An Interval Valued K-Nearest Neighbors Classifier

Joaquín Derrac[1] Francisco Chiclana[2] Salvador García[3] Francisco Herrera[3]

[1]Affectv: Affectv Limited, 33-34 Alfred Place, London, WC1E 7DP, United Kingdom.
[2]Faculty of Technology, De Montfort University, Leicester LE1 9BH, United Kingdom.
[3]Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain.

Abstract

The K-Nearest Neighbors (k-NN) classifier has become a well-known, successful method for pattern classification tasks. In recent years, many enhancements to the original algorithm have been proposed. Fuzzy sets theory has been the basis of several proposed models towards the enhancement of the nearest neighbors rule, being the Fuzzy K-Nearest Neighbors (FuzzyKNN) classifier the most notable procedure in the field.

In this work we present a new approach to the nearest neighbor classifier based on the use of interval valued fuzzy sets. The use and implementation of interval values facilitates the membership of the instances and the computation of the votes in a more flexible way than the original FuzzyKNN method, thus improving its adaptability to different supervised learning problems. An experimental study, contrasted by the application of nonparametric statistical procedures, is carried out to ascertain whether the Interval Valued K-Nearest Neighbor (IV-KNN) classifier proposed here is significantly more accurate than k-NN, FuzzyKNN and other fuzzy nearest neighbor classifiers. We conclude that the IV-KNN is indeed significantly more accurate than the rest of classifiers analyzed.

Keywords: Fuzzy Nearest Neighbor, Interval Valued Fuzzy Sets, Supervised Learning, Classification.

1. Introduction

The k Nearest Neighbors classifier (k-NN) [1] is one of the most popular supervised learning methods. It is a nonparametric method which does not rely on building a model during the training phase, and whose classification rule is based on a given similarity function between the training instances and the test instance to be classified. k-NN has become one of the top ten algorithms in data mining [2], being an integral part of many applications of machine learning in various domains [3, 4].

Despite its usefulness, the k-NN classifier also presents several drawbacks which have been the focus of much research effort by the research community. Many researches have proposed different techniques to address such drawbacks, among which it is worth mentioning: the use of data reduction [5, 6], the development of methods for the automatic selection of the parameter k [7], the design of new similarity measures [8, 9], the introduction of weights for adapting the influence of instances and features [10, 11] or the definition of fast approximate versions of the nearest neighbor rule [12, 13].

The implementation of fuzzy sets for representing the degree of membership of each instance to the classes of the problem, known as the Fuzzy K-Nearest Neighbor (FuzzyKNN) classifier, has been shown an improvement of k-NN [14]. This fuzzy approach overcomes the drawback associated to the k-NN classifier , by which equal importance is given to every instance in the decision rule, regardless of its typicalness as a class prototype and the distance between it and the pattern to be classified. Indeed, fuzzy memberships enable FuzzyKNN to achieve higher accuracy rates in most classification problems. This is also the reason why it has been the preferred choice in several applications in medicine [15, 16], economy [17], bioinformatics [18], industry [19] and many other fields.

The present research contribution extends the FuzzyKNN classifier via the implementation of interval values [20] to represent the membership of each instance to the classes and the votes cast by each neighbor in the decision rule. The new Interval Valued k-Nearest Neighbors (IV-KNN) classifier overcomes a drawback associated to the FuzzyKNN, by which two particular parameters are to be fixed in advanced, and it provides a higher degree of flexibility throughout the whole decision process.

An experimental study is carried out to compare the performances of these and some advanced fuzzy nearest neighbor classifiers. In this study, their classification accuracy is tested over several well-known classification problems. The results are contrasted using nonparametric statistical procedures, able to safely confirm the conclusions drawn from the results.

The rest of this contribution is organized as follows: Section 2 describes the k-NN and FuzzyKNN classifiers, highlighting the enhancements to the former introduced by the latter. Section 3 presents the IV-KNN, as a natural extension of FuzzyKNN and k-NN. Section 4 is devoted to the experimental study performed and the analysis of its results. Lastly, Section 5 concludes the contribution.

