Artificial Intelligence Research and Development
A. Cortés et al. (Eds.)
© 2022 The authors and IOS Press.
This article is published online with Open Access by IOS Press and distributed under the terms of the Creative Commons Attribution Non-Commercial License 4.0 (CC BY-NC 4.0).
doi:10.3233/FAIA220336

# Adapting a Fuzzy Random Forest for Ordinal Multi-Class Classification

# Jordi PASCUAL-FONTANILLES<sup>a,1</sup>, Lenka LHOTSKA<sup>b</sup> Antonio MORENO<sup>a</sup>, and Aida VALLS<sup>a</sup>

 <sup>a</sup> ITAKA, Dept. Enginyeria Informàtica i Matemàtiques Universitat Rovira i Virgili, Tarragona, Catalonia, Spain
 <sup>b</sup> Czech Institute of Informatics, Robotics and Cybernetics Czech Technical University in Prague, Czechia

Abstract. Fuzzy Random Forests are well-known Machine Learning ensemble methods. They combine the outputs of multiple Fuzzy Decision Trees to improve the classification performance. Moreover, they can deal with data uncertainty and imprecision thanks to the use of fuzzy logic. Although many classification tasks are binary, in some situations we face the problem of classifying data into a set of ordered categories. This is a particular case of multi-class classification where the order between the classes is relevant, for example in medical diagnosis to detect the severity of a disease. In this paper, we explain how a binary Fuzzy Random Forest may be adapted to deal with ordinal classification. The work is focused on the prediction stage, not on the construction of the fuzzy trees. When a new instance arrives, the rules activation is done with the usual fuzzy operators, but the aggregation of the outputs given by the different rules and trees has been redefined. In particular, we present a procedure for managing the conflicting cases where different classes are predicted with similar support. The support of the classes is calculated using the OWA operator that permits to model the concept of majority agreement.

Keywords. Fuzzy Random Forest, Multi-class ordinal classification, Ensemble classifiers, OWA operator

#### 1. Introduction

A Fuzzy Random Forest (FRF) is an extension of Random Forests which makes use of fuzzy logic. This addition allows them to manage uncertainty and imprecision of the data. It is composed by a set of Fuzzy Decision Trees (FDT), which can be constructed using several algorithms. This paper continues our previous work on the construction and use of FRF for binary classification in health care. Our construction method is based on Yuan and Shaw's induction algorithm [1] with some extensions presented in [2]. The algorithm has two parameters:  $\alpha$  is the threshold indicating the minimum membership degree considered during inference, and  $\beta$  is the minimum truth level required to generate a new rule. In the FRF model presented in [2], the classification has 2 steps. The first step is at the FDT level, where the predictions of the rules are aggregated to decide the output class given by the tree. The second step consists on aggregating the outputs of

<sup>&</sup>lt;sup>1</sup>Corresponding Author: Jordi Pascual-Fontanilles. E-mail: jordi.pascual@urv.cat

all the FDTs to make the final binary class assignment. Two parameters were introduced in these steps ( $\delta_1$  and  $\delta_2$ ) to allow the assignment of the *Unknown* category when the system is not sure about which of the two classes is the winner.

In ordinal multi-class decision problems we must assign a class to an instance from a set of *k* ordered possibilities  $C = \{Class_0, Class_1, ..., Class_{k-1}\}$ , where k > 2. Depending on the problem, the order can be of increasing or decreasing preference, also called Gain or Cost. For example, in medical diagnosis, the usual order goes from the best to the worst medical conditions, so that *Class*<sub>0</sub> has the healthy people and the greater the class index, the worse is the disease level.

In this paper, we adapt the 2-step classification process of FRF for the case of ordinal multi-class decision problems. In Section 2 the first classification step is adapted. Section 3 explains the modifications on the second classification step. Section 4 shown experimental results. Finally, Section 5 gives the conclusions and future work.

