Using Cluster–Context Fuzzy Decision Trees in Fuzzy Random Forest

Łukasz Gadomer[(✉)] and 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. Cluster–Context Fuzzy Decision Tree is the classifier which joins C–Fuzzy Decision Tree with Context–Based Fuzzy Clustering method. The idea of using this kind of tree in the Fuzzy Random Forest is presented in this paper. The created ensemble classifier has similar assumptions to the Fuzzy Random Forest, but differs in the kind of used trees and all aspects connected with this difference. The quality of the created classifier was evaluated by several experiments performed on different datasets. There were tested both datasets with discrete and continuous attributes and decision classes. The aspect of using a randomness in the created classifier was also evaluated.

Keywords: Context–Cluster Fuzzy Decision Tree · C–Fuzzy Decision Tree · Context–Based Fuzzy Clustering · Fuzzy Random Forest

1 Introduction

The classification and regression are popular problems in data science. There were created many solutions in order to deal with these issues. In this paper authors present their innovative ensemble classifier which was designed in order to meet both these problems. The idea of this solution is creating a classifier which works with the similar assumptions to Fuzzy Random Forest [1] but instead of Janikow Fuzzy Trees [2] it uses Cluster–Context Fuzzy Decision Trees [3]. This kind of tree connects the Context–Based Fuzzy Clustering [4,5] with C–Fuzzy Decision Tree [6]. The objective of this paper was to prove that the created classifier can successfully deal with both regression and classification problems.

The first part of this paper treats about theoretical aspects of the created classifier. The theory about Context–Based Fuzzy Clustering [4,5], C–Fuzzy Decision Trees [6], Context–Cluster Fuzzy Decision Trees [3] and C–Fuzzy Random Forest [1] is described there. After that, the idea of using Cluster–Context Fuzzy Decision Trees in Fuzzy Random Forest is presented. Then, the performed experiments with the achieved results are shown. The results achieved by created trees

grouped into the forest were compared with the ones obtained with the trees working singly. The influence of using randomness during the tree construction process on the achieved results is also tested. The quality of the created solution is evaluated on the different datasets, containing continuous and discrete attributes, both for classification and regression problems.

2   Notation

In this paper we used the following notations (based on [1,6]):

– K is the number of contexts,
– k is a particular context
– T is the number of tree groups in the ensemble; in each tree group there are K trees connected with the contextes,
– t is the particular tree group,
– t_k is a particular tree in t group
– N_t is the number of nodes in the tree group t_k,
– n is a particular leaf reached in a tree group t_k,
– I is the number of classes,
– i is a particular class,
– C is the number of clusters,
– c is a particular cluster,
– E is a training dataset,
– e is a data instance,
– $V_k = [V_{1k}, V_{2k}, ..., V_{bk}]$ is the variability vector for k context,
– $U_k = [U_{1k}, U_{2k}, ..., U_{|E|k}]$ is the tree's partition matrix of the training objects for k context,
– $U_{ik} = [u_{1k}, u_{2k}, ..., u_{Ck}]$ are memberships of the ith object to the c cluster for k context,
– $B = \{B_1, B_2, ..., B_b\}$ are the unsplitted nodes,
– S is the number of objects from the dataset for which the membership function value is greater than 0 for t_k tree in k context,
– s is the particular object,
– $X = [X_1, X_2, ..., X_K]$ is the vector of objects from the dataset for which the membership function value is greater than 0 for whole tree t,
– $X_k = [X_{1k}, X_{2k}, ..., X_{Ck}]$ is the vector of objects from the dataset for which the membership function value is greater than 0 for c cluster,
– $X_{ck} = [x_1, x_2, ..., x_S, y_k]$ is the vector of objects from the dataset for which the membership function value is greater than 0 for c cluster for k context.

3   Related Work

3.1   C–Fuzzy Decision Trees

C–Fuzzy Decision Trees are the kind of trees proposed by W. Pedrycz and Z.A. Sosnowski in [6]. The main motivation to create these trees was the awareness

of problems and limits of traditional decision trees, which usually operate on a relatively small set of discrete attributes, choose the single attribute which brings the most information gain to split the node during the tree construction process and are designed to operate on discrete class problems (in their traditional form – the continuous problems are handled by regression trees). The creating of C–Fuzzy Decision Trees was intended to be a solution of these problems. According to their assumptions, C–Fuzzy Decision Trees treat data as collection of information granules, analogous to fuzzy clusters. These granules are generic building blocks of the tree – the data is grouped in such multivariable granules characterized by high homogenity (low variablity).

