defined as:

$$\varepsilon(X_i) = \frac{\sum_{j=1}^{N_F} f_{C_k}^{DT_j}(X_i)}{N_F} \tag{1}$$

We introduce the degree of quality κ of a fuzzy decision tree. This degree depends on the degrees of ease of the examples it correctly classifies.

Let us consider the binary function:

$$cc_i = \begin{cases} 1 & \text{if } X_i \text{ is correctly classified} \\ 0 & \text{else} \end{cases} \tag{2}$$

Definition 2: The degree of quality κ of a fuzzy decision tree DT_j is defined as:

$$\kappa(DT_j) = \frac{\left(\sum_{i=1}^{n} (1 - \varepsilon(X_i))\right) \cdot cc_i}{n} \tag{3}$$

B. Proposed approach: HQT forest summarization

Once a fuzzy decision forest F is learned on a training database T, the forest is tested on a test database D. The quality κ of each tree of F is evaluated on D. Two classes of trees are build: the class of high quality trees (HQT) and the class of low quality trees (LQT). The separation is based on a threshold κ_t on κ. In this paper, the threshold is set to the average κ.

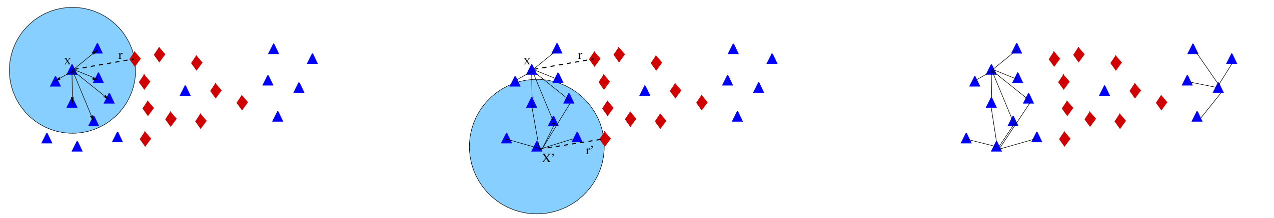
Forest's algorithm is applied to the whole forest. It means that the database of examples are the trees of F belonging to HQT or LQT. Each tree is described by means of the degrees it provided for the $X_i \in D$.

Once the subclasses are obtained, a tree is chosen for each subclass of HQT. The classification process is performed once again with these selected trees.

Input : $D = \{X_1, \ldots, X_n\}$, $C = \{C_1, \ldots, C_K\}$, $F = \{DT_1 \ldots, DT_{N_F}\}$, κ_t
for each $X_i \in D$ **do**
  – compute $\varepsilon(X_i)$
end for
for each $DT_j \in F$ **do**
  – compute $\kappa(DT_j)$
end for
– create two classes of trees: $HQT = \{DT_j / \kappa(DT_j) \geq \kappa_t\}$, $LQT = \{DT_j / \kappa(DT_j) < \kappa_t\}$
– perform Forest's algorithm on $F = HQT \cup LQT$ where DT_j is described by $\{f_{C_k}^{DT_j}(X_1), \ldots, f_{C_k}^{DT_j}(X_n)\}$
– choose a tree in each subclass of HQT.

V. Experimental results

In order to validate our approach (HQT forest summarization), a set of experiments have been conducted on Gisette dataset from the UCI repository [29]. This dataset is one of the datasets of the NIPS 2003 feature selection challenge [30]. The goal is to discriminate between handwritten digits: the four and the nine. The examples are described by 5000 sparse continuous variables (approximately 13% of the entries are non zero). The dataset is composed of two databases: a training database of 6000 examples, equally distributed on the two classes; and a validation database of 1000 examples also equally distributed on the two classes.



