

An Enhancement of Fuzzy K-Nearest Neighbor Classifier Using Multi-Local Power Means

Mahinda Mailagaha Kumbure^a and Pasi Luukka^a and Mikael Collan^a

^aSchool of Business and Management, LUT University, Skinnarilankatu 34, 53850 Lappeenranta, Finland, mahinda.mailagaha.kumbure@student.lut.fi; pasi.luukka@lut.fi; mikael.collan@lut.fi

Abstract

This study introduces a new method to the family of fuzzy k-nearest neighbor (FKNN) classifiers that is based on the use of power means in the calculation of multi-local means that are used in classification of samples. The proposed new classifier is called multi-local power means fuzzy k-nearest neighbor classifier (MLPM-FKNN). The proposed method can be adapted to the context (of different data sets), due to the power mean being parametric and thus allowing for testing to find the parameter value that can be optimized for the classification accuracy. Furthermore, we can find optimal value for the number of local observations used in calculation of the multi-local mean. The proposed method is usable for example in situations, where class distribution is significantly different and there is only few observations in some classes. The performance of the MLPM-FKNN classifier is studied by testing it with four datasets. The performance is benchmarked against that of the original k-nearest neighbor and the fuzzy k-nearest neighbor classifiers. We find that MLPM-FKNN classifier is able to reach a statistically significantly higher classification accuracy than the benchmarks used and has reasonable performance metrics.

Keywords: Accuracy, Classification, Fuzzy k-nearest neighbor, Performance, Power mean.

1 Introduction

The ability to classify samples to classes is a key property in a variety of scientific and engineering systems that range throughout human activities. Methods

used in classification are typically divided into either supervised classification methods (with classes pre-defined) and unsupervised classification methods (with classes not pre-defined) [17]. In this research we concentrate on the k-nearest neighbor algorithm (KNN) [3] and the family of classification algorithms built around it. The KNN is a well-known supervised classification method that has been applied in many fields and is among the more popular techniques used in classification. The KNN is based on calculating the distance between already classified (labeled) samples and the sample to be classified (unlabeled) in determining the class membership [4]. Many enhancements to the original algorithm have been proposed, see, e.g., [2, 11].

One of the variants of the original KNN is the fuzzy k-nearest neighbor (FKNN) algorithm that was introduced by Keller in 1985 [8]. FKNN models the uncertainty in the data by introducing membership degrees to classes. In this research we develop the FKNN classifier further by using multi-local mean vectors (originally introduced by Pan et al. [11] to k-harmonic mean nearest neighbor classifier) to represent the known classes. These multi-local means are further generalized by using power mean and resulting mean vectors are used to calculate the distance of the unlabeled sample from the classes. Examples on such a prototype selection that was used to obtain better accuracy in the nearest classification have been reported in [14, 13, 9]. Moreover, since power mean is used in the calculation of the class representative mean vectors, we need to find what kind of mean is suitable for the particular data set by finding proper parameter value for power mean. By varying the mean controlling parameter in the power mean allows one to find the parameter value (and the mean) that can enhance the classification accuracy since suitable mean value is data dependent. We observe that by adjusting the parameter of the power mean one can arrive at several well-known means like the arithmetic, the geometric, and the harmonic means. The classification is done in

a way that an un-labeled sample is assigned membership in the class that has the representative multi-local power mean vector that has the highest membership degree to it.

The performance of the proposed method is studied with four data sets that include binary and multi-class data and compared with the performance of the classical KNN and the FKNN. We also test whether there is a statistically significant difference between the classification results from the MLPM-FKNN and the two benchmark classifiers. In addition to accuracy also performance measures, such as sensitivity and specificity are reported.

The rest of this paper is structured into five sections. Section 2 discusses the theoretical points of departure, the KNN, the FKNN, and the power mean, which all underlie this research study. The proposed MLPM-FKNN approach is described in section 3. Section 4 introduces the data and then goes through the testing of the proposed method and section 5 presents the results obtained with a comprehensive discussion. Section 6 shortly summarizes the findings and some conclusions are drawn.

2 Preliminaries

The FKNN and KNN classifiers, and the power mean are briefly described in this section.

2.1 The k-nearest neighbor and the fuzzy k-nearest neighbor classifiers

The k-nearest neighbor classification is one of the most well-known data mining techniques and has been widely applied for data classifications in many real-world applications. The KNN algorithm starts with the estimation of similarity (distance) of a new query sample (sample to be classified) with the samples in the training set (already classified). A set of k nearest neighbors for the query sample are identified, regardless of the class they belong to. Each neighbor "casts a vote" about to which class the query sample should belong. In the end of the process, the query sample is assigned to a class by way of counting the votes - the query sample is assigned to the class that has the most votes (from the k neighbors). To give a simple example, if $k = 1$, then the query sample is classified to the class of its closest neighbor.

A formal definition of the above procedure can be given as follows: Suppose a training set X shaped by N samples as $X = x_1, x_2, \dots, x_N$ and C classes. Each sample $x_j = x_j^1, x_j^2, \dots, x_j^S, x_i^S$ is described by S input variables and an output class variable t ($t \in C$). The nearest neighbor classifier finds the k nearest neighbors

in X for a new query sample Q by using a distance function (Euclidean distance is widely used). Then, the class label for Q is assigned by taking into account the class labels t of the k nearest neighbors [4].