#### 2. Fusion in a Fuzzy Decision Tree

In a FDT we have a hierarchical structure with r branches from the root node to the leaves. Each branch corresponds to a different fuzzy rule with one or more premises consisting of linguistic variables defined on fuzzy sets. When rules are learned automatically from examples, in the leave nodes we can store a rule support value for each possible class.

When a new instance is classified, each rule provides a decision support value for each of the available classes, obtained from the product of the rule premises activation and the rule support for each class. Therefore, for each class  $Class_i$  we obtain a tuple with *r* decision support values, one for each rule:  $D_{i,1}, D_{i,1}, ..., D_{i,r}$ .

To decide which is the final class assigned to the example, we must take into account the overall support received by each class. To merge the values provided by all rules, in [3] we analysed several aggregation operators, and we proposed the use of the Choquet fuzzy integral, with a fuzzy measure based on the distorted probability. Before continuing, the support value is normalized using the truth level threshold  $\beta$  used for constructing the rules. The maximum support allowed is 1. So, for the *i*-th class, we have the support calculated using Eq. (1).

$$D_i^N = \min\left(1, \frac{ChoquetIntegral(D_{i,1}, D_{i,2}, \dots, D_{i,r})}{\beta}\right)$$
(1)

In our previous work [2], a threshold value  $\delta_1$  was introduced for binary classification, to determine if the FDT had a clear consensus on determining the winner class. To avoid mistakes, the label *Unknown* was introduced. When the difference between the two decision support values is lower than  $\delta_1$ , we assume that the FDT is not sure and, hence, the label *Unknown* is assigned. The use of  $\delta_1$  is maintained in this proposal for multiclass classification, but, because of the multiple classes, the method has been adapted. In binary classification,  $\delta_1$  was just compared with the difference of the support to the two classes. To use this threshold to the case of multiple classes, we propose the following strategy. We order decreasingly the set  $C^* = \{C \cup Unknown\}$ , according to the normalized decision support values  $D_i^N$ . Let us consider that  $C_a$  is the category with the highest decision support and  $C_b$  is the second most-supported category. The result of the analysis of the *j*-th FDT is a tuple with the predicted class and its support  $(P_j, S_j)$ . The final prediction associated with the FDT is chosen between these two categories, as described by Eq. (2). One of the strengths of an ensemble is the diversity of models composing it. An unknown prediction is preferred when the model has not a unique preferred class, which is better than making an incorrect prediction. The next section explains how the ensemble aggregates the predictions of the different trees to get the final decision.

$$(P_j, S_j) = \begin{cases} (Unknown, 0), & \text{if } D_a^N - D_b^N < \delta_1 \\ (C_a, D_a^N), & \text{otherwise} \end{cases}$$
(2)

#### 3. Fusion in a Fuzzy Random Forest

Once all the FDTs have made a prediction about the output class, all the predictions on the ensemble are aggregated to decide the final class assignment and its support. An ensemble formed by *n* FDTs has a set of *n* predicted classes, each with a support value:  $(P_1, S_1), (P_2, S_2), ..., (P_n, S_n)$ 

In the following subsections, we explain the proposal to aggregate all the predictions of the ensemble on the multi-class case. Its main elements are a weighted voting, some heuristics for the final class assignment and an OWA-based decision support score.

#### 3.1. Weighted Voting

A voting process is used to find the consensus class from the ensemble of different FDTs. Each FDT has a weight assigned to it, which represents its prediction quality. It is computed using the out-of-bag examples on the training phase. A quality metric has to be properly chosen to represent the overall quality of each FDT.

To aggregate all the predictions of a class through weighted voting, the weights of the trees that predicted it are summed, Eq. (3). As a result, each of the classes obtains a voting value  $v_i$ , which is used to decide the final prediction of the ensemble, as explained in the next subsection. Thus,

$$v_i = \sum_{j \in I} w_j \tag{3}$$

, where  $w_j$  is the weight assigned to the *j*-th tree of the set  $I = \{t \mid P_t = Class_i\}$ .

