A new fuzzy k-nearest neighbor classifier based on the Bonferroni mean

Mailagaha Kumbure Mahinda, Luukka Pasi, Collan Mikael

DOI: 10.1016/j.patrec.2020.10.005

Proof Central

Please use this PDF proof to check the layout of your article. If you would like any changes to be made to the layout, you can leave instructions in the online proofing interface. Making your changes directly in the online proofing interface is the quickest, easiest way to correct and submit your proof. Please note that changes made to the article in the online proofing interface will be added to the article before publication, but are not reflected in this PDF proof.

Pattern Recognition Letters xxx (xxxx) xxx

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

A new fuzzy k-nearest neighbor classifier based on the Bonferroni mean☆

Q1 Mahinda Mailagaha Kumbure*, Pasi Luukka, Mikael Collan

School of Business and Management, LUT University, Yliopistonkatu 34, Lappeenranta 53850, Finland

A R T I C L E I N F O

Article history:
Received 24 October 2019
Revised 10 May 2020
Accepted 10 October 2020
Available online xxx

Keywords:
Bonferroni mean
Classification
Fuzzy k-nearest neighbor
Performance measures
Local means

A B S T R A C T

We present a new generalized version of the fuzzy k-nearest neighbor (FKNN) classifier that uses local mean vectors and utilizes the Bonferroni mean. We call the proposed new method Bonferroni-mean based fuzzy k-nearest neighbor (BM-FKNN) classifier. The BM-FKNN classifier can be easily fitted for various contexts and applications, because the parametric Bonferroni mean allows for problem-based parameter value fitting. The BM-FKNN classifier can perform well also in situations where clear imbalances in class distributions of data are found. The performance of the proposed classifier is tested with six real-world data sets and with one artificial data set. The results are benchmarked with classification results obtained with the classical k-nearest neighbor-, the local mean-based k-nearest neighbor-, the fuzzy k-nearest neighbor- and other three selected classifiers. In addition to this, an enhancement of the local mean-based k-nearest neighbor classifier by using the Bonferroni means is also proposed and tested. The results show that the proposed new BM-FKNN classifier has the potential to outperform the benchmarks in classification accuracy and confirm the usefulness of using the Bonferroni mean in the learning part of classifiers.

1. Introduction

In this paper, we focus on the k-nearest neighbor (KNN) classification method and its generalizations. The objective of classification (algorithms) is to identify the class to which a new unclassified object or sample belongs to. In supervised machine-learning based algorithms the classification is done based on previous training of the algorithm with pre-classified data. The KNN algorithm introduced in [1] is a well-known supervised machine-learning based classification technique that is used in a wide range of applications and is one of the most used methods in classification today. The KNN classifier confronts the classification problem by first measuring the similarity (distance) between a new to-be-classified sample and training samples, to observe the k nearest neighbors for the new sample, and then determines the membership of the new sample to the class that has the largest number of neighbors with the new sample [2].

The performance of the KNN classifier is generally good, however, it is well known that the prediction accuracy of the method can be negatively influenced by outliers, which are likely to distort the class-distribution [3]. To deal with this problem, a local mean-based k-nearest neighbor (LM-KNN) classifier was introduced in [4]. The LM-KNN variant utilizes the local mean vectors for each class to classify a query sample to a particular class. The LM-KNN algorithm first finds the local mean vectors in each class in terms of all k nearest neighbors and then allocates the query sample to the class represented by the local mean vector that has the lowest Euclidean distance from the query sample [5,6]. The robustness and the simplicity of the LM-KNN algorithm has invited researchers to develop a variety of enhanced method variants (see examples in [2,6–10]) and to construct variant-based classification systems [11].

One propellant for the development of new KNN variants has been the observation that the original method has weaknesses. For example, in the original KNN algorithm, the already classified samples are assumed to have the same importance in the classification process of a new sample [12]. This simplification can harm the classification performance especially in situations, where class distributions are not in balance [13]. Another difficulty with the KNN model is that once the new sample is allocated to a particular class, the "strength" of membership in the class of the classified sample is not considered [14]. To remedy these problems, Keller [14] applied idea of including membership degrees [15] in

☆ Handle by Associate Editor Junwei Han.

* Corresponding author.
E-mail addresses: mahinda.mailagaha.kumbure@lut.fi, mahinda.mailagaha.kumbure@student.lut.fi (M. Mailagaha Kumbure), pasi.luukka@lut.fi (P. Luukka), mikael.collan@lut.fi (M. Collan).

https://doi.org/10.1016/j.patrec.2020.10.005
0167-8655/

the KNN approach, to produce a fuzzy version of the algorithm. Consequently, the fuzzy *k*-nearest neighbor (FKNN) classifier was created and is one of the most popular directions of the KNN developments. The FKNN technique performs the classification by introducing membership degrees to classes, while dealing with the uncertainty in the data. In this study, we extend the FKNN classifier further, by utilizing local mean vectors, which are formed by using the known classes of *k* nearest neighbor sample vectors. To generalize these local mean vectors, the Bonferroni mean operator is used and the resulting local Bonferroni mean vectors are used to measure the similarity of the new sample to the classes.

KNN is based on the majority voting principle, where the class of a new sample is based on nearest neighbors and their majority class. In the case that a data set is clearly imbalanced an observed drawback of majority voting principle is that the classified samples of the class or classes with a large number of samples tend to dominate the prediction of the new sample simply due to the fact that they often are more numerous among the *k* nearest neighbors [16]. A way to overcome this drawback is to use local means calculated from the classes that are represented within the nearest neighbors instead. Class assignment is then done based on the closest local mean vector rather than based on the number of nearest neighbors. In this way the classes with the highest number of samples will not have such domination over the less numerous classes. The FKNN bases the classification on the most frequent class and also the distance of the unclassified sample to the nearest neighbors. The distance that can be interpreted as imprecision with regards to similarity of individual samples also affects classification. Averaging operators also are able to overcome problems with “individual imprecision” - this can be understood also as the “wisdom of the crowd” and the first one to discuss this issue was Aristotle [17]. Later Francis Galton made this notion popular by his famous example of a country-fair contest of weight estimation [18]. Based on these precursors one can expect that using local means should have a better predictive power than individuals alone.

