Fuzzy Random Forest with C–Fuzzy Decision Trees

Łukasz Gadomer, Zenon A. Sosnowski

Fuzzy Random Forest with C–Fuzzy Decision Trees

Łukasz Gadomer, Zenon A. Sosnowski

Faculty of Computer Science
Bialystok University of Technology
Wiejska 45A, 15-351 Bialystok, Poland
{l.gadomer,z.sosnowski}@pb.edu.pl
http://www.wi.pb.edu.pl

Abstract. In this paper a new classification solution which joins C–Fuzzy Decision Trees and Fuzzy Random Forest is proposed. Its assumptions are similar to the Fuzzy Random Forest, but instead of fuzzy trees it consists of C–Fuzzy Decision Trees. To test the proposed classifier there was performed a set of experiments. These experiments were performed using four datasets: Ionosphere, Dermatology, Pima–Diabetes and Hepatitis. Created forest was compared to C4.5 rev. 8 Decision Tree and single C–Fuzzy Decision Tree. The influence of randomness on the classification accuracy was also tested.

Keywords: Fuzzy Tree, C–Fuzzy Decision Tree, Fuzzy Random Forest

1 Introduction

Authors propose a new kind of classifier which joins C–Fuzzy Decision Trees and Fuzzy Random Forest. In this paper the construction of this ensemble classifier is presented. It is built similar to the Fuzzy Random Forest but instead of Janikow Fuzzy Trees [7] it uses C–Fuzzy Decision Trees [9]. The first part of this paper consists of all of the techniques connected with the proposed ensemble classifier: Fuzzy Decision trees, C–Fuzzy Decision Trees and Ramdom Forests. In the next part the details of Fuzzy Random Forest with C–Fuzzy Decision Trees classifier are described. After that, the experiments are described their results are presented. The quality of classification acquired using Fuzzy Random Forest with C–Fuzzy Decision Trees is compared with C–Fuzzy Decision Trees working singly. Also, the strength of randomness is tested by comparing results obtained using random node selection with the results achieved without it.

2 Related work

The idea of Fuzzy Random Forest with C–Fuzzy Decision trees is based on the Fuzzy Random Forest. Before presenting this issue it is worth to take a look at two classifiers which are the fundaments of the mentioned forest. The first one is the Fuzzy Tree, the second one is the Random Forest. Both of these issues are described in following paragraphs.

2.1 Fuzzy decision trees

Fuzzy trees are modification of traditional decision trees. The characteristic feature of this kind of trees is the fact of using fuzzy logic in their construction, learning and decision making process. There are a lot of works which concern this issue, for example [11] or [6]. One of the most popular articles about fuzzy trees is C. Z. Janikow's paper [7]. This article is described in following paragraphs.

C. Z. Janikow created fuzzy trees [7] in order to join advantages of fuzzy logic and decision trees. This kind of trees have simple, clear and intelligible knowledge structure, which is characteristic for decision trees and they can deal with noises, imprecise information etc., which is possible thanks to the fuzzy logic. It allows for using fuzzy decision trees at areas where decision trees didn't work well.

Fuzzy decision trees [7] are based on two popular decision tree creation algorithms: CART [4] and ID3 [10]. C. Z. Janikow decided to build his version of tree following assumptions of ID3, but modifying them in the way that allows for successful working with both discrete and continuous values. Proposed tree differs from the traditional one in two ways. First of them is the fact it uses the different inference procedures. The second one is about using node division criterions based on the fuzzy relations.

2.2 Random Forests

Random forest was created by L. Brieman and presented in [5]. A forest is classifier which consists of many trees. Each of these trees makes its own decision about assigning the object to the given class. After that, forest decides about the class where the object belongs, using to all of the decisions made by single trees. The thing which differs random forest from the standard one is the fact of using randomness during the tree construction process. It reduces correlation between trees with keeping the accuracy of classification. L. Briemann proposed two methods of using randomness to create random forest.