2. K nearest neighbors and fuzzy K nearest neighbors classifiers

The k-NN and FuzzyKNN classifiers require the measuring of the similarity of a new query instance (the new instance to be classified) to the instances stored in the training set. On the next step, a set of k nearest neighbors is found. Every neighbor casts a vote on the class to which the query instance should be assigned. Finally, a class is assigned to the query instance by combining these votes.

The above procedure can be formally defined as follows: Let X be a training set, composed by N instances $X = x^0, x^1, \ldots, x^N$ which belong to C classes. Each instance $x^i = x_0^i, x_1^i, \ldots, x_M^i, x_\omega^i$ is characterized by M input attributes and one output attribute ω ($\omega \in C$). For a new query instance Q, a nearest neighbor classifier finds its k nearest neighbors in X, using a particular similarity function. Next, the class of Q is predicted as the aggregation of the class attributes ω of the k nearest neighbors.

Initially, training instances of k-NN are labeled using a *hard* scheme: The membership U of an instance x to each class of C is given by an array of values in $\{0, 1\}$, where $U_\omega(x) = 1$ and $U_c(x) = 0, c \in C, c \neq \omega$. In this scheme each instance belongs completely to one class and does not belong to any of the rest.

In the case of FuzzyKNN, the above scheme is extended into a softer scheme: Memberships are quantified in $[0, 1]$, and obtained using the following membership function

$$U_c(x) = \begin{cases} 0.51 + (nn_c/kInit) * 0.49 & \text{if } c = \omega \\ (nn_c/kInit) * 0.49 & \text{otherwise} \end{cases}$$
(1)

where nn_c are the number of instances belonging to class c found among the $kInit$ [1] neighbors of x.

This fuzzy scheme causes that instances close to the center of the classes will still keep the original crisp memberships in $\{0, 1\}$ but instances close to the decision boundaries will spread half of their membership between the neighbors' classes.

Once a query instance Q has been presented, its k nearest neighbors are searched in the training set. Although many different similarity functions can be considered for this task, the preferred choice in nearest neighbor classification is to define it via the Euclidean distance, which should suit for most of classification problems if the training data is normalized in the domain $[0, 1]$. Throughout the rest of the contribution we will follow this methodology: the Euclidean distance is used and the attributes are normalized.

Once the k nearest neighbors have been determined, the final output of the classifier is obtained by aggregating the votes cast by its neighbors. In the case of k-NN, the votes are obtained by simply adding the memberships of the k neighbors. In the case of FuzzyKNN, the Euclidean norm and the memberships are weighted to produce a final vote for each class and neighbor, using Equation (2).

$$V(k_j, c) = \frac{U_c(k_j) \cdot 1/(\|Q - k_j\|)^{2/(m-1)}}{\sum_{i=1}^{k} 1/(\|Q - k_i\|)^{2/(m-1)}}$$
(2)

where k_j is the j-th nearest neighbor ($k_j \in k$) and $m, m > 1$ is a parameter (generally $m = 2$). The votes of each neighbor are finally added to obtain the final classification, as in the case of k-NN.

Thus, both classifiers obtain their final output as a majority vote of the classes of the k nearest neighbors. However, in the case of FuzzyKNN, the use of the soft labeling scheme and the weighted votes allows to achieve a more robust classification [21], specially for instances located next to the decision boundaries, whose crisp classification would be unclear.

3. K nearest neighbors classifier based on interval valued fuzzy sets

IV-KNN is proposed as an improvement of FuzzyKNN. By introducing interval valued fuzzy sets, the membership values of every instance in the training set is represented as an array of intervals, depicting a more flexible representation of the typicalness of the instances to every class of the problem. Intervals are also considered in the computation of the votes cast by each of the k nearest neighbors in the decision rule. Using this approach we aim at reducing the sensitivity of the original FuzzyKNN classifier to the *kInit* and m parameters, alleviating the necessity of tunning their values for each specific problem.

3.1. Computation of memberships to the classes

In FuzzyKNN, the definition of the memberships of the training instances is governed by Equation (1). It is designed so that the class to which an instance originally belongs obtains more than half (0.51) of the total membership, whereas the rest is shared among the rest of the classes of the problem. By searching for the *kInit* nearest instances, local information about the relative neighborhood of the instance is considered. Therefore, this set up encloses both expert knowledge (the ω classes already assigned int he original data) and structural knowledge, obtaining with this mixture a more accurate representation of the true nature of the instance than with the k-NN classifier [21].