The Bonferroni mean is an aggregation operator that was originally introduced in [19] and further developments were discussed, e.g., in [20,21]. It can be defined as a function of means and it has been used as a very useful indicator in many applications [see 22,23]) due to its capability to perceive inter-relationships and to allow multiple comparisons between input arguments [24].

Some previous studies have noted that using the arithmetic mean is not producing optimal results with classifiers, instead performance could be improved by using alternative averaging operators, for example, generalized means [25], ordered weighted means (OWA) [26], and harmonic mean [6]. We note that as the generalized mean is a special case of the Bonferroni mean and results gained with generalized mean are at least as good as with arithmetic mean and often better, we can expect that results gained using Bonferroni mean are at least as good as with generalized mean and in some cases better. As the Bonferroni mean operator is applied to compute the class-representative local-mean vectors, one must be aware of the possibility to optimize the parameters to fit the context (particular data sets). Changing the parameters of the Bonferroni mean allows us to find good (optimal) parameter values, which will enhance the classification accuracy. By altering the parameters, one can “choose” several well-known means through the Bonferroni mean operator, such as the geometric, arithmetic, quadratic, and power means.

We study the performance of the proposed variant by using both artificial and real-life data sets containing binary and multi-class classification problems. To compare the performance of the proposed BM-FKNN method we benchmark its performance with the performances of FKNN, LM-KNN, KNN, SVM, NB, and the similarity classifier. In addition to this, we also investigate the classification performance of an improved variant of the LM-KNN classier that uses the Bonferroni mean - this is the second new variant proposed in this research. To evaluate the performance we use accuracy, sensitivity, and specificity as our performance measures. Besides this we also test whether differences between the classification accuracy of the BM-FKNN and the benchmarks is statistically significant.

2. *K*-nearest neighbor classifier variants and the Bonferroni mean

In this section, we briefly present the theoretical underpinnings of the KNN, LM-KNN, FKNN classifiers, and the Bonferroni mean operator.

2.1. K-nearest neighbor, fuzzy KNN and local mean based k*-nearest neighbor classifiers*

A formal definition of the KNN method is presented below.

Let X $(x_1, x_2, \ldots, x_N)$ be a training set, formed by N samples, and C $(\omega_1, \omega_2, \ldots, \omega_C)$ classes (that is, $X = \{x_j, c_j\}_{j=1}^{N}$, where $c_j \in C$). Each sample x_j $(x_j^1, x_j^2, \ldots, x_j^S, x_{c_j})$ contains S features. If a new query sample y is given, then it is assigned into a class (ω^*) correctly by using the following steps:

1 Choose the number of k nearest neighbors $(1 \le k \le N)$ to the new sample
2 Compute the Euclidean distances from y to x_j for all j. Also other distance measures can be used.
3 Find the set of k nearest neighbors from the X by using sorted distances in an ascending order.
4 Identify the classes represented by the k nearest neighbors.
5 Classify y into the class to which the largest number of k nearest neighbors belong to.

LM-KNN algorithm is a simple and robust extension of the KNN method [4]. In this method, a local mean vector of k nearest neighbors in each class is used to assign the correct class for the query sample. The process of the LM-KNN algorithm can be summarized as follows:

1 Find the k nearest neighbors from the training set X for each class ω_i by using the Euclidean distance in an ascending order.
2 Compute the local mean vector for each class using the k nearest neighbors found in the step 1.
3 Assign y into the class in which the local mean vector has the minimum Euclidean distance from y.

Underlying idea in the FKNN method is that a membership degree to each class is assigned to the new query sample and the highest membership degree dominates the decision about classification [14]. Membership degree indicates the proportion to which the query sample belongs to each one of the available classes. These membership degrees are weighted by the inverse of the distance of the query sample to its k nearest neighbors in the membership function. Along with this, a fuzzy strength parameter m is

Please cite this article as: M. Mailagaha Kumbure, P. Luukka and M. Collan, A new fuzzy *k*-nearest neighbor classifier based on the Bonferroni mean, Pattern Recognition Letters, https://doi.org/10.1016/j.patrec.2020.10.005

employed to provide the relative importance to the distance to be weighted, when determining the contribution of the neighbors to the membership degree. The assigned membership degree of the query sample y in each class i is labeled by the k nearest neighbors and is measured as follows:

$$u_i(y) = \frac{\sum_{j=1}^{k} u_{ij}(1/\left\|y - x_j\right\|^{2/(m-1)})}{\sum_{j=1}^{k}(1/\left\|y - x_j\right\|^{2/(m-1)})} \quad (1)$$

where, u_{ij} is the membership of the jth sample in the ith class of the training set and $m \in (1, +\infty)$ ($m = 2$ is often used).

2.2. The Bonferroni mean operator

The Bonferroni aggregation operator was introduced by Bonferroni [27] in the1950's and later extended by other researchers (see [20,28–30], and [31]). The Bonferroni mean consists of two parts, outer and inner part. Each argument of the outer part is the product of one argument and the average of all the other remaining inner arguments, this combination is what makes it a unique in terms of aggregation, [28]. The Bonferroni mean is defined as:

Let $\mathbf{x} = (x_1, x_2, \ldots, x_n), x_i \in [0, 1]\ \ \forall i \in \mathbb{N}$ be a vector with at least one $x_i \neq 0\ \ \forall\ \ i = 1, 2, \ldots, n$ and $\alpha,\ q \geq 0$ be parameters. The general Bonferroni mean of x_i is defined by Bonferroni [27]:

$$B^{p,q}(X) = \left(\frac{1}{n}\sum_{i=1}^{n} x_i^p \left(\frac{1}{n-1}\sum_{i,j=1, j\neq i}^{n} x_j^q\right)\right)^{\frac{1}{p+q}} \quad (2)$$

As an averaging operator Bonferroni mean satisfies all necessary axioms (see [20]) that an averaging operator is typically required to satisfy.