First method is based on random set of attributes selection before the node split. The nubmer of elements in this set is constant and equal for the each tree. When attributes are selected, the best candidate to divide the node is chosen. The choice is performed from the mentioned set. Brieman tested two sizes of attributes sets used to random selection. The first set's size is one, which means that from available attributes there is randomly chosen one in order to divide the node. The second set's size is the biggest number lower than $log_2 M + 1$ where M is a total number of attributes in the dataset. Tree growth is performed according to the assumptions of CART method [4], trees are not pruned. L. Brieman called the structure created the described way Forest–RI.

The second proposed method can be used when the dataset has relatively small number of features. In such situation the random choice is being made from linear combinations of attributes instead of the attributes. The structure created that way was called by its author Forest–RC.

2.3 Fuzzy random forests

The classifier with joins two solutions described in previous paragraphs was first presented in [1] and then widely described in [3] and [2]. The mentioned classifier was based on two papers cited before: [7] and [5]. Fuzzy random forest, according to its assumptions, combines the robustness of ensemble classifiers, the power of the randomness to decrease the correlation betweeen the trees and increase the diversity of them and the flexibility of fuzzy logic for dealing with imperfect data [2].

Fuzzy random forest construction process is similar to Forest–RI, described in [5]. After the forest is constructed, the algorithm begins its working from the root of each tree. First, a random set of attributes is chosen (it has the same size for each node). For each of these attributes information gain is computed, using all of the objects from training set. Attribute with the highest information gain is chosen to node split. When the node is splitted, selected attribute is removed from the set of attributes possible to select in order to divide the following nodes. Then, for all of the following tree nodes, this operation is repeated using a new set of randomly selected attributes (attributes which were used before are excluded from the selection) and the same training set.

During the tree construction process, when the node is dividing, the given object's membership degree to the given node is computed. Before the division, for each node the membership degree is 1. When the division is completed, each object can belong to any number of created leaves (at least one). If the object belongs to one leaf, its membership degree to this leaf achieves 1 (for the other leaves it is equal to 0). If it belongs to more than one leaf, the membership degree to each leaf can take values between 0 and 1 and it sums to 1 in the set of all children of the given node. If the division is performed using attribute with missing value, the object is assigned to each split node with the same membership degree.

According to described algorithm trees are constructed. Each tree is created using randomly selected set of attributes, different for each tree, which ensures diversity of trees in the forest.

Bonissone et al. proposed two fuzzy random forest decision making strategies. First of them assumpts making decisions by each tree separately – then, using achieved results, forest is making its final decision about the class where the object belongs. The second strategy is about making one common decision by the forest using all of the information collected by all of the trees. For each of these strategies authors proposed several decision making methods.

2.4 C–Fuzzy Decision Trees

In [9] W. Pedrycz and Z. A. Sosnowski proposed the new kind of decision trees, called C–Fuzzy Decision Trees. This class of trees was created in order to deal with the main problems of traditional trees. There are some fundamentals of decision trees. They usually operate on a relatively small set of discrete attributes. To split the node in the tree construction process, the single attribute which

brings the most information gain is chosen. In their traditional form decision trees are designed to operate on discrete class problems – the continuous problems are handled by regression trees. These fundamentals bring some problems. To handle continuous values it is necessary to perform the discretization. It can impact on the overall performance of the tree negatively. What is more, information bringed by the nodes which were not selected to split the node is kind of lost.

C–Fuzzy Decision Trees were developed to deal with these problems of traditional trees. The idea of this kind of trees assumed treating data as collection of information granules. These granules analogous to fuzzy clusters. Authors decided to span the proposed tree over them. The data is grouped in such multivariable granules characterized by high homogenity (low variablity) which are treated as generic building blocks of the tree.