The first step of C–Fuzzy Decision Tree construction process is grouping the data set into c clusters. It is performed in the way that the similar objects are placed in the same cluster. Each cluster is characterized by its centroid, called prototype, which is randomly selected first and then improved iteratively. After the grouping objects into clusters is finished, the given heterogenity criterion is used to compute the diversity of the each of these clusters. This value decides if the node is selected to split or not. From all of the nodes the one with the lowest diversity value is chosen to split. This node is divided into c clusters using fuzzy clustering method [7]. After that, for each node created that way, the diversity is computed and the selection to split is performed. These steps are repeated until the algorithm achieves the given stop criterion. Each node of the tree has 0 or c children. The growth of the tree can be breadth or deep intensive.

The tree growth stop criterion could be, for example, defined in the following way: [6]

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

After the tree is constructed it can be used in classification mode. Each object which has to be classified starts from the root node. The membership degrees (numbers between 0 and 1 which sums to 1 for the node's children) of this object to the children of the given node are computed. The object gets to the node where he belongs with the highest membership among the computed ones. The same operation is repeated until the object achieves to the node which has no children. The classification result is the class assigned to this node.

3.2   Context–Based Fuzzy Clustering

Clustering is a tool used for data analysis which purpose is to find structures (groups) in multivariable datasets. The idea of context–based clustering [4] is to search such groups of data with applying the context. The context is a kind of information granule, defined in a decision attribute, using which the search for

structure in the data is focused. The general task of clustering, formulated as *reveal a structure in data* $\boldsymbol{X}$, with context–based clustering is reformulated as *reveal a structure in data* $\boldsymbol{X}$ *in context* A, where A is the information granule of interest (context of clustering).

The conditioning aspect (context sensitivity) of the clustering mechanism is used in the algorithm by taking into consideration the conditioning variable (context) assuming the values $f_1, f_2, ..., f_N$ on the corresponding patterns. In other words, f_k is the level of involvement of $\boldsymbol{x}_k$ in the considered context, $f_k = A(\boldsymbol{x}_k)$. f_k can be connected with computed membership values of $\boldsymbol{x}_k$, say $u_{1k}, u_{2k}, ..., u_{Ck}$ the way expressed in the following formula:

$$\sum_{i=1}^{c} u_{ik} = f_k, k = 1, 2, ..., N \tag{1}$$

It is important that the selected context directly impacts the resulting data to be considered. The finite support of context A does not take into consideration these data points which the membership values are equal to zero. It means only a certain subset of the original data to be used for further clustering. Considering this fact, the partition matrix U, previously defined as

$$U = \left\{ u_{ik} \in [0,1] \,\middle|\, \sum_{i=1}^{c} u_{ik} = 1 \text{ and } 0 < \sum_{k=1}^{N} u_{ik} < N \text{ for all } i = 1, 2, ..., c \right\} \tag{2}$$

can be modified into the family

$$U(A) = \left\{ u_{ik} \in [0,1] \,\middle|\, \sum_{i=1}^{c} u_{ik} = f_k \forall_k \text{ and } 0 < \sum_{k=1}^{N} u_{ik} < N \forall_i \right\} \tag{3}$$

The overall Context–Based Fuzzy Clustering algorithm can be summarized as the following sequence of steps (the number of clusters c is given):

1. Select the termination criterion ε ($\varepsilon > 0$), distance function $||\cdot||$, fuzzification parameter m (by default $m = 2.0$), then initialize the partition matrix $U \in U$.
2. Calculate prototypes (centers) of the clusters the same way as in standard FCM algorithm [7]:

$$v_i = \frac{\sum_{k=1}^{N} u_{ik}^{m} \boldsymbol{x}_k}{\sum_{k=1}^{N} u_{ik}^{m}}, i = 1, 2, ..., c \tag{4}$$

3. Update partition matrix

$$u_{ik} = \frac{f_k}{\sum_{j=1}^{c} \left(\frac{||\boldsymbol{x}_k - \boldsymbol{v}_i||}{||\boldsymbol{x}_k - \boldsymbol{v}_j||} \right)^{\frac{2}{m-1}}}, i = 1, 2, ..., c, j = 1, 2, ..., N \tag{5}$$

4. Compare U' to U. If $||U' - U|| < \varepsilon$, then stop, else return to step (2) and proceed with computing by setting up U equal to U'

Used distance function is the weighted Euclidean distance function, defined as follows:

$$||\boldsymbol{a} - \boldsymbol{b}|| = \sum_{i=1}^{n} \frac{(a_i - b_i)^2}{\sigma_i^2} \tag{6}$$

where σ_i are standard deviations of the corresponding attributes.