However, a drawback of this approach is that *kInit* must be fixed in advance. Some rules of thumb may be considered when aiming for a proper set up, such as not setting it to an extremely low

---

[1] *kInit* is usually set to an integer value between $[3, 9]$

Table 1: Data sets considered in the study

Data set	#Ins.	#At.	#Cl.	Data set	#Ins.	#At.	#Cl.
Appendicitis	106	7	2	Penbased	10992	16	10
Balance	625	4	3	Phoneme	5404	5	2
Banana	5300	2	2	Pima	768	8	2
Bands	539	19	2	Ring	7400	20	2
Bupa	345	6	2	Satimage	6435	36	7
Cleveland	297	13	5	Segment	2310	19	7
Dermatology	358	34	6	Sonar	208	60	2
Ecoli	336	7	8	Spambase	4597	57	2
Glass	214	9	7	Spectfheart	267	44	2
Haberman	306	3	2	Tae	151	5	3
Hayes-roth	160	4	3	Texture	5500	40	11
Heart	270	13	2	Thyroid	7200	21	3
Hepatitis	80	19	2	Titanic	2201	3	2
Ionosphere	351	33	2	Twonorm	7400	20	2
Iris	150	4	3	Vehicle	946	18	4
Led7Digit	500	7	10	Vowel	990	13	11
Mammographic	830	5	2	Wdbc	569	30	2
Marketing	6876	13	9	Wine	178	13	3
Monk-2	432	6	2	Winequality-red	1599	11	11
Movement	360	90	15	Winequality-white	4898	11	11
New Thyroid	215	5	3	Wisconsin	683	9	2
Page-blocks	5472	10	5	Yeast	1484	8	10

value - with $kInit = 1$ or $kInit = 2$ very few neighbors are included, and hence most of the local structural information about the data is lost - or not setting it to a very high value - which would make memberships approximately equal to the global distribution of classes in the training data, and thus discarding again the local information. Beyond this, any fixed value of $kInit$ could potentially be selected.

We argument that the use of interval values fuzzy sets to represent membership to classes could provide an alternative and efficient solution to the above drawback, and therefore making the fixation of an specific value of $kInit$ superfluous. Indeed, the use of interval values for membership could accommodate the use of different values of $kInit$ at once. That is, Equation 1 can be parametrized with $kInit$

$$U_c(x, kInit) = \begin{cases} 0.51 + (nn_c/kInit) * 0.49 & \text{if c} = \omega \\ (nn_c/kInit) * 0.49 & \text{otherwise} \end{cases}$$
(3)

and then the membership of a training instance x to a class c can be represented as an interval

$$U_c(x) = [\min [U_c(x, kInit)], \max [U_c(x, kInit)]]$$
(4)

with $kInit$ being any value of a particular set of values. Following the same recommendations that in the case of FuzzyKNN, we could consider that $kInit$ could be any of the integer values in [3,9].

Following this scheme, a more flexible and accu-

rate representation of the training instances is obtained:

- Instances located at the center of their respective classes, surrounded only by instances of its same class, will maintain full membership to it ([1.0, 1.0]) and null membership to the rest of classes ([0.0, 0.0]). This is equivalent to FuzzyKNN and k-NN.
- Instances located near the boundaries between classes, surrounded by instances of the same class, but also by some instances of other classes, will get their memberships modified as follows:
  - The lower value of the membership to ω, $\min[U_w(x, kInit)]$, can be regarded as a measure of how relevant and how many neighboring instances with a class different to ω are. The higher the number of these neighboring instances and closer to the training instance they will be, the closer to 0.51 this lower value will be.
  - The upper value of the membership to ω, $\max[U_w(x, kInit)]$, is a direct measure of how far the first neighbor not belonging to ω is. It will be 1.0 if it is not among the first nearest neighbors (according with the set up for $kInit$ chosen), and slightly lower if it is, being the specific value again dependent of the number and position of the neighboring instances not belonging to ω.
  - The lower value of the membership to the