(a) A sphere $S_{X,blue,r}$ and the associated graph of direct friends (b) A sphere $S_{X',blue,r'}$ and the associated graph of direct friends (c) Connected components of the blue class

Fig. 1. Some steps of Forest's algorithm

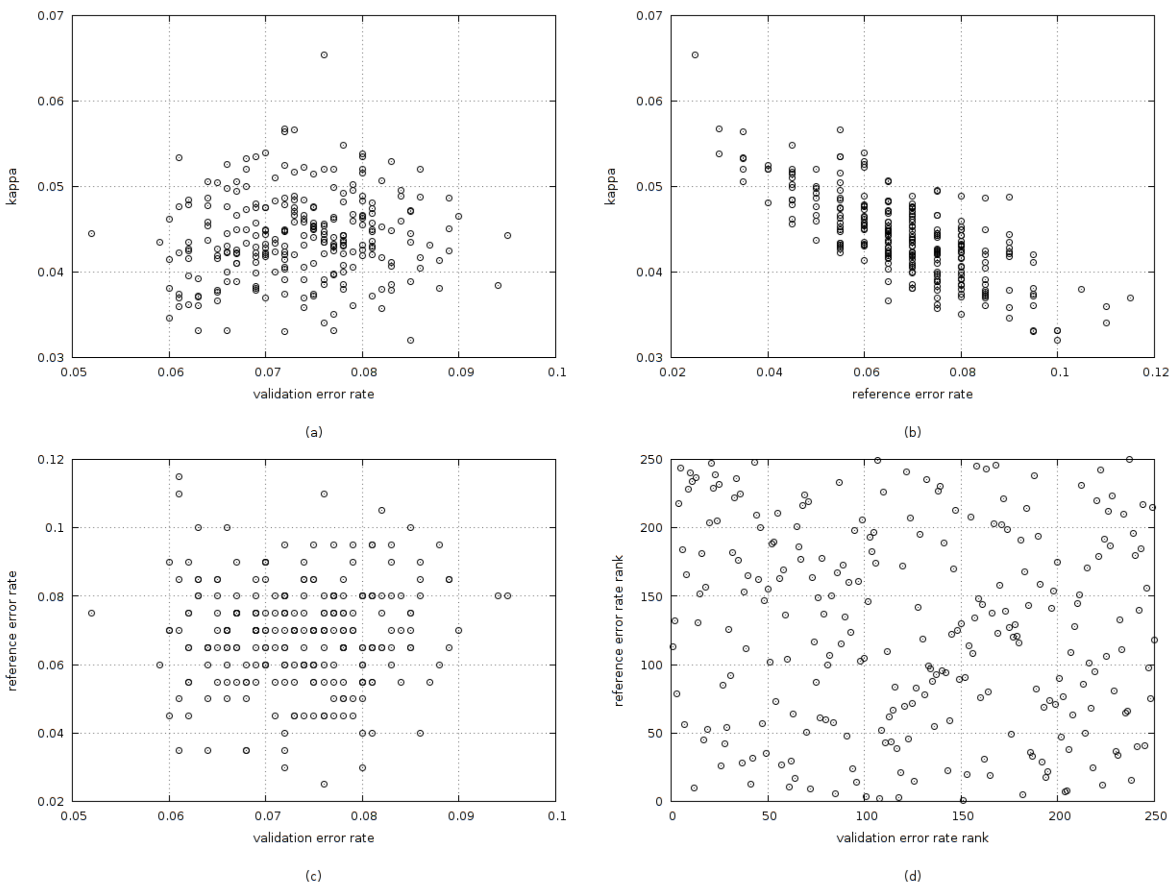


Fig. 2. Results on experiment 1

A. Protocol description

The training database is arbitrary divided in two databases: one called train with 5800 examples, the other one called reference with 200 examples. The choice of a convenient size for this reference set deserves a more deeper study that will be done in further work.

The train database is used to construct a fuzzy decision forest. The reference database is used to test the obtained forest. The κ degrees are calculated on the forest results on the reference database and Forest's algorithm is then performed from these κ degrees. The performances of different approaches are evaluated on the validation database.

We have conducted two experiments, exp1 and exp2, by varying the train and reference set. The fuzzy decision forest (F_1 for exp1 and F_2 for exp2), in both experiments, is constituted of 250 fuzzy trees. The error rate of this forest on the validation database is low: 0.038 for F_1 and 0.037 for F_2. Moreover, the error rate is stable from a forest of 225 trees.