3. Proposed fuzzy *k*-nearest neighbor classifier, based on Bonferroni mean vectors

By adding a computation of the local Bonferroni mean vectors into the learning (training) part of the FKNN algorithm, we introduce the new BM-FKNN classifier. As in the FKNN method, the BM-FKNN classifier starts with the estimation of the distances from the query sample y to the labeled samples $\{x_j, c_j\}_{j=1}^{N}$ and the set of k nearest neighbors $nn^k(y)$ is observed. The idea is then to group the k nearest neighbors into sub-samples based on the classes they belong to. These sub-samples representing each class are used in the calculation of the Bonferroni mean vectors. That is, if the $nn^k(y)$ is $\{x_j, c_j\}_{j=1}^{k}$ and $c_j \in (\omega_1, \omega_2, \ldots \omega_C)$, then the local Bonferroni mean vectors with the corresponding classes are $\{B_r, \omega_r\}_{r=1}^{t},\ \ 1 \leq t \leq C$. This also implies that the number of local mean vectors relies on the number of classes that appear in the set of k nearest neighbors. Then the Euclidean distances (d_{EUC}) between the query sample y and the local Bonferroni mean vectors are computed. These distances $d_{EUC}(y, \{B_r\}_{r=1}^{t})$ are used to measure the membership degrees of the query sample with the classes the mean vectors represent $\{\omega_r\}_{r=1}^{t}$ by using the Eq. (1). Finally, the query sample y is classified to the class ω^* with which the sample has the highest membership degree with.

The pseudo code for the BM-FKNN algorithm is summarized as:

Algorithm 1 BM-FKNN.

Input: $\{x_j, c_j\}_{j=1}^{N}$ (labeled set), y (query sample), $k(1 \leq k \leq N)$, p, q ($p, q > 0$)
Output: The class label for y
Begin

for $j = 1$ to N **do**
  Compute $d_{EUC}(y, x_j)$ from y to x_j
  if $j < k$ **then**
    Add x_j to $nn^k(y)$
  else if x_j is closer to y than any of neighbors in $nn^k(y)$ **then**
    Drop the farthest neighbor from the set $nn^k(y)$ and add x_j
  end if
end for
for $r = 1$ to t **do**
  Find B_r in the set $nn^k(y)$ using using equation (2) and set the correspond class ω_r.
  Compute $d_{EUC}(y, B_r)$ from y to B_r.
  Assign membership $u_r(y)$ to ω_r in terms of weighed distanceaccording to:

$$u_r(y) = \frac{\sum_{r=1}^{t} u_{rr}(1/\|y - B_r\|^{2/(m-1)})}{\sum_{r=1}^{t}(1/\|y - B_r\|^{2/(m-1)})} \quad (3)$$

  where u_{rr} is 1 for known class and 0 for other classes.
end for
return ω^* (predicted class that has the highest membership degree) for y, $\omega^* \in (\omega_1, \omega_2, \ldots, \omega_t)$.
End

The proposed method uses the local sub-samples to create local mean vectors for all classes that are represented by the k nearest neighbors. In other words, in BM-FKNN the locally created representative vectors for each class, well-positioned to perceive the class-information, are used instead of comparing the query sample directly to the original k nearest neighbors. The class imbalance-problems which have found to be difficult to original KNN, due to domination of majority class, can this way be overcomed by using the local means. Moreover, problems that appear when using imprecise data in situations, where the samples from different classes are very close to each other [32], can also be remedied.

Selection of the k value (number of nearest neighbors used) has typically a critical role in classification accuracy. A very low k may produce inadequate classification results, while a too high k may cause outliers to affect the classification [5]. In connection with the proposed method, the k values selected can be quite high, because this allows the method to capture larger class-representative sub-samples and to create more accurate local Bonferroni mean vectors.

3.1. LM-KNN classifier with Bonferroni means

In addition to the main contribution of introducing a new BM-FKNN classifier, we also investigated how using the Bonferroni mean influences the performance of the LM-KNN classifier, specifically the application to the computation of the local mean vectors. In other words, we present and test a new LM-KNN classifier variant with Bonferroni means. For the purpose of simplification, we address this method as BM-KNN in the following sections.

Table 1
Information on the data sets used.

Data set	Database	Instances	Features	Classes
Car	KEEL	1728	6	4
Vehicle	KEEL	846	18	2
Ionosphere	UCI	351	34	2
Mammogram	UCI	961	6	2
Wine	UCI	178	13	3
Page Blocks	KEEL	548	10	5

4. Data sets and testing methodology

This section briefly introduces the used data sets and presents the testing methodology of the proposed new methods.

4.1. Artificial data with imbalance rate modifications

In most of the classification problems, we have to deal with imbalanced classes that is, the number of samples per class is not the same or even similar [33]. Typically imbalance is defined as a ratio between the number of samples in the larger class and the smaller class(es) [34]. As already discussed, this can be a problem for the classical KNN and means that the more frequently present class may tend to dominate the prediction of the new samples, because they are often more common among the k nearest neighbors. Because of this we also test how imbalance between classes affects the performance of BM-FKNN and the benchmarks. The testing data included two classes: class $1 \sim \mathcal{N}(9, 4^2)$ with 10 features and a sample size of 100, and class $2 \sim \mathcal{N}(10, 6^2)$ with 10 features and a sample size of n that was variable from the set (100, 90, 80, 70, 60, 50, 40, 30, 20, 10). In this way, data with the imbalance ratio (1/1,1/0.9,....,1/0.1) was adjusted in ascending order and the tested classifiers' performance measured for each case.