Table 2: Results obtained

Data sets	IV-KNN	FuzzyKNN	IF-KNN	IT2KNN	k-NN
Appendicitis	87.00	**87.91**	**87.91**	**87.91**	**87.91**
Balance	88.64	88.63	83.84	**88.80**	88.48
Banana	89.30	89.19	89.40	89.51	**89.58**
Bands	70.08	**71.31**	71.02	69.51	69.75
Bupa	64.05	62.50	59.73	**65.21**	62.53
Cleveland	**57.31**	55.97	55.58	56.97	56.92
Dermatology	96.34	96.33	**96.90**	96.34	96.34
Ecoli	**82.76**	82.46	82.17	82.45	82.45
Glass	**72.61**	72.57	69.99	72.11	66.83
Haberman	68.61	67.34	**71.55**	68.96	69.90
Hayes-roth	**65.63**	**65.63**	50.00	63.75	28.75
Heart	80.00	**80.37**	79.26	78.52	79.26
Hepatitis	84.67	83.42	82.51	84.67	**89.19**
Ionosphere	84.32	84.04	**84.90**	84.32	84.03
Iris	94.67	**96.00**	94.00	95.33	**96.00**
Led7Digit	71.40	**71.60**	69.80	71.40	43.40
Mammographic	79.28	80.37	80.49	79.54	**81.71**
Marketing	**30.92**	30.79	30.11	30.81	29.51
Monk-2	84.46	89.69	**96.29**	82.50	89.16
Movement	**82.22**	**82.22**	80.56	81.11	72.50
New thyroid	93.98	93.98	**94.44**	93.98	92.58
Page-blocks	**96.05**	95.91	95.81	95.87	95.47
Penbased	99.17	99.24	**99.33**	99.14	99.13
Phoneme	**90.01**	89.36	88.45	89.65	87.75
Pima	73.32	72.93	**74.10**	73.58	72.93
Ring	59.86	60.77	64.38	58.84	**67.46**
Satimage	90.29	90.55	**90.68**	90.12	90.52
Segment	96.28	**96.36**	95.80	96.15	94.81
Sonar	**83.10**	81.64	82.10	82.14	80.21
Spambase	**91.28**	91.15	90.17	90.93	89.34
Spectfheart	76.05	74.23	72.72	77.55	**77.58**
Tae	**67.00**	66.29	51.08	**67.00**	45.08
Texture	98.40	98.53	**98.62**	98.35	98.31
Thyroid	93.97	93.97	93.92	93.94	**93.99**
Titanic	78.06	75.69	73.65	**78.15**	76.24
Twonorm	97.01	97.01	96.49	97.04	**97.07**
Vehicle	71.64	70.81	70.45	71.16	**72.34**
Vowel	**97.78**	97.47	97.47	97.07	88.69
Wdbc	**97.36**	96.65	96.31	97.18	97.18
Wine	96.60	96.01	95.49	**97.19**	96.63
Winequality-red	**69.04**	67.98	59.66	68.73	55.29
Winequality-white	**68.60**	67.52	57.35	68.38	50.92
Wisconsin	96.96	**97.25**	96.53	96.96	**97.25**
Yeast	59.71	58.89	56.88	**59.91**	57.49
Best (out of 44)	**15**	9	10	7	11
Average	**81.27**	81.10	79.72	81.11	78.15

rest of classes will be 0.0, except if one of the first nearest neighbors belongs to that class. The upper value can be regarded as a relative measure of the presence of this class among the neighborhood of the training instance, never greater than 0.49.

- Instances badly identified (possibly noise), surrounding only by instances of other classes, will get only half membership to ω ($[0.51, 0.51]$) whereas the membership to the rest of classes will be a representation of the true nature of the instances.

3.2. Interval valued voting procedure

The votes cast by each neighbor in the computation of the decision rule (Equation 2) can also be represented by intervals. In this expression, the parameter m can be used for varying the influence of the neighbors, depending on the specific value chosen.