The construction of C–Fuzzy Decision Tree starts from grouping the data set into c clusters. It is performed in the way that the similar objects should be placed in the same cluster. Each cluster is characterized by its prototype (centroid), which is randomly selected first and then improved iteratively during the tree counstruction process. When objects are grouped into clusters, the diversity of the each of these clusters is computed using the given heterogenity criterion. The computed diversity value decides if the node is selected to split or not. From all of the nodes the most heterogenous is chosen to split. The selected node is divided into c clusters using fuzzy clustering. Then, for the newly created nodes, the diversity is computed and the selection to split is performed. This algorithm works until it achieves the given stop criterion. The growth of the tree can be deep or breadth intensive. Each node of such tree has 0 or c children.

To make the paper self contained we describe the tree construction process in a formal way. Let's do the following assumptions:

- c is a number of clusters,
- $i = 1, 2, ..., c$,
- N is a number of training instances,
- $k = 1, 2, ..., N$,
- d_{ik} is a distance function between the ith prototype and the kth instance,
- m is a fuzzification factor (usually $m = 2$),
- $U = [u_{ik}]$ is a partition matrix,
- $\boldsymbol{Z} = \{\boldsymbol{x}(k), y(k)\}$ is an input–output pair of data instances,
- $\boldsymbol{z}_k = [x_1(k)x_2k...x_n(k)y(k)]^T$,
- f_i is the prototype of the cluster.

Constructing clusters and grouping objects into them is based on Fuzzy C–Means technique (FCM), which is is an example of a fuzzy clustering. Clusters are built through a minimization of objective function Q, which assumes the format:

$$Q = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^{m} d_{ik}^{2} \tag{1}$$

During the iterations of Fuzzy C–Means process partitions u_{ik} and prototypes f_i are updated. For partitions it is performed according to the following equation:

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} (\frac{d_{ik}}{d_{jk}})^{2/(m-1)}} \tag{2}$$

Prototypes are updated using the following expression:

$$f_i = \frac{\sum_{k=1}^{N} u_{ik}^m \boldsymbol{z}_k}{\sum_{k=1}^{N} u_{ik}^m} \tag{3}$$

In order to describe the node splitting criterion let's do the following assumptions:

- V_i is the variability of the data in the output space existing at the given node,
- m_i is the representative of this node positioned in the output space,
- $\boldsymbol{X}_i = \{\boldsymbol{x}(k) | u_i(\boldsymbol{x}(k)) > u_j(\boldsymbol{x}(k)) \text{ for all } j \neq i\}$, where j pertains to the nodes originating from the same parent, denotes all elements of the data set which belong to the given node in virtue of the highest membership grade,
- $\boldsymbol{Y}_i = \{y(k) | \boldsymbol{x}(k) \in \boldsymbol{X}_i\}$ collects the output coordinates of the elements that have already been assigned to $\boldsymbol{X}_i$,
- $\boldsymbol{U}_i = [u_i(\boldsymbol{x}(1)) u_i(\boldsymbol{x}(2)) ... u_i(\boldsymbol{x}(\boldsymbol{Y_i}))]$ is a vector of the grades of membership of the elements in $\boldsymbol{X}_i$,
- $\boldsymbol{N}_i =< \boldsymbol{X}_i, \boldsymbol{Y}_i, \boldsymbol{U}_i >$,

According to these notation, m_i is the following weighted sum:

$$m_i = \frac{\sum_{(\boldsymbol{x}(k), y(k)) \in \boldsymbol{X}_i \times \boldsymbol{Y}_i} u_i(\boldsymbol{x}(k)) y(k)}{\sum_{(\boldsymbol{x}(k), y(k)) \in \boldsymbol{X}_i \times \boldsymbol{Y}_i} u_i(\boldsymbol{x}(k))} \tag{4}$$

The variability is computed as follows:

$$V_i = \sum_{(\boldsymbol{x}(k), y(k)) \in \boldsymbol{X}_i \times \boldsymbol{Y}_i} u_i(\boldsymbol{x}(k)) (y(k) - m_i)^2 \tag{5}$$

The tree growth stop criterion could be, for example, defined in the following way: [9]

- There aren't enough elements in any node to perform the split. The minimal number of elements in the node which allows for the split is c. Normally the boundary number of the elements in each node would be the multiplicity of c, for example $2 \times c$ or $3 \times c$,

- All nodes achieve lower diversity than assumed boundary value,
- The structurability index achieves the lower value than assumed boundary value,
- The number of iterations (splits) achieved the assumed boundary value.