4.2. Real-world data

In addition to the artificial data, this study uses also six real-world data sets: Car data, Vehicle data, Ionosphere data, Mammogram, Wine data, and Page Blocks data, all of which are freely available at the KEEL repository [35] and the UCI Machine Learning repository [36]. Vehicle, Ionosphere, and Mammogram data represent binary class problems and Car, Wine, and Page Blocks data multi-class problems. The entry errors and quality issues on the data were studied and fixed before using them. The characteristics of each of data set are summarized in Table 1.

4.3. Performance measures used

Next, we shortly go through the performance metrics we used in this study. Since we have multi-class classification problems in our study, we also shortly present their multi-class analogs. To evaluate classification methods the most common metric used is *accuracy* [2,12,37] as a percentage of correct predictions with respect to the total number of original tested samples. Reporting accuracy results alone is often not enough to conclude that that the performance of a classifier is useful for a given task. Hence, additional performance measures such as the *sensitivity* and the *specificity* are also needed to more comprehensively evaluate the performance of classifiers. Here we use all three measures to better understand the "goodness" of the proposed classifiers and their benchmarks.

4.3.1. Binary-class problem

In the binary classification, there are only two classes, one is a positive (P) and other is a negative (N) class. There are four possible outcomes from the classification model such as true positive (TP), true negative (TN), false positive (FP), and false negative (FN), and T and F are shaped by predicted class and P and N are shaped by the actual class. Using these metrics, the performance measures in the classification are defined as $Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$, $Sensitivity = \frac{TP}{TP+FN}$ and $Specificity = \frac{TN}{FP+TN}$.

4.3.2. Multi-class problems

Multi-class classification refers to classification tasks, where there are more than two classes. In this research, we utilize the performance measures computation for the multi-class problems proposed in [38]. General notation for performance metrics for the multi-class classification is defined in the following way:

Suppose a confusion matrix with C (>2) classes, represented by $\{a_{l,m}\}_{l=1,m=1}^{C}$, and $a_{l,m}$ is an element of a row l and a column m in a matrix. When $l = m$, $a_{l,m}$ indicates the number of samples classified correctly to the correspond class and when $l \neq m$ indicates the number of misclassified samples of class ω_l as class ω_m. Then the number of true positives, true negatives, false positives, and false negatives for each class ω_i $(i \in C)$ can be measured as follows: $TP(\omega_i) = a_{i,i}$, $\forall\, i \in C$, $TN(\omega_i) = \sum_{l=1}^{C}\sum_{m=1\, l,m\neq i}^{C}(a_{l,m})$, $FP(\omega_i) = \sum_{l=1}^{C}(a_{l,i}) - TP(\omega_i)$ and $FN(\omega_i) = \sum_{m=1}^{C}(a_{i,m}) - TP(\omega_i)$. The accuracy, sensitivity, and specificity are computed for each class using above measures. The averages of these measures for all classes are considered as the final performance measures, in vein with [39].

4.4. Experimental setting and evaluation

In each selected data set (including artificial data set), the data sets were separated into a 40% training set, a 40% validation set, and a 20% testing set. The stratified random sampling technique was used in the sampling to ensure that class proportions in each of the divided sets are the same as they are in the whole data set. The hold out method was used for the cross-validation, in which 30 splits of the training and validation sets were randomly generated (30-fold cross validation).

We considered the number of neighbors k from the set $\{1, 2, \ldots, 25\}$. This was due to our assumption that the performance of the proposed BM-FKNN method would increase (and relatively increase), when the value of k increases. Pan et al [6] had provided evidence in favor of this assumption by showing that a multi-local means based k harmonic nearest neighbor classifier achieved better performance in the classification with high k values. The values for the parameters p and q of the Bonferroni mean were chosen from the range $\{0, 1, \ldots, 9, 10\}$. We first optimized the parameter values with the training & validation step and the gained optimal values were then used to test the performance of the new method with the testing sample. Following the recommendations in [2,14], the fuzzy strength parameter m was kept at $m = 2$ for both BM-FKNN and FKNN classifiers. The results are presented in terms of mean values for all performance measures.

To validate the performance of the proposed new methods, we compare the classification results of BM-FKNN and BM-KNN classifiers with the original KNN, FKNN, and LM-KNN and also with support vector machines (SVM) [40], naive Bayes classifier (NB) [41], and similarity based classifier (Similarity) [42]. The same training and validation samples were used for these classifiers for all data sets and their classification performance was registered for the optimized model with the test samples. We carried out the comparative test essentially on the real-world data sets in terms of the accuracy and other performance measures discussed above.

A paired t-test, in vein with [37] was also performed to reveal whether the performance difference of the proposed methods is statistically significant when compared to the benchmarks, a 0.05

Please cite this article as: M. Mailagaha Kumbure, P. Luukka and M. Collan, A new fuzzy k-nearest neighbor classifier based on the Bonferroni mean, Pattern Recognition Letters, https://doi.org/10.1016/j.patrec.2020.10.005

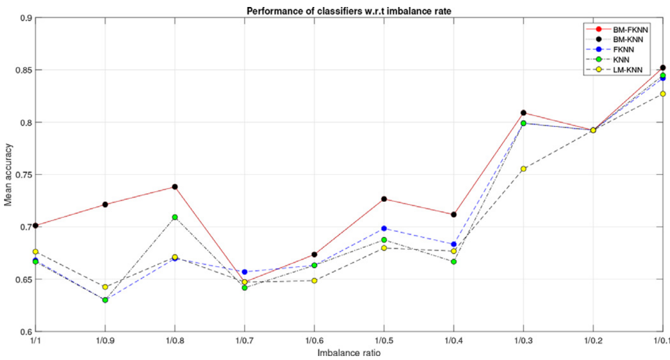

Fig. 1. Classifier performance with respect to imbalance ratio of classes.

level of significance was used. For this analysis, the samples from the hold out method (size of 1×30) were considered for each classifier, when the optimal parameters were used. In addition, the confidence interval and variances were calculated.