Studies have identified and addressed several weaknesses with the original KNN model. One identified problem is that the KNN model considers each of the k nearest neighbors to have the same importance in the classification [12]. Another observed issue with the KNN model is that once a sample is assigned to a particular class, the strength of the membership in the class is not indicated [8]. To remedy these issues Fuzzy k-nearest neighbor (FKNN) model has been introduced as an extension of the KNN [8]. Compared to the KNN, the range of k in FKNN is often wider and the classification results depend on the membership degree value of the unlabeled samples in the labeled classes in FKNN [16].

In the FKNN, for the query sample a degree of membership is assigned for each class and the decision is based on the highest membership degree [8]. In the membership function membership degrees are weighed by the inverse of distance between query sample and k nearest neighbors. In addition to this a fuzzy strength parameter m is used. The allocated degree of memberships of the query sample Q in the classes represented by the k nearest neighbors is measured as:

$$u_i(Q) = \frac{\sum_{j=1}^K u_{ij}(1/\|Q - x_j\|^{2/(m-1)})}{\sum_{j=1}^K (1/\|Q - x_j\|^{2/(m-1)})} \quad (1)$$

where, u_{ij} is the membership of the j^{th} sample in the i^{th} class in the training set, and $m > 1$ is the fuzzy strength parameter that influences to the membership degree. Often used value is $m = 2$. To define u_{ij} , there are two main approaches, one is through the crisp membership and other is through the fuzzy membership [2]. One way to define the fuzzy membership degree is introduced in [8], where k nearest neighbors are found for each training sample (x_j) and, the degree of membership of x_j in each represented class is computed as:

$$u_{ij}(x_j) = \begin{cases} 0.51 + (n_j/K) * 0.49, & \text{if } j = i \\ (n_j/K) * 0.49, & \text{if } j \neq i \end{cases} \quad (2)$$

where, n_j indicates the number of neighbors observed to belong in the j^{th} class. Once all the degrees of memberships are calculated for the query sample, the class that has the highest degree of membership is assigned to the query sample.

2.2 Power mean used in local mean computation

The power mean, also called generalized mean, is a function of a family of means. If x_1, x_2, \dots, x_n is a set of real numbers, and p is a parameter ($\in \mathfrak{R}$) then power mean (M_p) can be defined as follows.

$$M_p = \begin{cases} \prod_{i=1}^n x_i^{1/n}, & \text{if } p = 0 \\ (\frac{1}{n} \sum_{i=1}^n x_i^p)^{1/p}, & \text{if } p \neq 0 \end{cases} \quad (3)$$

The formula is defined generally for the two cases, one is when $p = 0$ (it is equal to geometric mean¹) and other is $p \neq 0$. With the power mean we can get other well-known means, when parameter p is adjusted. These special cases can be defined as harmonic mean ($p = -1$), geometric mean ($p = 0$), arithmetic mean ($p = 1$), quadratic mean ($p = 2$), cubic mean ($p = 3$), and so on.

3 Fuzzy k-nearest neighbor classifier based on the multi-local power mean vectors

As in the FKNN method, the MLPM-FKNN determines first the distance from the labeled samples $\{X_i, C_i\}_{i=1}^{N_{tr}}$ to the query sample Q and the k nearest neighbors $nn^k(Q)$ are identified (where $C_i \in (\omega_1, \omega_2, \dots, \omega_T$: class labels)). Each sample X_i ($X_i^1, X_i^2, \dots, X_i^n$) contains n attributes. The k nearest neighbors are grouped, based on the class they represent into sub-samples. Then power mean vectors for the sub-samples representing each class are computed, these are called "multi-local power mean vectors". That is, if the $nn^k(Q)$ is $\{X_i, C_i\}_{i=1}^k$ and $C_i \in (\omega_1, \omega_2, \dots, \omega_T)$, then the local power mean vectors for the corresponding classes are $\{M_{p,r}, \omega_r\}_{r=1}^t$, $1 \leq t \leq T$. This also means that the number of local mean vectors depends on the number of classes that can be found from among the k nearest neighbors. Then the Euclidean distance between the local power mean vectors, representing the corresponding classes, and the the query sample is calculated (for each local power mean vector). The distance $d_{EUC}(Q, \{M_{p,r}\}_{r=1}^t)$ between each local power mean vector and the query sample is weighted by the degree of membership in each represented class $\{M_{p,r}, \omega_r\}_{r=1}^t$. Finally, the query sample is assigned membership in the class ω^* . Algorithm 1 summarizes the MLPM-FKNN method in pseudo-code.

One motivation for building the MLPM-FKNN is that it is well-known [10] that the KNN method encounters problems, when there is a clear imbalance in the

¹It is well known that $M_p \rightarrow \prod_{i=1}^n x_i^{1/n}$ when $p \rightarrow 0$

Algorithm 1 MLPM-FKNN classifier

Inputs:

labeled(classified) samples: $\{X_i, C_i\}_{i=1}^{N_{tr}}$, unlabeled (query) sample: Q , k : number of neighbors and p : power mean scale.