B. Considered approaches for trees selection

Our aim is to reduce the size of a forest without increasing the error rate. We compare our approach (denoted $HQTF$) with several scenarios. We denote by S the number of subclasses of HQT found by our approach. Because the choice of the tree representing a subclass is not unique, we choose 100

TABLE I. ERROR RATE RESULTS

	exp1	*exp2*
RandomS	0.0432 ± 0.0032	0.0416 ± 0.0023
BestKappaS	0.0450	0.0390
BestErrorS	0.0441 ± 0.0021	0.0401 ± 0.0015
HQTF	$\mathbf{0.0425 \pm 0.0017}$	$\mathbf{0.0393 \pm 0.0011}$

different possibilities of forest. In exp1, $S = 15$ and in exp2 $S = 27$ with a tolerance of 0. This S is used to build different types of forest in order to evaluate our forest summarization $HQTF$ against forests of the same size. It has to be noticed that, usually, to reduce the size of a forest, no information is available regarding the number of trees.

The first approach, called RandomS, considers a forest composed of a random sample of S trees among the 250 trees. The selection is performed 100 times.

The second approach, called BestKappaS, considers a forest composed of the S best trees regarding their κ degrees.

The third approach, called BestErrorS, considers a forest composed of the S best trees regarding their error rates on the reference database. Because of the possible ties in the error rate, the selection is performed several times depending on the number of ties.

C. Analysis of the results

In table I giving error rates with associated standard deviations for all the considered approaches, we can see that $HQTF$ performs better in both experiments with the lowest standard deviation. Hence, the number S and the degree κ are important information to take into account for the summarization of a forest without loss of accuracy.

In order to analyze more precisely the benefits of considering the κ degree in the forest summarization process, it is interesting to see if there is a correlation between several indicators as the error rate of each fuzzy decision tree on the reference database, its error rate on the validation database and its κ degree on the reference database.

Figures 2a and 3a give the plot of the validation error rate vs κ. Figures 2b and 3b give the plot of the reference error rate vs κ. Figures 2c and 3c give the plot of the validation error rate vs the reference error rate. One can notice that there is no correlation between the validation error rate and κ no more than between the validation error rate and the reference error rate, while κ is quite correlated to the reference error rate.

In other words, knowing the error rate of a tree on the reference database does not enable to predict its error rate on the validation database. Similarly, knowing the κ degree of a tree on the reference database does not provide an information on its error rate on the validation database. This absence of correlation is highlighted in a more visible way on the Figures 2d and 3d: in these figures, the error rate ranks are distributed on the whole space. This observation illustrates the principle of decision forest: a set of weak learners provide a single strong learner.

Contrary to the comparison between validation database and reference database, there is a correlation between the κ of a tree and its error rate on the reference database. It is an expected result as the κ of a tree depends on its accuracy on the reference database. Nevertheless, this correlation is not so strong: it is impossible to deduce κ only with the error rate.

VI. CONCLUSION

In this paper, a new approach to reduce the size of fuzzy decision forest is proposed.

This approach is based on the use of a subclass discovery algorithm to summarize a fuzzy decision forest. Subclasses of trees are discovered by means of their quality evaluated by a degree κ that we have introduced. This degree lies on the idea that the quality of a tree should not be estimated only on its accuracy but should be balanced by the level of difficulty of the examples it has to classify.

We have shown on experimental studies, that our approach enables to determine a reduced forest without a minimal loss of performance comparing to the entire forest. Comparing to forests of the same size, our $HQTF$ performs better in a more stable way.

In future works, other experiments will be expanded to other datasets, with various sizes, in order to obtain a better view of the main advantages and drawbacks of this new algorithm. For instance, relations between the reduction of the size of the forest and the accuracy could be studied. The selection of the trees after the characterization process will be studied. We plan also to study better the aggregation of the results of the selected fuzzy decision trees in the final fuzzy decision forest. Moreover, the size of the reference database should be studied in order to observe the possible links between the quality of the summarization and the size of the reference database. A global comparison of the whole approach with other ensemble approaches will also be done on various datasets.

We think also that our HQT forest summarization can be applied to extremely randomized trees [31].

REFERENCES

[1] J. Forest, M. Rifqi, and B. Bouchon-Meunier, "Class segmentation to improve fuzzy prototype construction: Visualization and characterization of non homogeneous classes," in *IEEE World Congress on Computational Intelligence (WCCI 2006)*, 2006, pp. 555–559.