5. Results and discussion

In this section we first present the findings obtained for the artificial case that was generated to investigate the difference between the new proposed classifiers and the benchmark classifiers. This is followed by a presentation of the results for the real-world data sets for the training & validation step and the testing step separately.

5.1. Results for the artificial data

The artificial data was used to test the class imbalance. For this data we present a mean classification accuracy plot, taken in the testing phase for the proposed two classifiers and the three KNN-based benchmarks. From Fig. 1 one can see how the mean classification accuracy develops with respect to the imbalance ratio. We point out that the performance of both the new proposed method is the same.

In Fig. 1, imbalance ratio is presented as the sample percentage of the larger class in terms of the smaller class. For example, "1/0.3" denotes a ratio that class 1 has the sample size of 100 and class 2 has the sample size of 30. It is evident from the Fig. 1 that classifier performance is at its best, when one class has the lowest number of instances in comparison to other class (in binary class problems). A gradual increase in the mean accuracy for all classifiers can be seen with the increase of the imbalance ratio. It is clearly visible that over all imbalance ratios the BM-FKNN and BM-KNN classifiers have achieved higher accuracies than the benchmarks, the difference is most pronounced with the high imbalances. In general, this result indicates that the proposed methods are less sensitive to the class imbalance problem than the benchmarks.

5.2. Results for the real-world data

The obtained results for accuracy, sensitivity, and specificity as well as for the difference between the classification accuracies of the new methods and the benchmarks are presented.

5.2.1. Performance with the training & validation data

The parameters of the proposed new approaches and the benchmarks were optimized with the training and validation steps by using the holdout method for ensuring sample similarity. A thirty-fold cross validation was performed with the data. The obtained results for performance measures and the optimal parameter values are presented as an example for the proposed BM-FKNN classifier in Table 2. In the table, "Mean Acc." indicates the mean accuracy from the 30 sample folds gained by using the optimal parameters. Sensitivity and specificity results are also reported with mean values accordingly. The highest mean accuracy was used to determine the optimal parameter values for p, q and k.

Table 2
BM-FKNN and the BM-KNN classifier results in the validation part.

Data set	Mean Acc.	Sensitivity	Specificity	Opt. parameters
Car	0.9271	0.8079	0.9637	$k=3, p=1, q=1$
Vehicle	0.9340	0.8557	0.9591	$k=4, p=3, q=1$
Ionosphere	0.8775	0.8611	0.9245	$k=7, p=1, q=0$
Mamm	0.7939	0.7901	0.7984	$k=21, p=2, q=1$
Wine	0.7414	0.7388	0.8730	$k=25, p=2, q=2$
Page Blocks	0.9358	0.9775	0.8679	$k=3, p=2, q=2$

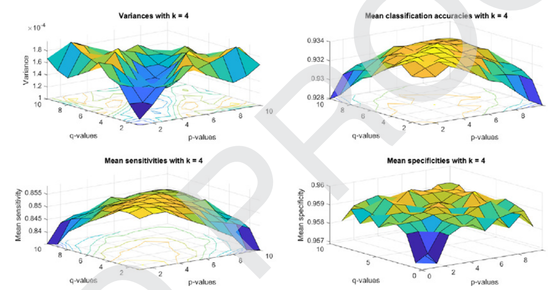

Fig. 2. Variance and performance measures for different parameter combinations (p, q) with Vehicle data for the BM-FKNN.

From the Table 2, we can see that the highest mean accuracy was reached with settings $p \in \{1, 2, 3\}$ and $q \in \{0, 1, 2\}$ for all data sets considered.

Also the sensitivity and specificity values indicate reasonable results. In addition, it is also apparent that for all cases, the specificity is higher than the sensitivity. Fig. 2 illustrates the impact of the different combinations of the parameters p and q (with the optimal k) on the selected performance measures for the Vehicle data in the training & validation step.

5.2.2. Performance with the test samples

In this sub-section, we present the classification results of the classifiers with the testing data samples, which were initially separated from the original data sets. We also include the comparison to other classifiers. Optimized parameter values and saved training samples in the validation step were used to test the classifiers with the previously unused test samples. Table 3 summarizes the results for mean classification accuracy, mean sensitivity, mean specificity, variance, and confidence interval (CI) obtained for the proposed BM-FKNN and BM-KNN methods and for the benchmarks over all considered data sets. The results for the BM-FKNN and the BM-KNN are the same and they are presented in the same column.

The results from the test sets show that the proposed classifiers have high classification accuracy compared to the benchmarks.

From Table 3 one can observe that the proposed new methods outperform all benchmarks with two data sets and that the performance is second-best with three data sets. The mean sensitivity and specificity remains high for all data sets. Besides this, interestingly BM-KNN classifier obtained the exact results which were also obtained with BM-FKNN for all data sets. This reveals that the influence of Bonferroni mean inside the learning part of the classifier has dominating effect compared to membership degree computation in fuzzy KNN. In particular BM-KNN and BM-KNN classifiers significantly improved the accuracy compared to KNN, FKNN and LM-KNN methods. This indicates that introducing the concepts of Bonferroni mean local vectors as nearest representatives instead of k nearest samples one can generate more reasonable class repre-

Table 3
Classification results with the testing samples.