In the previous binary approach, we tested several metrics for weighting the trees. An average accuracy balancing sensitivity (2/3) and specificity (1/3) was used [4]. This balanced accuracy is specially useful in domains such as the medical one, in which a good sensitivity is a priority in order to avoid false negatives.

For the case of multi-class problems, the most appropriate and usual quality measures are F1 (balancing precision and recall) and the Weighted Cohen's Kappa  $\kappa$ . If we take into account the order between the classes, then  $\kappa$  is the best performance index, because it allows to define different penalization for mistakes between classes depending on the distance between them [5]. For this reason, we propose to use  $\kappa$  in the weighting process of FRF on ordinal multi-class classification.

# 3.2. Final Class Assignment

In binary classification, the final class assignment is made similarly to the selection of the class in a FDT, with the comparison of the support obtained by the two classes, in this case, the votes. In [2] the constant parameter  $\delta_2$  was introduced to detect the cases where the difference in votes between the two classes is not significant. In that case, when the difference in votes is lower than  $\delta_2$ , the *Unknown* category is returned by the classification model to avoid mistakes. So, the final class A was obtained as follows:

$$A = \begin{cases} Unknown, & \text{if } v_0 - v_1 < \delta_2 \\ C_a, & \text{otherwise } (Class_{0/1} \text{ with higher support}) \end{cases}$$
(4)

With multiple classes, we will take  $C_a$  and  $C_b$  again as the first and second most voted classes respectively. In this paper, for the multi-class proposal,  $\delta_2$  is preserved, but it is defined as a function depending also on the difference between the two most voted classes, according to their position in the ordered set of possible categories *C*. Let us define  $\Delta v_{ab}$  as the normalized difference of votes between  $C_a$  and  $C_b$ , Eq. (5).

$$\Delta v_{ab} = \frac{v_a - v_b}{\sum_{i \in C^*} v_i} \tag{5}$$

With this normalization, the  $\delta_2$  threshold is now defined in two parts, Eq. (6). A first constant part  $d \in [0, 0.5]$ , which is the minimum difference in votes that permits to distinguish the support of the classes and make a class assignment. In addition, the separation between the classes in the ordered scale *C* is also relevant to define when a difference in votes is important or not. It is not the same choosing between consecutive classes than between extreme classes in *C*. For that reason, the second part of  $\delta_2$  is given by the square of the difference between the positions of the classes. The value is limited to 0.5, for all the cases where the most voted class has more than half of the total votes.

Formally, let us define  $index : class \rightarrow [0, k-1]$  as the function that returns the position of a given class in the ordered set *C*, and the distance between classes as  $dist(C_a, C_b) = |index(C_a) - index(C_b)|$ . Then, the definition of  $\delta_2$  is the following:

$$\delta_2 = \begin{cases} d, & \text{if } C_a = Unk \text{ or } C_b = Unk \\ \min(0.5, d + \frac{dist(C_a, C_b)^2}{100}), & \text{otherwise} \end{cases}$$
(6)

Using this new  $\delta_2$  definition in Eq. (4) is a quite conservative approach that generates many assignments to the *Unknown* category. To avoid that the classifier does not provide an answer in too many cases, the following heuristics for assignments are proposed:

- H1: When one of the two most voted classes is the *Unknown* category and the difference in votes is small, then the model returns the class  $C_n \neq Unknown$ . However, if the difference is large enough, then the model returns the most voted class. It may be *Unknown* or the other one.
- H2: If the two classes are not unknown and the difference in votes is large enough, the most voted class must be the output of the classification model.
- H3: In the cases where the two classes are not unknown and the number of votes is similar,  $\Delta v_{ab} < \delta_2$ , two options are considered, depending on the distance of the classes in the ordered set *C*. If there is a big distance between positions of the classes in the ordered set *C*, the ensemble is considered not being certain about the prediction, and the label *Unknown* is assigned. In the case of close classes in *C*, the selected class is the one with a higher index in this ordered set. The distance threshold is based on the number of classes *k*.