3.3   Cluster–Context Fuzzy Decision Trees

The main idea of Cluster–Context Fuzzy Decision Trees is joining Context–Based Clustering, presented in Sect. 3.2 and C–Fuzzy Decision Trees, described in Sect. 3.1. This kind of trees were presented in [3]. Author predicted that joining these two algorithms allows to achieve better results than C–Fuzzy Decision Trees, especially for regression problem.

It is important to notice that notation presented in Sect. 2 each Cluster–Context Fuzzy Decision Tree t consists of k C–Fuzzy Decision Trees t_k. It means that the structure called "tree" t when writing about Cluster–Context Fuzzy Decision Trees refers to the group of C–Fuzzy Decision Trees t_k, not a single tree. As it can be confusing, it is worth to remember about it.

The first thing which should be done before starting construction of Cluster–Context Fuzzy Decision Tree is dividing decision attribute into contexts. The number of contexts is the algorithm parameter which should be adjusted according to the dataset. In the theoretically perfect situation the number of contexts should respond the number of object groups in the dataset. The division can be performed using any membership function. In this research three membership functions were chosen: gaussian, trapezoidal and triangular. The example division result using these functions into five contexts is presented in Fig. 1.
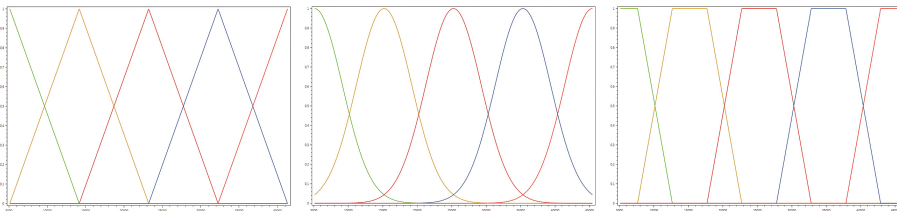
Fig. 1. Example division of decision attribute into five contexts using (from left) triangular, gaussian and trapezoidal membership functions.

In order to fit the division to the given problem in the best possible way it is also possible to configure the shape of membership function. In the created solution authors allowed to do this using "context configuration" parameter – its default value is 1, lower numbers makes contexts wider, higher numbers – shorter. Figure 1 showed divisions for default context configuration value, on Fig. 2 the example divisions using values 0.6 and 1.6 are presented.
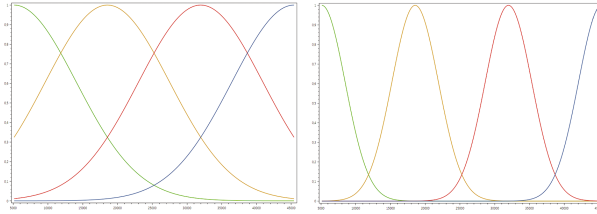
Fig. 2. Example division of decision attribute into four contexts using gaussian function with context configuration value (from left) 0.6 and 1.6.

Algorithm 1. Cluster–Context Fuzzy Decision Tree learning

1: **procedure** CC–FDTLearning
2:     **for** 1 to K **do**
3:         1. Start with the examples in X_k
4:         2. Create C–Fuzzy Decision Tree for context k using Algorithm 2
5:     **end for**
6: **end procedure**

When contexts are prepared, it is possible to create and learn Cluster–Context Fuzzy Decision Trees according to the Algorithm 1.

Each C–Fuzzy Decision Tree in Cluster–Context Fuzzy Decision Tree is created using Algorithm 2.

Algorithm 2. C–Fuzzy Decision Tree learning

1: **procedure** C–FDTLearning
2:     1. Start with the examples in X_k for k context,
3:     2. Create the partition matrix U_k randomly
4:     3. Perform FCM
5:     **while** Stop criterion is not satisfied **do**
6:         4. Divide the samples belonging to the splitted node into its children
7:         5. Make a random selection of nodes from the set of unsplitted nodes B
8:         6. Compute the variability matrix V_k
9:         7. Choose the node with maximum variability to split nodes
10:         8. Perform FCM
11:     **end while**
12: **end procedure**

Created Cluster–Context Fuzzy Decision Tree can be used in classification and regression process. It is performed according to the Algorithm 3.