Data set	Measure	BM-FKNN/ BM-KNN	FKNN	LM-KNN	KNN	SVM	NB	Similarity classifier
Car	Mean Accuracy	**0.9292**	0.8905	0.8956	0.8845	0.8506	0.8158	0.6984
	Variance	1.37E-04	1.56E-04	1.98E-04	1.49E-04	1.67E-04	2.50E-04	2.29E-04
	CI	[0.9237 0.9347]	[0.8846 0.8963]	[0.8890 0.9022]	[0.8795 0.8895]	[0.8445 0.8566]	[0.8083 0.8232]	[0.6913 0.7055]
	Mean Sensitivity	0.8251	0.658	0.6345	0.6173	0.7	0.4005	0.6599
	Mean Specificity	0.9659	0.9291	0.9197	0.9142	0.9133	0.8946	0.9017
Vehicle	Mean Accuracy	0.9556	0.9456	0.9371	0.9456	**0.9562**	0.7015	0.6988
	Variance	1.38E-04	7.96E-05	2.77E-04	7.96E-05	8.01E-04	2.27E-04	8.81E-05
	CI	[0.9501 0.9611]	[0.9414 0.9497]	[0.9309 0.9434]	[0.9414 0.9497]	[0.9430 0.9695]	[0.6944 0.7085]	[0.6944 0.7032]
	Mean Sensitivity	0.8852	0.879	0.8695	0.879	0.9077	0.4351	0.4315
	Mean Specificity	0.9796	0.967	0.9594	0.967	0.9714	0.9444	0.9359
Ionosphere	Mean Accuracy	0.8914	0.8529	0.8914	0.8529	0.8906	**0.9164**	0.8621
	Variance	4.60E-04	0.0013	4.60E-04	0.0013	4.83E-04	9.30E-04	0.0026
	CI	[0.8814 0.9015]	[0.8362 0.8695]	[0.8814 0.9015]	[0.8362 0.8695]	[0.8813 0.9009]	[0.9022 0.9307]	[0.8382 0.8861]
	Mean Sensitivity	0.8562	0.824	0.8562	0.824	0.8601	0.9446	0.8737
	Mean Specificity	0.9523	0.9541	0.9523	0.9541	0.9649	0.8762	0.8596
Mammogram	Mean Accuracy	**0.7927**	0.7844	0.7909	0.7833	0.7906	0.7844	0.7789
	Variance	1.16E-04	1.87E-04	1.18E-04	8.68E-05	1.16E-04	1.15E-04	2.99E-05
	CI	[0.7877 0.7977]	[0.7780 0.7908]	[0.7858 0.7960]	[0.7790 0.7877]	[0.7856 0.7957]	[0.7793 0.7894]	[0.7763 0.7815]
	Mean Sensitivity	0.7528	0.7275	0.7437	0.7303	0.7819	0.7457	0.7054
	Mean Specificity	0.8343	0.8535	0.8434	0.8456	0.8058	0.827	0.885
Wine	Mean Accuracy	0.8306	0.8097	0.8306	0.8069	0.8833	**0.9722**	0.9681
	Variance	0.002	0.0028	0.002	0.0028	0.0022	2.08E-31	1.04E-04
	CI	[0.8095 0.8516]	[0.7850 0.8344]	[0.8095 0.8516]	[0.7822 0.8317]	[0.8616 0.9051]	[0.9722 0.9722]	[0.9633 0.9728]
	Mean Sensitivity	0.811	0.7897	0.811	0.7897	0.8712	0.9722	0.9724
	Mean Specificity	0.9105	0.9012	0.9105	0.9012	0.9379	0.9848	0.9846
Page Blocks	Mean Accuracy	0.9255	0.92	0.915	0.9191	0.8918	**0.9259**	0.6586
	Variance	3.74E-05	2.74E-05	3.72E-05	2.56E-04	2.52E-05	3.07E-04	0.0018
	CI	[0.9219 0.9290]	[0.9165 0.9235]	[0.9121 0.9179]	[0.9165 0.9235]	[0.8895 0.8942]	[0.9211 0.9307]	[0.7185 0.7579]
	Mean Sensitivity	0.9597	0.9619	0.9597	0.9615	0.9337	0.989	0.9954
	Mean Specificity	1	1	1	1	NaN	0.7366	0.4735
	Average (overall)	**0.8875**	0.8672	0.8767	0.8654	0.8772	0.8527	0.7775

Table 4
Results of the *t*-test on the performance of the proposed methods vs. the six benchmarks on the test sample data.

Data set	Paired-t with BM-FKNN / BM-KNN	P-value	test-statistic
Car	FKNN	2.4770e-12	significant
	LM-KNN	6.30E-10	significant
	KNN	3.88E-14	significant
	SVM	6.75E-22	significant
	NB	1.12E-25	significant
	Similarity classifier	1.57E-37	significant
Vehicle	FKNN	0.0042	significant
	LM-KNN	1.24E-05	significant
	KNN	0.0042	significant
	SVM	0.9317	not significant
	NB	5.63E-41	significant
	Similarity classifier	4.30E-42	significant
Ionosphere	FKNN	3.48E-04	significant
	LM-KNN	1	not significant
	KNN	3.48E-04	significant
	SVM	0.0025	significant
	NB	5.62E-07	significant
	Similarity classifier	0.4808	not significant
Mammogram	FKNN	0.0387	significant
	LM-KNN	0.597	not significant
	KNN	0.0055	significant
	SVM	0.5443	not significant
	NB	0.019	significant
	Similarity classifier	9.38E-06	significant
Wine	FKNN	0.0871	not significant
	LM-KNN	1	not significant
	KNN	0.0839	not significant
	SVM	1.05E-22	significant
	NB	3.34E-34	significant
	Similarity classifier	2.31E-32	significant
Page Blocks	FKNN	0.0096	significant
	LM-KNN	3.77E-06	significant
	KNN	1.02E-04	significant
	SVM	6.56E-20	significant
	NB	0.6917	not significant
	Similarity classifier	1.49E-21	significant

sentative vectors. Regarding SVM, NB and similarity classifiers even though in some cases they are able to achieve little higher accuracies BM-FKNN and BM-KNN classifiers still outperform them on majority of the data sets.