When the tree is constructed it can be used for classification. Each object which has to be classified starts from the root node. The membership degrees of this object to the children of the given node are computed. These membership degrees are the numbers between 0 and 1 and they sum to 1. The node where the object belongs with the highest membership is chosen and the object is getting there. The same operation is repeated as long as the object achieves to the node which has no children. The classification result is the class assigned to achieved node.

3 Fuzzy Random Forest with C–Fuzzy Decision Trees

To describe created classifier we used the following notations (based on [2] and [9]):

- T is the number of trees in the C–FRF ensemble,
- t is the particular tree,
- N_t is the number of nodes in the tree t,
- n is a particular leaf reached in a tree,
- I is the number of classes,
- i is a particular class,
- C_FRF is a matrix with size $(T \times MAX_{N_t})$ with $MAX_{N_t} = max\{N_1, N_2, ..., N_t\}$, where each element of the matrix is a vector of size I containing the support for every class provided by every activated leaf n on each tree t; this matrix represents C–Fuzzy Forest or Fuzzy Random Forest with C–Fuzzy Decision Trees,
- c is the number of clusters,
- E is a training dataset,
- e is a data instance,
- $V = [V_1, V_2, ..., V_b]$ is the variability vector.
- $U = [U_1, U_2, ..., U_{|E|}]$ is the tree's partition matrix of the training objects,
- $U_i = [u_1, u_2, ..., u_c]$ are memberships of the ith object to the c cluster,
- $B = \{B_1, B_2, ..., B_b\}$ are the unsplitted nodes,

We propose creating the new kind of classifiers: Fuzzy Random Forest with C–Fuzzy Decision Trees. It is the forest, based on the idea of Fuzzy Random Forest, which consists of C–Fuzzy Decision Trees. The Fuzzy Random Forest uses randomness to improve the classification quality while C–Fuzzy Decision Tree is constructed randomly by definition – centroids of its clusters (the partition matrix) are selected randomly first. Combination these two structures is expected to give promising results.

The randomness in Fuzzy Random Forest with C–Fuzzy Decision Trees is ensured by two main aspects. The first of them refers to the Random Forest.

During the tree's construction process, node to split is selected randomly. This randomness can be full, which means selecting the random node to split instead of the most heterogenous, or limited, which assumpts selecting the set of nodes with the highest diversity, then randomly selecting one of them to perform the division (the size of the set is given and the same for the each split). The second aspect refers to the C–Fuzzy Decision Trees and it concerns the creation of partition matrix. At first, the centroid (prototype) of the each cluster selection is fully random. Objects which belong to the parent node are divided into clusters grouped around these centroids using the shortest distance criterion. Then the prototypes and the partition matrix are being corrected as long as they achieve the stop criterion. Each tree in the forest, created the described way, can be selected from the set of created trees. To create the single tree which will be chosen to the forest there can be build the set of trees. Each tree from such set is tested and the best of these trees (the one which achieved the best classification accuracy for the training set) is being chosen as the part of forest. The size of the set is given and the same for the each tree in the forest.