These heuristics are formalized in the following equation:

$$A = \begin{cases} C_n, & \text{if } (C_a = Unk \text{ or } C_b = Unk) \text{ and } \Delta v_{ab} < \delta_2 \\ C_a, & \text{if } (C_a = Unk \text{ or } C_b = Unk) \text{ and } \Delta v_{ab} \ge \delta_2 \\ Unk, & \text{if } C_a \neq Unk \text{ and } C_b \neq Unk \text{ and } \Delta v_{ab} < \delta_2 \text{ and } dist(C_a, C_b) \ge \lfloor \frac{k}{2} \rfloor \\ C_m, & \text{if } C_a \neq Unk \text{ and } C_b \neq Unk \text{ and } \Delta v_{ab} < \delta_2 \text{ and } dist(C_a, C_b) < \lfloor \frac{k}{2} \rfloor \\ C_a, & \text{if } C_a \neq Unk \text{ and } C_b \neq Unk \text{ and } \Delta v_{ab} \ge \delta_2 \end{cases}$$
(7)

, where  $C_n \neq Unknown, n \in \{a, b\}$ , and  $m = \max(index(C_a), index(C_b))$ .

Notice that we assumed a minimization goal, where wrong classification to less severe classes is not desired. If the goal is maximization, then *m* should be the minimum.

#### 3.3. Final Decision Support

Together with the predicted class, *A*, the FRF calculates a decision support value of the prediction. This support is obtained from the corresponding support values given by each decision tree. An arithmetic average is usually used as aggregation operation. In this work, we propose the Ordered Weighted Average (OWA) to perform the aggregation [6]. OWA is a parameterized operator that permits to make a conjunctive or disjunctive aggregation. The polarity of the operation is defined with a set of weights assigned to the input values according to their position after their reordering. Having a set of support values  $S_j$  obtained with Eq. (2) for each tree, and having a weight for each position  $w_i$ , i = 1..n. The result *F* is obtained with Eq. (8), where  $S_{\sigma(i)} < S_{\sigma(i+1)}$ .

In a FRF the number of trees, *n*, is usually large (i.e. hundreds), but only a subset of the trees corresponds to the final class *A*. Given the randomness in the selection of attributes, some of these trees may produce low support values. However, if a sufficient number of trees, m <<< n, is highly supporting the selected class, the confidence about this class should be high (disjuctive policy). The weighting vector, where  $\sum_{i=1}^{n} w_i = 1$ , has been defined with weights that decrease, Eq. (8).

$$F = \sum_{i=1}^{n} w_i S_{\sigma(i)}, \text{ where } w_i = \frac{i}{\sum_{j=n-m}^{n} j}, \text{ for } i \in [n-m,n] \text{ , and } w_i = 0 \text{ otherwise} \quad (8)$$

## 4. Experiments

#### 4.1. Dataset

The experiments will be done with the diabetic retinopathy (DR) risk detection problem. In the last years, we developed the RETIPROGRAM system [7]. It is based on a binary FRF classifier, which proved to give the best results for this problem [2]. The model considers 9 different attributes (6 numerical and 3 categorical) to distinguish between the positive and the negative class. The numerical attributes were fuzzified with the ophthalmologists expertise, defining appropriate linguistic labels [7].

To test the ordinal multi-class proposal, we used a dataset from a private regional hospital. The dataset includes real data from 2084 diabetic patients. The ETDRS standard classification is considered for the target DR attribute [8]. They are ordered from lowest to highest degree of DR,  $C = \{NoDR, Mild, Moderate, Severe\}$ . The data has been split in two different datasets, training (80%) and testing (20%). Table 1 shows the distribution of the data among the target attribute classes, which has a large imbalance towards the first class, *NoDR*.