Moreover, it seems that the performance of the classification has been significantly increased by using the higher values for the parameter k with the proposed methods. Obviously, this is interesting since the low values of k are performing better for the benchmarks and it is also confirmed by showing that the KNN classifier worked well with $k = 1$ with two data sets considered. This finding is in agreement with previous findings by Derrac et al. in [2].

The preliminary conclusion that can be stated based on the results is that the proposed new classifiers outperform the KNN-based benchmarks and thus excels with data sets where KNN-based classification fits well.

Table 4 presents the paired *t*-test results for the BM-FKNN and BM-KNN and benchmark classifiers with the test samples. From the evidence on the table, it is visible that the BM-FKNN and BM-KNN methods have yielded statistically significantly higher classification accuracies in the cases where the accuracies produced were superior.

6. Conclusion

This paper introduced two new methods to the family of fuzzy k-nearest neighbor classifiers that are both developed by using the Bonferroni mean in the computation of local mean vectors, which are used in the classification of new query samples to know classes. The proposed BM-FKNN and BM-KNN methods differ from FKNN and LM-KNN methods in that they use the bonferroni mean in the computation of local mean vectors for the set of k nearest neighbors, where the difference with the FKNN is that no local mean vectors were previously calculated and with the LM-KNN mean operator was arithmetic mean.

To illustrate and study the performance of the proposed classifiers they were tested with an artificial data set and six real-world data sets. The obtained results show that the new methods can give improved classification accuracy compared to the benchmarks used. Specifically it can be mentioned that the proposed new methods matched or outperformed all KNN-based benchmarks in all performance tests. The results were tested for statistical significance and it was found that the proposed methods had better classification accuracy than the benchmarks.

From the artificial data set experiment we found that the new methods are less sensitive to class imbalances, than its "original" counterparts. From the results with the real-world data, the most obvious finding to emerge for the new methods is that the best classification accuracy is achieved with a relatively high number (k) of nearest neighbors. This is reasonable, because when the sample size increases, the mean of the sample gets closer to a precise representation of the sample. However, we should note that due to the more complex calculations involved the execution of the proposed BM-FKNN method takes little more time than that of the benchmarks. Also finding a suitable parameters for Bonferroni mean requires a lot more classification runs since grid search is used and this takes time. In other words, computational complexity of the proposed approach is rather high in comparison to the classical methods.

Moreover, this study offers some insight into our understanding of the Bonferroni means and its usage in the classifiers and learning algorithms. In fact, further research directions include testing the effect of combining Bonferroni means together with other known variants of the KNN algorithm such as IV-KNN [2], kNN-TSC [43], and modified evidential KNN [10]. It also would be interesting to see how Bonferroni means can be employed in some other machine learning applications, e.g. in [44–47], where arithmetic mean has been extensively used.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

Q2 This research was supported by the Finnish Strategic Research Council project #313396, "Manufacturing 4.0".