The split selection idea is similar to the one used in Fuzzy Random Forest. The difference is about the nature of tree used in the classifier. In Fuzzy Random Forest, the random attribute was being chosen to split. The node which was chosen to split was specified by tree growth strategy. In Fuzzy Random Forest with C–Fuzzy Decision Trees there isn't any attribute chosen – for each of the splits all of the attributes are considered. The choice concerns the node to split selection which means some nodes does not have to be splitted (when the stop criterion is achieved). The same idea is expressed in two completely different ways of building trees. Each C–Fuzzy Decision Tree in the forest can be completely different or very similar – it depends on the stop criterion and the number of clusters. The influence of randomness can be set using algorithm parameters which allows classifier to fit the given problem in a flexible way.

Prototypes of each cluster are selected randomly and then corrected iteratively, which means some of created trees can work better than others. Diversity of trees created that way depends on the number of the correction process' iterations. It is possible to build many trees and choose only the best of them to the forest in order to achieve better results. The diversity of the trees in the forest can be modified by changing the size of the set from which the best tree is chosen and the number of iterations. These parameters specifiy the strength of randomness in the classifier. Operating on these values also allow to fit to the given problem to improve the classification quality.

3.1 Fuzzy Random Forest with C–Fuzzy Decision Trees learning

The process of Fuzzy Random Forest with C–Fuzzy Decision Trees learning is analogous to the learning of Fuzzy Random Forest, proposed in [2]. The differences concern two aspects. First is about the kind of trees used in the forest. In the proposed classifier there are used C–Fuzzy Decision Trees instead of Janikow's Fuzzy Trees. The second aspect refers to the way of random selection of the node to split, which was described before.

The Fuzzy Random Forest with C–Fuzzy Decision Trees is created using Algorithm 1.

Algorithm 1 Fuzzy Random Forest with C–Fuzzy Decision Trees learning

procedure FRFwC–FDTlearning
  for 1 to T **do**
    1. Take a random sample of $|E|$ examples with replacement from the dataset E
    2. Apply Algorithm 2 to the subset of examples obtained in the previous step to construct C–Fuzzy Decision Tree
  end for
end procedure

Each tree in Fuzzy Random Forest with C–Fuzzy Decision Trees is created using Algorithm 2.

Algorithm 2 C–Fuzzy Decision Tree learning

procedure C–FDTlearning
  1. Start with the examples in E
  2. Create the partition matrix U randomly
  3. Perform FCM
  while Stop criterion is not satisfied **do**
    4. Divide the samples belonging to the splitted node into its children
    5. Make a random selection of nodes from the set of unsplitted nodes B
    6. Compute the variability matrix V
    7. Choose the node with maximum variability to split nodes
    8. Perform FCM
  end while
end procedure

3.2 Fuzzy Random Forest with C–Fuzzy Decision Trees classification

After the Fuzzy Random Forest with C–Fuzzy Decision Trees is constructed it can be used for new object's classification. The decision–making strategy used in the proposed sollution assumpts making decision by forest after each tree's decisions are made.[1]. It is performed according to the algorithm 3. It can be described by equation, similar to the one presented in [2]:

[1] There is also another decision–making strategy which assumpts making the single decision by the whole forest. It is described in [2]

$$D_F RF(t,i,C\_FRF) = \begin{cases} 1 \text{ if i} = arg \max\limits_{j,j=1,2,\ldots,I} \left\{ \sum\limits_{n=1}^{N_t} C\_FRF_{t,n,j} \right\} \\ 0 \text{ otherwise} \end{cases}$$

Algorithm 3 Fuzzy Random Forest with C–Fuzzy Decision Trees classification

```
1: procedure FRFwC–FDTclassification
2:     DecisionOfTrees
3:     DecisionOfForest
4: end procedure
5: procedure DecisionOfTrees
6:     for 1 to T do
7:         1. Run the example e to obtain the tree's partition matrix U_i
8:         2. Choose the class c where c = arg max_{i,i=1,2,...,I} D_FRF_{t,i,C_FRF}
9:     end for
10: end procedure
11: procedure DecisionOfForest
12:     Assign to class according to the simple majority vote of trees decisions
13: end procedure
```

4 Experimental studies

To test a quality of created classifier there were performed several experiments. These experiments were performed on four popular datasets from UCI Machine Learning Repository [8]:

- Ionosphere,
- Dermatology,
- Pima–Diabetes,
- Hepatitis.