Table 1. Diabetic retinopathy data distribution

|          | Training Testing |             | Total |
|----------|------------------|-------------|-------|
| NoDR     | 1394 (83.6%)     | 349 (83.7%) | 1743  |
| Mild     | 191 (11.5%)      | 48 (11.5%)  | 239   |
| Moderate | 58 (3.5%)        | 14 (3.4%)   | 72    |
| Severe   | 24 (1.4%)        | 6 (1.4%)    | 30    |
| Total    | 1667             | 417         | 2084  |
|          |                  |             |       |

#### 4.2. Study of the Weights of FDTs in the Voting Stage

From the different contributions presented in this paper for the case of ordinal multiclass assignments with FRF, we start by testing the effect of using the  $\kappa$  index instead of Accuracy to give a weight to each of the trees in Eq. (2). We compare 3 versions of the FRF classification algorithm:

- 1. Base algorithm: it does not consider the category *Unknown*, so that we always classify an instance to one of the output classes. Hence,  $\delta_1 = 0$  and  $\delta_2 = 0$ .
- 2. Base- $\delta$  algorithm: it takes into account situations where two classes have similar conditions and then the answer is unknown, to try to avoid mistakes.
- 3. New- $\delta$  algorithm: it corresponds to the new procedure explained in this paper.

We will denote as FN (False Negatives) to the examples where the model predicts as a class lower than the real (i.e. underestimation or type-II error). Similarly, we call FP (False Positives) when the predicted class is higher than the real one (i.e. overestimation or type-I error). FNs are a kind of error non desirable in medical diagnosis, because the system does not detect the real risk for the health of the person.

We have defined the Base version as the model to improve, as it makes too many mistakes. The confusion matrix in Table 2 shows the results of the Base version. For example, in *Mild*, from the total of 48 patients, we have 15 classified to *NoDR* (FN=31%). Similarly, there is a 28% of FN in *Moderate* and 33% in *Severe*.

| Real/Predicted | NoDR | Mild | Moderate | Severe |
|----------------|------|------|----------|--------|
| NoDR           | 278  | 30   | 21       | 20     |
| Mild           | 15   | 13   | 8        | 12     |
| Moderate       | 2    | 2    | 3        | 7      |
| Severe         | 2    | 0    | 0        | 4      |

Table 2. Base method confusion matrix

Table 3 compares the results of using Accuracy or  $\kappa$  as the quality metric in the weighted voting. The 3 versions of the algorithm are compared on the Accuracy (Acc), Accuracy including unknowns as errors (Acc Unk), and Kappa index. Thresholds used are  $\delta_1 = 0.1$  and  $\delta_2 = 0.25$ , which have proven through empirical testing to be the best values. An in depth analysis of  $\delta_2$  is shown in subsection 4.3.

Table 3. Comparison of two weighted voting quality metrics Accuracy Woight 14 Waight

| Method/Weight  | A     | Accuracy weight |       |       | k weight |       |  |
|----------------|-------|-----------------|-------|-------|----------|-------|--|
|                | Acc   | Acc Unk         | Kappa | Acc   | Acc Unk  | Карра |  |
| Base           | 0.71  | 0.71            | 0.34  | 0.715 | 0.715    | 0.345 |  |
| Base- $\delta$ | 0.908 | 0.259           | 0.529 | 0.923 | 0.115    | 0.509 |  |
| New-δ          | 0.702 | 0.6             | 0.312 | 0.734 | 0.561    | 0.318 |  |

Better quality values are obtained using  $\kappa$  as weights of the trees. The accuracy index increases in the 3 versions. Weighted Kappa index is maintained to a similar level. Acc Unk metric decreases a bit, meaning the  $\kappa$  Weight produces more unknown predictions than Accuracy Weight. To further analyse the weight selection in the voting stage, Figure 1 shows the distribution of correct, incorrect and unknown predictions.