References

[1] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Trans. Inf. Theory 13 (1967) 21–27.
[2] J. Derrac, F. Chiclana, S. Garcia, F. Herrera, An interval valued *k*-nearest neighbors classifiers, in: in Proc. of the Int. Joint Conf. IFSA-EUSFLAT-2015, 2015.
[3] K. Fukunaga, Introduction to Statistical Pattern Recognition (2nd Ed.), Academic Press Prof., Inc., San Diego, CA, USA, 1990.
[4] Y. Mitania, Y. Hamamotob, A local mean-based nonparametric classifier, Pattern Recognit. Lett. 27 (2006) 1151–1159.
[5] J. Gou, Z. Yi, L. Du, T. Xiong, A local mean-based *k*-nearest centroid neighbor classifier, Comput. J. 55 (2012) 1058–1071.
[6] Z. Pan, Y. Wang, W. Ku, A new *k*-harmonic nearest neighbor classifier based on the multi-local means, Expert Syst. Appl. 67 (2017) 115–125.
[7] J. Chai, H. Liu, B. Chen, Z. Bao, Large margin nearest local mean classifier, Signal Process. 90 (2010) 236–248.
[8] G. Jianping, Z. Yongzhao, R. Yunbo, S. Xiangjun, W. Xiaoming, H. Wu, Improved pseudo nearest neighbor classification, IEEE Trans. Syst. 70 (2014) 361–375.
[9] M. Sumet, S. Xiangjun, G. Jiangping, N. Dejiao, A new nearest centroid neighbor classifier based on k local means using harmonic mean distance, Information 9 (2018) 234.
[10] T. Denœux, O. Kanjanatarakul, S. Sriboonchitta, A new evidential *K*-nearest neighbor rule based on contextual discounting with partially supervised learning, Int. J. Appr. Reason. 113 (2019) 287–302, doi:10.1016/j.ijar.2019.07.009.
[11] Z.G. Liu, Q. Pan, J. Dezert, G. Mercier, Hybrid classification system for uncertain data, IEEE Trans. Syst. Man Cybern.Syst. 47 (10) (2017) 2783–2790, doi:10.1109/TSMC.2016.2622247.
[12] F.C. Rhee, H. Hwang, An interval type-2 fuzzy *k*-nearest neighbor, in: Proc. of the 12th IEEE Int. Conf. on Fuzzy Systems-2003. FUZZ'03, 2003, pp. 802–807.
[13] M. Beckmann, N.F.F. Ebecken, B.S.L. Pires-de Lima, A KNN undersampling approach for data balancing, J. Intell. Learn. Syst.Appl. 7 (2015) 104–116.
[14] J.M. Keller, M.R. Gray, J.A. Givens, A fuzzy *k*-nearest neighbor algorithm, IEEE Trans. Syst. 15 (1985) 580–585.
[15] L.A. Zadeh, Fuzzy sets, Inf. Control 8 (1965) 338–353.
[16] D. Coomans, D.L. Massart, Alternative *k*-nearest neighbour rules in supervised pattern recognition : part 1. *K*-nearest neighbour classification by using alternative voting rules, Anal. Chim. Acta 136 (1982) 15–27.
[17] Aristotle, Politics, 4th century BC.
[18] F. Galton, Vox populi, Nature 75 (1949) 450–451.
[19] C. Bonferroni, Sulle medie multiple di potenze, Bolletino Math. Italiana 5 (1950) 267–270.
[20] R.R. Yager, On generalized Bonferroni mean operators for multi-criteria aggregation, Int. J. Approximate Reasoning 50 (2009) 1279–1286.
[21] S. Hongchun, S. Min, Generalized Bonferroni harmonic mean operators and their application to multiple attribute decision making, J. Comput. Inf. Syst. 8 (2012) 5717–5724.
[22] W. Guiwu, Z. Xiaofei, L. Rui, S. Hongjun, Uncertain linguistic Bonferroni mean operators and their application to multiple attribute decision making, Appl. Math. Model. 37 (2013) 5277–5285.
[23] O. Kurama, P. Luukka, M. Collan, A similarity classifier with Bonferroni mean operators, Adv. Fuzzy Syst. (2016). ID 7173054
[24] B.M. Fabio, M.M. Jose, K. Janusz, Bonferroni means with distance measures and the adequacy coefficient in entrepreneurial group theory, Knowl. Based Syst. 111 (2016) 217–227.
[25] P. Luukka, T. Leppälampi, Similarity classifier with generalized mean applied to medical data, Comput. Biol. Med. 39 (2006) 1026–1040, doi:10.1016/j.compbiomed.2005.05.008.
[26] P. Luukka, O. Kurama, Similarity classifier with ordered weighted averaging operators, Expert Syst. Appl. 40 (2013) 995–1002, doi:10.1016/j.eswa.2012.08.014.
[27] C. Bonferroni, Sulle medie multiple di potenze, Bolletino Math. Italiana 5 (1950) 267–270.
[28] G. Beliakov, S.H. Bustine, T. Calvo, A Practical Guide to Averaging Functions, Springer International Publishing, Switzerland, 2016.
[29] G. Beliakov, S. James, J. Mordelova, R.R. Yager, Generalized Bonferroni mean operators in multicriteria aggregation, Fuzzy Sets Syst. 161 (2010) 2227–2242.
[30] G. Beliakov, A. Pradera, T. Calvo, Aggregation Functions: A Guide for Practitioners, Springer-Verlag, Berlin, Heidelberg, 2007.
[31] R.R. Yager, On the dispersion measure of OWA operators, Inf. Sci. 179 (2009) 3908–3919.
[32] Z. Liu, P. Quan, D. Jean, A new belief-based *k*-nearest neighbor classification method, Pattern Recognit. 46 (2013) 834–844.
[33] S. Piri, D. Delen, T. Liu, A synthetic informative minority over-sampling (SIMO) algorithm leveraging support vector machine to enhance learning from imbalanced datasets, Decis. Support Syst. 106 (2018) 15–29.
[34] V. García, J.S. Sánchez, R.A. Mollineda, On the effectiveness of preprocessing methods when dealing with different levels of class imbalance, Knowl. Based Syst. 25 (1) (2012) 13–21.
[35] J. Alcala-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. Garc-a, L. Snchez, F. Herrera, Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework, J. Multiple-Valued Logic Soft Comput. 17 (2011) 255–287.
[36] D. Dheeru, E.K. Taniskidou, UCI machine learning repository, 2017.
[37] H.L. Chen, D.Y. Liu, B. Yang, J. Liu, G. Wang, S.J. Wang, An adaptive fuzzy *k*-nearest neighbor method based on parallel particle swarm optimization for bankruptcy prediction, in: Lecture Notes in Computer Science 6634 LNAI (PART 1), 2011, pp. 249–264.
[38] A. Tharwat, Classification assessment methods, Appl. Comput. Inf. xxx (2018) xxx–xxxx, doi:10.1016/j.aci.2018.08.003.
[39] C. Ferri, J. Hernandez-Orallo, R. Modroiu, An experimental comparison of performance measures for classification, Pattern Recognit. Lett. 30 (2009) 27–38.
[40] C. Cortes, V. Vapnik, Support-vector networks, Mach Learn 20 (1995) 273–297, doi:10.1023/A:1022627411411.
[41] D.D. Lewis, Naive (bayes) at forty: the independence assumption in information retrieval, in: Machine Learning: ECML-98, Springer Berlin Heidelberg, Berlin, Heidelberg, 1998, pp. 4–15.
[42] P. Luukka, K. Saastamoinen, V. Kononen, A classifier based on the maximal fuzzy similarity in the generalized Lukasiewicz-structure, in: 10th IEEE International Conference on Fuzzy Systems. (Cat. No.01CH37297), vol. 1, 2001, pp. 195–198.
[43] Y.H. Lee, C.P. Wei, T.H. Cheng, C.T. Yang, Nearest-neighbor-based approach to time-series classification, Decis. Support Syst. 53 (1) (2012) 207–217.
[44] H. Shi, S. Pan, J. Yang, C. Gong, Positive and unlabeled learning via loss decomposition and centroid estimation, in: IJCAI International Joint Conference on Artificial Intelligence 2018-July, 2018, pp. 2689–2695, doi:10.24963/ijcai.2018/373.
[45] Y.F. Li, J.T. Kwok, Z.H. Zhou, Semi-supervised learning using label mean, in: Proceedings of the 26th International Conference On Machine Learning, ICML 2009, 2009, pp. 633–640.
[46] C. Gong, H. Shi, T. Liu, C. Zhang, J. Yang, D. Tao, Loss decomposition and centroid estimation for positive and unlabeled learning, IEEE Trans. Pattern Anal. Mach. Intell. (2019), doi:10.1109/tpami.2019.2941684.
[47] F. Nie, J. Yuan, H. Huang, Optimal mean robust principal component analysis, in: 31st International Conference on Machine Learning, ICML 2014, 4, 2014, pp. 2755–2763.