If $m = 2$, the vote of each neighbor is weighted by the reciprocal of the squared Manhattan distance. As m increases, distances between the different neighbors will be evenly weighted, and thus the relative distances will have less effect in the determination of the votes (with $m = 3$ the weight becomes the reciprocal of the Euclidean distance). Similarly, if m is decreased, the relative distances will have a greater effect, reducing the contribution of the furthest instances (as m approaches to 1).

Although the choice recommended in [21] was to simply let $m = 2$, it is possible to consider this parameter in a more flexible way, by introducing intervals. This allows to consider a range of possible values of m instead of a single one, resulting in a more general voting mechanism.

To represent this, Equation 2 becomes:

$$V(k_j, c) = U_c(k_j) \cdot D(k_j) \qquad (5)$$

where

$$
\begin{aligned}
D(k_j) \;=\; & [\min(D(k_j, m_a), D(k_j, m_b)), \\
& \max(D(k_j, m_a), D(k_j, m_b))] \quad (6)
\end{aligned}
$$

$$D(k_j, m) = \frac{1/(\|Q - k_j\|)^{2/(m-1)}}{\sum_{i=1}^{k} 1/(\|Q - k_i\|)^{2/(m-1)}} \qquad (7)$$

and m_a, m_b are the minimum and maximum values chosen for the parameter m. Note that since the elements of Equation 5 are intervals, their product must be computed as

$$[a_1, a_2] * [b_1, b_2] = [a_1 * b_1, a_2 * b_2] \qquad (8)$$

Table 3: Parameters configuration of the algorithms

Algorithm	Parameters
IV-KNN	k value: 7, $kInit$: (3,9), m: [1.5,2]
IF-KNN	k value: 3, μ_a: 0.6, μ_r: 0.3, ν_a: 0.4, ν_r: 0.7
IT2KNN	k value: 7, $kInit$: {1,3,5,7,9}, m: 2
FuzzyKNN	k value: 5, $kInit$: 3, m: 2
k-NN	k value: 7

3.3. Combination of votes

After the votes have been computed, the final classification is obtained as the class with the maximum vote overall. In the case of IV-KNN, the votes to every class are computed as

$$V(c) = \sum_{j=1}^{k} V(k_j, c) \qquad (9)$$

where the addition of two intervals is obtained by

$$[a_1, a_2] + [b_1, b_2] = [a_1 + b_1, a_2 + b_2] \qquad (10)$$

After votes for every class have been added, every interval is converted to a single value (the center of the interval). The final classification is obtained as the class whose highest center of interval. In the case of a tie, then only the contribution of the first nearest neighbor is considered to discriminate between the tied classes [2].

4. Experimental study

An experimental study has been carried out to compare the performances of the IV-KNN and several fuzzy nearest neighbor algorithms. The classic k-NN classifier has been also included in the comparison as a baseline.

The experiment has been conducted over 44 classification data sets, available at the KEEL-Dataset repository[3] [22, 23]. Table 1 summarizes following characteristics: number of instances (**#Ins.**), number of attributes (**#At.**) and number of classes (**#Cl.**). None of the data sets includes missing values, and no nominal (discrete) attributes has been considered. Attributes' values have been normalized in $[0, 1]$ and a 10-folds cross validation procedure has been followed throughout the experiments.

Besides FuzzyKNN and k-NN, we have chosen to compare IV-KNN with two representative fuzzy nearest neighbor classifiers:

- **IF-KNN**: This classifier, proposed in [24], is an implementation of intuitionistic fuzzy sets

---

[2]This is inspired on one of the possible tie-break procedures for k-NN, where the class of the first nearest neighbor is used to break ties. Note, however, that in the case of FuzzyKNN and IV-KNN such ties are very unlikely to happen

[3]http://www.keel.es/datasets.php

for nearest neighbor classification. In IF-KNN, Intuitionistic fuzzy sets are incorporated into the voting rule, where votes are weighted using the concepts of membership and nonmembership to a certain class: If the membership of a voting instance is above a lower threshold (μ_a) and the nonmembership below a higher threshold (ν_a), the vote is considered as positive. On the other hand, if the nonmembership of a voting instance is above a lower threshold (ν_r) and the membership is below a positive threshold (μ_r), the vote is considered as negative. Otherwise, the vote is null.