The Base- $\delta$  algorithm does not perform appropriately. It has very few errors, but there are too many unknown predictions. In contrast, the New- $\delta$  algorithm is able to reduce the incorrect predictions by introducing a moderate amount of unknowns. Comparing Accuracy and  $\kappa$  in the New- $\delta$  algorithm, even tough  $\kappa$  has less correct predictions, the amount of incorrect predictions is also smaller. We consider  $\kappa$  to have better results because of its more conservative results on unclear cases. Moreover, it obtains a better global accuracy.

#### 4.3. Study of $\delta_2$ for Class Assignment in Ordinal FRF

We studied the effect of using different d values to compute  $\delta_2$  for the final class assignment. Figure 2 shows the effect of d on the distribution of predictions among correct, incorrect and unknown. As expected, the higher d, the lower the number of unknown predictions. This is due mainly to decreasing the number of cases that enter to the second condition in Eq. (7). Accordingly, correct and incorrect assignments increase as the amount of unknowns decreases. We can see that the d parameter allows modelling the trade-off between correct, incorrect and unknown predictions. For the DR risk assessment problem, d = 0.25 was chosen for its balance of reducing incorrect predictions while not increasing unknowns and reducing correct predictions in excess. In other domains,  $\delta_2$  can be adapted according to the problem being solved, and the implications of miss-classifications.



Figure 2. Distribution of correct, incorrect and unknown class assignments for different d values

# 4.4. Study of the Heuristics for Class Assignment in Ordinal FRF

To study the effects of the proposed heuristics, the algorithm versions explained in 4.2 have been tested with two additional versions, Table 4. The additional versions differ in heuristic H3, which considers cases with the two most voted classes not being *Unknown* and a similar number of votes,  $\Delta v_{ab} < \delta_2$ . We eliminate the condition about the distance of the classes in the ordered set *C*, instead, a predetermined label is assigned. New- $\delta$ -Unk assigns *Unknown*, whereas New- $\delta$ -Max assigns the class with the higher index in *C*.

The performance improvement can be more clearly seen in the distribution of correct, incorrect and unknown assignments in Figure 3. Comparing New- $\delta$ -Unk and New- $\delta$ -Max with New- $\delta$ , we can conclude that by taking into account the distance between the majority classes, we can balance the number of errors and unknown assignments. Even though New- $\delta$ -Unk is the version with fewer errors, it is not the preferred version, as it could lead to having too many predictions assigned to *Unknown*. In the case of New- $\delta$ -Max, we can see on the FP the effect of classifying to the class with the higher index when the FRF does not have a clear consensus towards one class. By merging

|                | Accuracy Acc Unk |       | Kappa |  |
|----------------|------------------|-------|-------|--|
| Base           | 0.715            | 0.715 | 0.345 |  |
| Base- $\delta$ | 0.923            | 0.115 | 0.509 |  |
| New-δ-Unk      | 0.789            | 0.556 | 0.38  |  |
| New-δ-Max      | 0.686            | 0.561 | 0.227 |  |
| New- $\delta$  | 0.734            | 0.561 | 0.318 |  |

Table 4. Comparison of different versions of the method

both versions depending on the distance between classes, the amount of unknowns can be balanced while prioritising the classes with higher indexes. This behaviour is desired in ordinal cases such as the DR risk assessment, where a FN would have much worse consequences than a FP.



Figure 3. Distribution of correct, incorrect and unknown assignments in different versions of the FRF

#### 4.5. Study of OWA for Final Decision Support Averaging

To study the effect of a disjunctive OWA in the aggregation of the decision support, it has been compared to an arithmetic average aggregation (AA), Table 5. Experiments have been performed with n = 100 number of trees and  $m = \frac{n}{3}$  as the minimum number of trees supporting the selected class. The decision support values obtained from the test dataset, which can range in [0,1], have been split in three intervals, to indicate three levels of confidence on the answer given to the user. For each of them, the number of correct predictions is counted. We consider we should not have a low decision support in cases where a sufficient number of trees is sure about the prediction. This is the result achieved by OWA. The number of predictions in the higher intervals is greater than using an arithmetic average. As a consequence, the percentage of correct predictions in the higher interval is also greater. With AA the user has more uncertain answers, which in medicine are cases that require additional attention by the doctors, spending time and resources. So, OWA operator is recommended.