Each dataset was divided into five parts with equal size (or as close to the equal as it's possible) randomly. Each of these parts had the same proportions of objects representing each decision class as it is in the whole dataset (or as close to the same as it's possible). There were no situations when in some of parts there weren't any objects representing some of decision classes. This random and proportional division was saved and used for each experiment.

Each experiment was performed using 5–fold crossvalidation. Four of five parts were used to train the classifier, one to test the learned forest. This operation was repeated five times, each time the other part was excluded from

training and used for testing the classifier. After that, classification accuracy of all five out of bag parts were averaged.

For each dataset there were performed researches for both Fuzzy Forest with C–Fuzzy Decision Trees and Fuzzy Random Forest with C–Fuzzy Decision Trees. For each of these configurations there were performed expermients with 2, 3, 5, 8, 13 and 20 clusters. Each forest were consisting of 50 trees.

The objective of the research is to test how randomness influences the classification accuracy of the forest. There is also checked how the classification accuracy changes with the different number of clusters.

All of the results are presented in paragraph 5. They are all compared with themselves and also with single C–Fuzzy Decision Tree and C4.5 rev. 8 tree.

5 Results and discussion

Classification accuracies part consists of the following information:

- Classification accuracy achieved using C4.5 rev. 8 Decision Tree,
- Classification accuracy achieved using a single C–Fuzzy Decision Tree,
- Classification accuracies achieved using Fuzzy Forest with C–Fuzzy Decision Trees,
- Classification accuracies achieved using Fuzzy Random Forest with C–Fuzzy Decision Trees.

The results for tested datasets are presented in the following tables:

- Ionosphere – Table 1,
- Dermatology – Table 2,
- Pima–Diabetes – Table 3,
- Hepatitis – Table 4.

Number of clusters	2	3	5	8	13	20
C4.5 rev. 8	13,54					
Single C–Fuzzy Tree	15,1	15,68	14,82	17,11	38,71	36,17
Fuzzy Forest with C–Fuzzy Decision Trees	15,1	15,39	13,39	15,67	29,93	30,19
Fuzzy Random Forest with C–Fuzzy Decision Trees	14,25	14,82	12,24	13,96	26,2	25,92

Table 1. Results – Ionosphere

Number of clusters	2	3	5	8	13	20
C4.5 rev. 8	5,98					
Single C–Fuzzy Tree	6,02	6,3	5,47	17,46	39,36	48,42
Fuzzy Forest with C–Fuzzy Decision Trees	6,02	6,3	2,99	6,01	34,64	38,82
Fuzzy Random Forest with C–Fuzzy Decision Trees	6,02	6,3	2,18	5,17	31,86	34,65

Table 2. Results – Dermatology

Number of clusters	2	3	5	8	13	20
C4.5 rev. 8	27,95					
Single C–Fuzzy Tree	30,08	30,21	27,08	27,74	32,23	31,78
Fuzzy Forest with C–Fuzzy Decision Trees	28,52	29,68	26,31	27,47	29,03	29,43
Fuzzy Random Forest with C–Fuzzy Decision Trees	28,65	29,16	26,96	26,56	28,52	30,07

Table 3. Results – Pima–Diabetes

Number of clusters	2	3	5	8	13	20
C4.5 rev. 8	43,86					
Single C–Fuzzy Tree	34,84	41,93	37,42	41,29	44,51	51,61
Fuzzy Forest with C–Fuzzy Decision Trees	34,19	38,06	37,42	36,77	34,19	43,87
Fuzzy Random Forest with C–Fuzzy Decision Trees	34,19	37,42	36,13	38,06	41,29	40

Table 4. Results – Hepatitis

The general tendention that can be observed in all of those results is decreasing the classification accuracy with increasing the number of clusters. In most cases results were the best for around 5 clusters, a bit worse for 2 – 3 clusters and significally worse for 13 – 20 clusters. This dependence is a bit different for each dataset, which means the number of clusters should be chosen according to the given problem.