- **IT2KNN**: This classifier, proposed in [25], uses type-2 interval fuzzy sets to represent the membership to the classes of the training instances. In this way, several values for the parameter $kInit$ can be considered, obtaining the final membership degree as a combination of the initial values considered. The algorithm is similar to FuzzyKNN with respect to the rest of phases.

Table 3 shows the parameters configuration selected for each algorithm. The parameter k is chosen for each algorithm in the {3,5,7,9} range [4], selecting the best performing value. For IV-KNN, the initial intervals for $kInit$ and m have been determined according to preliminary experiments. The rest of parameters have been set up following the recommendations given by the authors of each technique. The Euclidean distance has been used as similarity measure in all the experiments.

Table 2 shows the accuracy results obtained. For each data set, the best results obtained have been highlighted in **bold**. This table also includes the number of times that every method achieves the best result for a data set, and the average accuracy over the 44 datasets. These results has been contrasted by using a nonparametric statistical procedure [26], namely the Wilcoxon Signed-Ranks test [27]. Table 4 shows the results of the test, including the ranks obtained in each comparison (R^+ and R^-) and the p-value associated.

Considering the results shown at Tables 2 and 4, we can make the following analysis:

- IV-KNN effectively improves the accuracy with respect to the k-NN classifier improving its accuracy by more than a 3% on average.
- The best results of the experiments are obtained by IV-KNN, which achieves the best average accuracy result and the best performance in a higher number of data sets.
- IV-KNN improves statistically the results of all the comparison algorithms. As Table 4 shows,

---

[4] Note that k=1 has been excluded since both IV-KNN and FuzzyKNN would become the 1-NN rule. Also, no higher values of k have been chosen because all classifiers would degenerate to a majority classifier, discarding the locality capabilities of nearest neighbor algorithms

Table 4: Results of the Wilcoxon test

Comparison	R^+	R^-	p-value
IV-KNN vs FuzzyKNN	636.5	309.5	0.04833
IV-KNN vs IF-KNN	744.0	246.0	0.00308
IV-KNN vs IT2KNN	616.0	330.0	0.08536
IV-KNN vs k-NN	674.0	272.0	0.01441

the p-values reported for all the comparisons is always lower than 0.1, which means that all the differences found are significant.

These results show us that IV-KNN is a very competitive fuzzy nearest neighbor classifier. The use of interval values for defining memberships and for computing the neighbors' votes produces better results than those using FuzzyKNN in its original definition. Furthermore, IV-KNN also achieves an accuracy improvement over the IF-KNN and IT2KNN classifiers.

5. Concluding Remarks

In this contribution we have proposed a new interval valued nearest neighbor classifier. Interval valued fuzzy sets are selected as an appropriate tool for representing the instances' memberships to the different classes of the problem. They also enable our classifier to represent several votes as a single interval, thus giving more flexibility to the decision rule computation, and ultimately, improving the generalization capabilities of the nearest neighbor rule. Indeed, this has also been corroborated experimentally and it can be concluded that IV-KNN is significantly more accurate than FuzzyKNN the classic k-NN k-NN classifier, and other two advanced fuzzy nearest neighbor classifiers.

As future work, we could apply evolutionary search techniques to the model as a self-optimization procedure for setting up the required parameters: the one applied in the definition of the membership function, and the other is used in the computation of the voting rule.

References

[1] T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

[2] X. Wu and V. Kumar, editors. *The Top Ten Algorithms in Data Mining*. Data Mining and Knowledge Discovery. Chapman & Hall/CRC, 2009.

[3] A. N. Papadopoulos and Y. Manolopoulos. *Nearest Neighbor Search: A Database Perspective*. Springer-Verlag Telos, 2004.

[4] G. Shakhnarovich, T. Darrell, and P. Indyk, editors. *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice.* The MIT Press, 2006.

[5] I. Triguero, J. Derrac, S. García, and F. Herrera. A taxonomy and experimental study on prototype generation for nearest neighbor classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 42(1):86–100, 2012.