3.4 Fuzzy Random Forests

The Fuzzy Random Forest classifier was first presented in [8] and then widely described in [1,9]. The mentioned classifier was based on two papers cited

Algorithm 3. Cluster–Context Fuzzy Decision Trees classification

---

1: **procedure** CC–FDTCLASSIFICATION
2:    **for** 1 to K **do**
3:        1. Run the example x_k to obtain the tree's partition matrix U_{ik}
4:        2. Choose the class c where $c = arg \max\limits_{i,i=1,2,...,I} U_{ik}$ (classification) or choose
     the node n where $n = \max U_{ik}$ (regression)
5:    **end for**
6:    Assign to class with the higher U_{ik} value from k achieved values (classification)
     or compute the distance between the y_k and the decision value in the chosen node
     n (regression)
7: **end procedure**

---

before: [2,10]. Fuzzy random forest, according to its assumptions, combines the robustness of ensemble classifiers, the power of the randomness to decrease the correlation between the trees and increase the diversity of them and the flexibility of fuzzy logic for dealing with imperfect data [1].

Fuzzy random forest construction process is similar to Forest–RI, described in [10]. 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.

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.

4   Using Cluster–Context Fuzzy Decision Trees in Fuzzy Random Forest

The classifier which uses Cluster–Context Fuzzy Decision Trees in Fuzzy Random Forest is proposed in this section. This ensemble classifier bases on the idea of Fuzzy Random Forest and uses Cluster–Context Fuzzy Decision Trees as constituent classifiers. The idea of Fuzzy Random Forest with C–Fuzzy Decision Trees was presented in [11]. It is expected that introducing the contexts with their advantages into the forest will increase the classification accuracy, especially in the area of continuous decision class problems.

The randomness in the created classifier is ensured by two main aspects. The first of them refers to the assumptions of Random Forest. When the tree is being constructed, the node to split is selected randomly. It can be full randomness (selecting the random node to split instead of the most heterogenous) or limited

(selecting the set of nodes with the highest diversity, then randomly selecting one of them to perform the split). The second aspect refers to the C–Fuzzy Decision Trees, which take part in Cluster–Context Fuzzy Decision Trees, and it concerns partition matrix creation process. The first coordinates of centroids (prototypes) of each clusters are selected randomly. Objects which belong to the parent node are divided into clusters grouped around these prototypes using the shortest distance criterion. After that the prototypes and the partition matrix are being corrected iteratively until they achieve the stop criterion. What is more, each tree in the forest can be selected from the set of created trees. Each tree from such set is tested and the best of them is being chosen as the part of forest. The size of this 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, but it refers to nodes instead of attributes. In Fuzzy Random Forest, the random attribute was being chosen to split during the Fuzzy Trees construction process. In Fuzzy Random Forest with Cluster–Context Decision Trees the choice concerns the node to split selection. Some nodes does not have to be splitted (it can happen when the stop criterion is achieved). Each Cluster–Context Fuzzy Decision Tree in the forest can be similar or different – it depends on the chosen algorithm parameters. It allows to adjust the classifier to the given problem in a flexible way.

The Fuzzy Random Forest with Cluster–Context Fuzzy Decision Trees is created using Algorithm 4.

Algorithm 4. Fuzzy Random Forest with Cluster–Context Fuzzy Decision Trees learning

---

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

The constructed forest can be used for classification and regression problems. For the classification issue, the decision–making strategy assumpts making the final decision by forest after individual decisions of trees are made. This process is performed according to the Algorithm 5.

The weighted averaging is performed to let the best trees in the forest (the ones which during the learning process achieved the lowest prediction error) have the biggest influence on the final prediction. It is performed according to the following formula:

Algorithm 5. Fuzzy Random Forest with Cluster–Context Fuzzy Decision Trees classification

---

1: **procedure** FRFwCC–FDT CLASSIFICATION
2:    Perform tree's individual decisions according to Algorithm 3
3:    Assign to class according to the simple majority vote of trees decisions (classification) or perform weighted averaging the final prediction (regression)
4: **end procedure**

$$result = \frac{\sum_{i=1}^{T} result_i \times result_{T-i}}{\frac{n(n+1)}{2}}, \forall i \ result_i < result_{i+1} \qquad (7)$$

5   Experimental Studies

The main objectives of performed experiments were to test the quality of created classifier on classification and regression process and to check the influence of the randomness on the results. To check the classification process, experiments were performed on four popular datasets from UCI Machine Learning Repository [12]: Ionosphere, Dermatology, Pima–Diabetes and Hepatitis. To check the regression process another two datasets from this repository were used: Automobile Data and Housing.