[2] P. W. Cooper, "The hypersphere in pattern recognition," *Information and control*, vol. 5, pp. 324–346, 1961.

[3] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, pp. 123–140, 1996.

[4] B. Efron and R. Tibshirani, "Improvements on cross-validation: The .632+ bootstrap method," *Journal of the American Statistical Association*, vol. 92, no. 438, pp. 548–560, 1997.

[5] E. Bauer and R. Kohavi, "An empirical comparison of voting classification algorithms: Bagging, boosting, and variants," *Machine Learning*, vol. 36, pp. 105–139, 1999.

[6] R. Schapire, "The strengh of weak learnability," *Machine Learning*, vol. 5, pp. 197–227, 1990.

[7] Y. Freund, R. Iyer, R. Schapire, and Y. Singer, "An efficient boosting algorithm for combining preferences," *Journal of Machine Learning Research*, vol. 4, pp. 933–969, 2003.

[8] Y. Freund and R. Shapire, "A short introduction to boosting," *Journal of Japanese Society for Artificial Intelligence*, vol. 14, no. 5, pp. 771–780, September 1999.

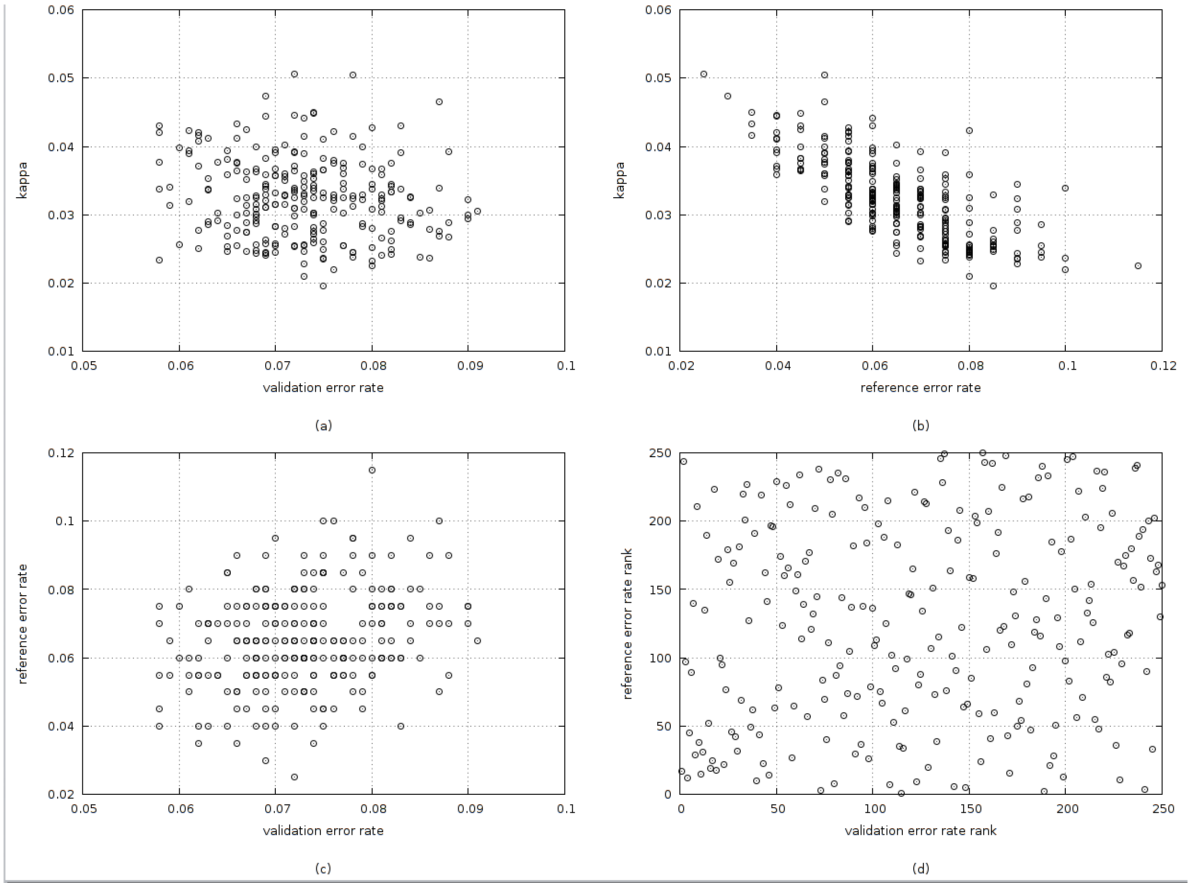


Fig. 3. Results on experiment 2

[9] ——, "Experiments with a new boosting algorithm," in *International Conference on Machine Learning*, 1996, pp. 148–156.