[6] S. García, J. Derrac, J. R. Cano, and F. Herrera. Prototype selection for nearest neighbor classification: Taxonomy and empirical study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):417–435, 2012.

[7] A. K. Ghosh. On optimum choice of k in nearest neighbor classification. *Computational Statistics & Data Analysis*, 50(11):3113–3123, 2006.

[8] P. Cunningham. A taxonomy of similarity mechanisms for case-based reasoning. *IEEE Transactions on Knowledge and Data Engineering*, 21(11):1532–1543, 2009.

[9] Y. Chen, E. K. Garcia, M. R. Gupta, A. Rahimi, and L. Cazzanti. Similarity-based classification: Concepts and algorithms. *Journal of Machine Learning Research*, 10:747–776, 2009.

[10] D. Wettschereck, D. W. Aha, and T. Mohri. A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. *Artificial Intelligence Review*, 11:273–314, 1997.

[11] J. Derrac, I. Triguero, S. García, and F. Herrera. Integrating instance selection, instance weighting and feature weighting for nearest neighbor classifiers by co-evolutionary algorithms. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, 42(5):1383–1397, 2012.

[12] P.J. Grother, G. T. Candela, and J. L. Blue. Fast implementations of nearest-neighbor classifiers. *Pattern Recognition*, 30(3):459–465, 1997.

[13] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Communications of the ACM*, 51(1), 2008.

[14] J. Derrac, S. García, and F. Herrera. Fuzzy nearest neighbor algorithms: Taxonomy, experimental analysis and prospects. *Information Sciences*, 260:98–119, 2014.

[15] D. Cabello, S. Barro, J.M. Salceda, R. Ruiz, and J. Mira. Fuzzy k-nearest neighbor classifiers for ventricular arrhytmia detection. *International Journal of Biomedical Computing*, 27:77–93, 1991.

[16] M. A. Chikh, M. Saidi, and N. Settouti. Diagnosis of diabetes diseases using an artificial immune recognition system2 (AIRS2) with fuzzy k-nearest neighbor. *Journal of medical systems*, 36:2721–2729, 2012.

[17] H.-L. Chen, B. Yang, G. Wang, J. Liu, X. Xu, S. Wang, and D. Liu. A novel bankruptcy prediction model based on an adaptive fuzzy k-nearest neighbor method. *Knowledge Based Systems*, 24(8):1348–1359, 2011.

[18] Y. Huang and Y. Li. Prediction of protein subcellular locations using fuzzy k-nn method. *Bioinformatics*, 20(1):21–28, 2004.

[19] T.W. Liao and D. Li. Two manufacturing applications of the fuzzy k-nn algorithm. *Fuzzy Sets and Systems*, 92:289–303, 1997.

[20] G. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications.* Prentice Hall, Upper Saddle River, NJ, 1995.

[21] J. M. Keller, M. R. Gray, and J. A. Givens. A fuzzy k-nearest neighbor algorithm. *IEEE Transactions on Systems, Man, and Cybernetics*, 15(4):580–585, 1985.

[22] J. Alcalá-Fdez, L. Sánchez, S. García, M. J. del Jesus, S. Ventura, J. M. Garrell, J. Otero, C. Romero, J. Bacardit, V. M. Rivas, J. C. Fernández, and F. Herrera. KEEL: a software tool to assess evolutionary algorithms for data mining problems. *Soft Computing*, 13(3):307–318, 2008.

[23] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera. Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing*, 17(2-3), 2011.

[24] L. I. Kuncheva. An intuitionistic fuzzy k-nearest neighbors rule. *Notes on Intuitionistic Fuzzy Sets*, 1:56–60, 1995.

[25] F. Chung-Hoon and C. Hwang. An interval type-2 fuzzy k-nearest neighbor. In *Proceedings of the 12th IEEE International Conference on Fuzzy Systems (IEEE FUZZY'03), St. Louis, Missouri, USA, May 25-28*, pages 802–807, 2003.

[26] S. García and F. Herrera. An extension on Statistical Comparisons of Classifiers over Multiple Data Sets for all pairwise comparisons. *Journal of Machine Learning Research*, 9:2677–2694, 2008.

[27] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.