Each dataset was randomly divided into five equal parts (or as close to the equal as it's possible). For classification problems, each of these parts had the same proportions of objects with each decision class (or as close to the same as it's possible). This random and proportional division was saved and used for each experiment.

Each experiment was performed with 5–fold crossvalidation. Four of five parts were used to train the forest, one to test it. This operation was repeated five times, each time the other part was excluded from the training process and used for evaluation. At the end all classification accuracies of the out of bag parts were averaged.

Classification parameters were chosen individually for each dataset with parameter optimization process. There were tested multiple combinations of parameters for forests which consist of five trees. According to these results the classification parameter combinations for the final classifier were chosen. Each forest in the experiment consist of fifty trees.

Achieved results were compared with Fuzzy Random Forest with C–Fuzzy Decision Trees and Fuzzy Random Forest C–Fuzzy Decision Trees. Some of the results which serve as a base to compare were presented in our previous work: [11].

All of the results are presented in Sect. 6.

6   Results and Discussion

The classifiers' parameters chosen for each dataset are presented in Table 1.

Table 1. Classifier parameters for datasets

Dataset	Hepatitis	Dermatology	Pima Indians Diabetes	Ionosphere	Auto data	Housing
Number of clusters	2	2	2	2	2	3
Number of contexts	3	2	3	2	9	2
Membership function	Gaussian	Triangular	Gaussian	Gaussian	Gaussian	Gaussian
Context configuration	0.6	1.6	0.6	0.6	0.6	0.6

6.1   Datasets with Continuous Decision Attribute

The results achieved for datasets with continuous decision attribute are presented in Table 2. Numbers in the table are the average distances between the predicted value and the original value.

As it was expected, ensemble classifiers which use Cluster–Context Fuzzy Decision Trees allowed to achieve better results that forests with C–Fuzzy Decision Trees. The differences were more significant for Auto Data dataset. For housing dataset all results achieved using ensembles with Cluster–Context Fuzzy Decision Trees were also better that using forests with C–Fuzzy Decision Trees, but the differences were smaller.

For each of created ensemble classifiers: Fuzzy Forest with C–Fuzzy Decision Trees, Fuzzy Random Forest with C–Fuzzy Decision Trees, Fuzzy Forest with Cluster–Context Fuzzy Decision Trees and Fuzzy Random Forest with Cluster–Context Fuzzy Decision Trees results achieved using weights were better that without using them. It clearly means that it is worth to use weighted average instead of arithmetic average. The arithmetic average of the results achieved by trees which are part of the forest does not allow to use the full strength of the ensemble. The weighted average allows forest to work the way it was designed for.

For the ensemble which use C–Fuzzy Decision Trees in most cases using randomness increased the quality of the classifier. The difference was more significant for Auto Data dataset. For Housing dataset the differences between using randomness and not were slight. It means that using randomness can improve the classification accuracy, but it depends on the dataset and the classifier's configuration.

Table 2. Results for datasets with continuous class attribute

Dataset	Auto data	Housing
Fuzzy Forest with C–Fuzzy Decision Trees without weights	3277.55	3.98
Fuzzy Forest with C–Fuzzy Decision Trees with weights	3244	3.92
Fuzzy Random Forest with C–Fuzzy Decision Trees without weights	3192.9	3.97
Fuzzy Random Forest with C–Fuzzy Decision Trees with weights	3178.26	3.94
Fuzzy Forest with Cluster–Context Fuzzy Decision Trees without weights	3064.58	3.91
Fuzzy Forest with Cluster–Context Fuzzy Decision Trees with weights	3025.41	3.86
Fuzzy Random Forest with Cluster–Context Fuzzy Decision Trees without weights	3023.92	3.87
Fuzzy Random Forest with Cluster–Context Fuzzy Decision Trees with weights	3015.59	3.85

6.2 Datasets with Discrete Decision Class

The results achieved for datasets with discrete decision class are presented in Table 3.