[10] T. K. Ho, "Random decision forests," in *Proc. of the Third Int. Conf. on Document Analysis and Recognition*, vol. 1, 1995, pp. 278–282.

[11] ——, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Learning*, vol. 20, no. 8, pp. 832–844, August 1998.

[12] T. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization," *Machine Learning*, vol. 40, pp. 139–157, 2000.

[13] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[14] C. Marsala and B. Bouchon-Meunier, "Forest of fuzzy decision trees," in *Seventh International Fuzzy Systems Association World Congress*, M. Mareš, R. Mesiar, V. Novák, J. Ramik, and A. Stupňanová, Eds., vol. 1, Prague, Czech Republic, June 1997, pp. 369–374.

[15] C. Z. Janikow and M. Faifer, "Fuzzy decision forest," in *Proceedings of the 19th International Conference of the North American Fuzzy Information Processing Society (NAFIPS'00)*, July 2000, pp. 218–221.

[16] K. Crockett, Z. Bandar, and D. Mclean, "Growing a fuzzy decision forest," in *Proceedings of the 10th IEEE International Conference on Fuzzy Systems*, December 2001, pp. 614–617.

[17] C. Z. Janikow, "Fuzzy decision forest," in *Proceedings of the 22nd International Conference of the North American Fuzzy Information Processing Society (NAFIPS'03)*, July 2003, pp. 480–483.

[18] P. P. Bonissone, J. M. Cadenas, M. C. Garrido, and R. A. Díaz-Valladares, "A fuzzy random forest: Fundamental for design and construction," in *Proceedings of the 12th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'08)*, Malaga, Spain, July 2008, pp. 1231–1238.

[19] C. Marsala, "Data mining with ensembles of fuzzy decision trees," in *Proceedings of the Symposium on Computational Intelligence and Data Mining*, Nashville, USA, March 2009, pp. 348–354.

[20] C. Marsala and M. Detyniecki, "High scale fuzzy video mining," in *Scalable Fuzzy Algorithms for Data Management and Analysis: Methods and Design*, A. Laurent and M.-J. Lesot, Eds. IGI, 2009, ch. 15, pp. 365–378.

[21] N. Labroche and C. Marsala, "Optimization of a fuzzy decision trees forest with artificial ant based clustering," in *Int. Conf. SOCPAR*, 2010.

[22] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification And Regression Trees*. New York: Chapman and Hall, 1984.

[23] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 86–106, 1986.

[24] C. Z. Janikow, "Fuzzy decision trees: Issues and methods," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 28, no. 1, pp. 1–14, 1998.

[25] C. Marsala and B. Bouchon-Meunier, "An adaptable system to construct fuzzy decision trees," in *Proc. of the NAFIPS'99 (North American Fuzzy Information Processing Society)*, New York, USA, 1999, pp. 223–227.

[26] D. L. Reilly, L. N. Cooper, and C. Elbaum, "A neural model for category learning," *Biological Cybernetics*, vol. 45, pp. 35–41, 1982.

[27] M. Marchand and J. Shawe-Taylor, "The set covering machine," *Journal of Machine Learning Research*, vol. 3, pp. 723–746, 2002.

[28] J. Wang, P. Neskovic, and L. N. Cooper, "A minimum sphere covering approach to pattern classification," in *International Conference on Pattern Recognition (ICPR)*, 2006, pp. 433–436.

[29] A. Asuncion and D. J. Newman, "UCI machine learning repository," 2007. [Online]. Available: http://www.ics.uci.edu/∼mlearn/MLRepository.html

[30] I. Guyon, S. R. Gunn, A. Ben-Hur, and G. Dror, "Result analysis of the nips 2003 feature selection challenge." 2004.

[31] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, no. 1, pp. 3–42, April 2006.