In most cases Fuzzy Random Forests with Fuzzy Decision Trees achieved better results than Fuzzy Forests with Fuzzy Decision Trees. The exception was Pima–Diabetes dataset, where at the same number of cases the results were better or worse, depending on the number of clusters. It means that randomness generally increased the classification accuracy.

For each dataset there was at least one number of clusters which in almost all cases allowed to achieve better result that C4.5 rev. 8 Decision Tree. For Hepatitis dataset the better result was achieved independently from the number of clusters (exception was 20 clusters), but for the rest of datasets there were only one or two number of clusters, where these results were better. It shows how important is the choice of the proper number of clusters for C–Fuzzy Decision Trees used in Fuzzy Forests with Fuzzy Decision Trees and Fuzzy Random Forests with Fuzzy Decision Trees.

In almost all of the cases (unusual exceptions) results achieved using Fuzzy Forest with Fuzzy Decision Trees and Fuzzy Random Forest with Fuzzy Decision Trees were better than using single C–Fuzzy Decision Tree. It clearly shows the strength ensemble classifier build of this kind of trees. They achieve much better results when working together that when working as a single classifier.

6 Conclusion

In the previous paragraphs of this article there was proposed Fuzzy Random Forest with Fuzzy Decision Trees classifier. The created solution was checked on four datasets. Ionosphere, Dermatology, Pima–Diabetes andHepatitis. There were tested how successfully the classifier works in comparison to the C4.5 rev. 8 Decision Tree and a single C–Fuzzy Decision Tree. There were also tested how randomness affects achieved classification quality.

Performed experiments proved than in most cases Fuzzy Random Forest with C–Fuzzy Decision Trees classifier gives better results than C4.5 rev. 8 Decision Tree and single C–Fuzzy Decision Tree classifiers. They also demonstrated that using randomness in the forest can increase the classification quality.

Acknowledgment

This work was supported by the grant S/WI/1/2013 from Bialystok University of Technology founded by Ministry of Science and Higher Education.

References

1. P. P. Bonissone, J. M. Cadenas, M. C. Garrido, and R. A. Diaz-valladares. A fuzzy random forest: Fundamental for design and construction. In *In Proceedings of the 12th International Conference on Information Processing and Management of Uncertainty in Knowledge- Based Systems (IPMU08)*, pages 1231–1238, 2008.
2. Piero Bonissone, Jose M. Cadenas, M. Carmen Garrido, and R. Andres Diaz-Valladares. A fuzzy random forest. *International Journal of Approximate Reasoning*, 51(7):729 – 747, 2010.
3. P.P. Bonissone, J.M. Cadenas, M.C. Garrido, and R.A. Diaz-Valladares. Combination methods in a fuzzy random forest. In *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*, pages 1794–1799, Oct 2008.
4. L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984.
5. Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, October 2001.
6. Robin L.P. Chang and Theodosios Pavlidis. Fuzzy decision tree algorithms. *Systems, Man and Cybernetics, IEEE Transactions on*, 7(1):28–35, Jan 1977.
7. C.Z. Janikow. Fuzzy decision trees: issues and methods. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 28(1):1–14, Feb 1998.
8. M. Lichman. UCI machine learning repository, 2013.
9. W. Pedrycz and Z.A. Sosnowski. C-fuzzy decision trees. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 35(4):498–511, Nov 2005.
10. J. R. Quinlan. Induction of decision trees. *Mach. Learn.*, 1(1):81–106, March 1986.
11. Yufei Yuan and Michael J. Shaw. Induction of fuzzy decision trees. *Fuzzy Sets Syst.*, 69(2):125–139, January 1995.