|           | Average  |             |           |          | OWA         |           |
|-----------|----------|-------------|-----------|----------|-------------|-----------|
|           | [0, 0.5] | (0.5, 0.75] | (0.75, 1] | [0, 0.5] | (0.5, 0.75] | (0.75, 1] |
| Total     | 44       | 205         | 70        | 5        | 113         | 201       |
| # correct | 37       | 149         | 48        | 5        | 80          | 149       |
| % correct | 84 %     | 73 %        | 69 %      | 100 %    | 71 %        | 74 %      |

Table 5. Decision support values with AA and disjunctive OWA

## 5. Conclusions and future work

In this paper, we presented an adaptation of a binary FRF model for ordered multi-class classification. We have focused on the 2 steps of the prediction stage, and we have redefined the procedure to manage conflicting cases. The different contributions presented have been studied on a DR dataset. From the results we conclude that:  $\kappa$  index works better than accuracy for weighting the trees; we can model the trade-off between predictions and unknowns using  $\delta_2$ ; the proposed heuristics balance the number of unknowns while prioritising classes with higher indexes, which is desired in medical applications. Finally, OWA gives an appropriate confidence value on the class assigned by the FRF.

As future work, we should test the proposed method with other datasets to confirm the observations. Then, we plan to test the method with other aggregation operators, as well as to study how it could make use of the dynamic updating method proposed in [4].

#### Acknowledgements

This work is funded by projects PI21/00064 and PI18/00169 (Instituto de Salud Carlos III & FEDER funds), and 2020PFR-B2-61 (URV). The first author has a pre-doctoral FI grant (2021 FLB 00139) from Generalitat de Catalunya and Fons Social Europeu. Prof. Lhotska is supported by the resources of the Czech Technical University in Prague.

#### References

- [1] Yuan Y, Shaw, MJ. Induction of fuzzy decision trees. Fuzzy Sets and systems. 1995;69(2):125-139.
- [2] Saleh E, Błaszczyński J, Moreno A, Valls A, Romero-Aroca P, de la Riva-Fernández S, et al. Learning ensemble classifiers for diabetic retinopathy assessment. Artif Intell Med. 2018;85:50–63.
- [3] Saleh E, Valls A, Moreno A, Romero-Aroca P, Torra V, Bustince H. Learning Fuzzy Measures for Aggregation in Fuzzy Rule-Based Models. In: Lecture Notes in Computer Science. Springer Verlag; 2018; p. 114–27.
- [4] Pascual-Fontanilles J, Valls A, Moreno A, Romero-Aroca P. Iterative Update of a Random Forest Classifier for Diabetic Retinopathy. Frontiers in Artificial Intelligence and Applications. 2021 Oct 14;339:207–16.
- [5] De La Torre J, Puig D, Valls A. Weighted kappa loss function for multi-class classification of ordinal data in deep learning. Pattern Recognition Letters. 2017;105:144–54.
- [6] Yager RR. On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decisionmaking. IEEE Trans Syst Man Cybern. 1988;18(1):183–90.
- [7] Romero-Aroca P, Valls A, Moreno A, Sagarra-Alamo R, Basora-Gallisa J, Saleh E, et al. A Clinical Decision Support System for Diabetic Retinopathy Screening: Creating a Clinical Support Application. Telemedecine and e-Health. 2019 Jan 1;25(1):31–40.
- [8] Wilkinson CP, Ferris FL, Klein RE, Lee PP, Agardh CD, Davis M, et al. Proposed international clinical diabetic retinopathy and diabetic macular edema disease severity scales. Ophthalmology. 2003;110(9):1677–82.