In most cases using forest with Cluster–Context Fuzzy Decision Trees allowed to achieve better results that with ensemble using C–Fuzzy Decision Trees. Using randomness in Fuzzy Random Forest with Cluster–Context Fuzzy Decision Trees in most cases also allowed to improve classification accuracy (for Dermatology dataset it was the same, for the other ones it was better). All of these improvements were small but noticeable.

It is worth to notice the reason classification of the accuracy improvement in the given classifier configuration. On the sample visualization of contexts presented in Fig. 2 it is showed that for the context configuration parameter equal to 0.6 all of the contexts are relative wide. For the decision attribute division into two or three contexts all of these contexts, for all of the decision attribute's values, has a value greater than zero. It means for each context no objects from the dataset are excluded from the C–Fuzzy Decision Trees which are part of Cluster–Context Decision Tree. In this case the ensemble classifier with Cluster–Context Fuzzy Decision Tree works the similar way to the forest with C–Fuzzy Decision Trees, but each tree is chosen from the best of K. In this special case choosing the best trees allowed to slightly improve classification accuracies. It is also worth to notice that during the parameters optimization the other (typical) cases produced a little worse results that the given one.

Table 3. Classification errors for datasets with discrete class attribute

Dataset	Hepatitis	Dermatology	Pima Indians Diabetes	Ionosphere
Fuzzy Forest with C–Fuzzy Decision Trees	34.19	2.99	26.31	13.39
Fuzzy Random Forest with C–Fuzzy Decision Trees	34.19	2.18	26.56	12.24
Fuzzy Forest with Cluster–Context Fuzzy Decision Trees	34.84	2.73	26.05	12.25
Fuzzy Random Forest with Cluster–Context Fuzzy Decision Trees	34.19	2.73	25.92	11.68

7   Conclusion

The Fuzzy Random Forest with Cluster–Context Decision Trees classifier was presented in this paper. The created solution was tested using datasets both with continuous and discrete decision attribute. The classification accuracy was compared with the results achieved with the Fuzzy Random Forest with C–Fuzzy Decision Trees. The experiments showed that in most cases Fuzzy Random Forest with Cluster–Context Decision Trees gives better results that Fuzzy Random Forest with C–Fuzzy Decision Trees, especially for datasets with continuous decision attribute. It was also showed that using weights are really important for datasets with continuous decision attribute. All these results showed that the Fuzzy Random Forest with Cluster–Context Decision Trees is the valuable ensemble classifier which can allow to achieve good results in many classification or regression problems.

References

1. Bonissone, P.P., Cadenas, J.M., Garrido, M.C., Di'az-Valladares, R.A.: A fuzzy random forest. Int. J. Approximate Reasoning **51**(7), 729–747 (2010)
2. Janikow, C.Z.: Fuzzy decision trees: issues and methods. IEEE Trans. Syst. Man Cybern. Part B (Cybern.) **28**(1), 1–14 (1998)
3. Sosnowski, Z.A.: Decision rules with fuzzy granulation of knowledge. Symulacja w Badaniach i Rozwoju **3**(4), 225–232 (2012). (in Polish)
4. Pedrycz, W.: Conditional fuzzy C-means. Pattern Recogn. Lett. **17**(6), 625–631 (1996)
5. Pedrycz, W., Sosnowski, Z.A.: Designing decision trees with the use of fuzzy granulation. IEEE Trans. Syst. Man Cybern. Part A Syst. Hum. **30**(2), 151–159 (2000)

6. Pedrycz, W., Sosnowski, Z.A.: C-fuzzy decision trees. IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.) **35**(4), 498–511 (2005)
7. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Kluwer Academic Publishers, Norwell (1981)
8. Bonissone, P.P., Cadenas, J.M., Garrido, M.C., Di'az-valladares, R.A.: 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 2008), pp. 1231–1238 (2008)
9. Bonissone, P.P., Cadenas, J.M., Garrido, M.C., Di'az-Valladares, R.A.: Combination methods in a Fuzzy random forest. In: IEEE International Conference on Systems, Man and Cybernetics, SMC 2008, pp. 1794–1799, October 2008
10. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
11. Gadomer, Ł., Sosnowski, Z.A.: Fuzzy random forest with C–fuzzy decision trees. In: Saeed, K., Homenda, W. (eds.) CISIM 2016. LNCS, vol. 9842, pp. 481–492. Springer, Cham (2016). doi:10.1007/978-3-319-45378-1_43
12. Lichman, M.: UCI machine learning repository (2013)