Outputs:

predicted class: $\omega^* (\in \omega_1, \omega_2, \dots, \omega_T)$ for the query sample.

procedure

step 1: Compute the Euclidean distance between $\{X_i\}_{i=1}^{N_{tr}}$ and Q and sort them in an ascending order.

step 2: Set the k nearest neighbors: nn^k of Q considering sorted distances as:

$$nn^k(Q) = \{X_i, C_i\}_{i=1}^k.$$

step 3: Find the local power mean sub-samples: $\{M_{p,r}\}_{r=1}^t$ from among the k nearest neighbors: $nn^k(Q)$ for each class: $\{\omega_r\}_{r=1}^t$, $1 \leq t \leq T$ by using eq. (3).

step 4: Calculate the Euclidean distance between Q and $\{M_{p,r}\}_{r=1}^t$.

step 5: Assign the membership to $\{\omega_r\}_{r=1}^t$ by using the weighted distance with the help of eq. (1) and crisp method for computing u_{ij} .

step 6: Classify Q to the corresponding class (ω^*) with which it has the highest degree of membership.

quantity of labeled samples in the represented classes in the "proximity" of the query sample. The class with more samples tends to dominate the prediction of the query sample. This undesirable dominance can be guided by using the local mean vectors instead of using the k nearest neighbors directly. In the proposed method, the local sub-samples are used to create a local mean vector for all classes that are represented within the k nearest neighbors and thus one vector represents the whole neighborhood of the sub-sample that represents each class. The value chosen for k has potentially a great importance, when producing the local power mean samples, and ultimately in the accuracy of the classification. The classification results might be inadequate, if the chosen k is too small. In contrast, also a too high value of k may cause problems with classification, because then outliers that may be quite dissimilar to the nearest neighbors can become a part of the k nearest neighbors [7]. It seems that the performance of the proposed new model is quite high, even when high k values are used. When the sub-samples size increases with higher k the multi-local power mean vectors can be said to become more realistic or more comprehensive representations of the classes they represent, this accordingly leads to the

MLPM-FKNN classifier acquiring higher classification accuracies compared with KNN and FKNN classifiers, when value of the parameter k grows. Best classification performance is achieved, when proper parameter values for k and p are search.

4 Data and testing the proposed model

In this section, the used data sets are shortly introduced and the testing methodology is presented.

4.1 Datasets used in testing

Testing the method is carried out by using four data sets all of which are freely available at the UCI Machine learning repository[5] and at the KEEL repository[1]. Table 1 summarizes the fundamental properties of the data sets.

Dataset	Database	Samples	Attributes	Classes
Car	KEEL	1728	7	4
Vehicle	KEEL	846	18	2
Ionosphere	UCI	351	34	2
Thyroid	UCI	215	6	3

Table 1: Information and properties of the data sets.

Before using the data we studied it for entry errors and quality issues and corrected any found problems. The data sets do not contain missing values or nominal variables.

4.2 Testing methodology and determining the parameter values

The proposed new method and the benchmark methods were coded and implemented using MATLAB 2018. In the code, basic calculations have been done using built-in MATLAB functions.

In the all data sets, the data were divided into 40% for training, 40% for validation, and 20% for testing. Stratified random sampling method was applied to ensure that all samples have the same proportions of units representing the different classes present as the whole data set. For cross validation we used hold out method where division between training samples and validation samples was randomly sampled 30 times. The testing (and the data) was divided into two main phases training & validation and testing. The training & validation in the context of this research means using the training and the validation data subsets to optimize the parameter values for the number of nearest neighbors used k and the mean used p . The value of the fuzzy strength parameter m was kept constant at $m = 2$ for both MLPM-FKNN and FKNN methods.

This value was chosen based on the recommendations given in [8, 4].

The number of neighbors k was selected from the range $\{1, 2, \dots, 25\}$. The assumption here was when the k is increasing the performance of proposed MLPM-FKNN classifier would increase. Evidence in favor of this expectation can be found in the paper by Pan et al.[11], where a k -nearest neighbor based on the multi-local means reached a drop in the error rate of classification with a higher k . The parameter value for the power mean, p , was chosen (and optimized) from the set $\{-8, -7, \dots, 7, 8\}$. When the optimal parameter values were found through iteration with the training and validation subsets the performance was tested with the test subset. The results presented for the proposed new method and the benchmarks are the mean results from the thirty runs made, e.g., the mean accuracy $((Mp)_{1 \times 30})$ and the variance of accuracy of the thirty runs with the optimal p and k .

4.3 Performance measures used

The key measure we used for the evaluation is accuracy [2, 4, 12], but in practice, presenting classification results with accuracy alone is often not enough to be able to fully understand the suitability of a method for a given task. For this reason, we compute also additional performance measures such as sensitivity and specificity and we employ the following definitions [15].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

$$Sensitivity = \frac{TP}{TP + FN} \quad (5)$$

$$Specificity = \frac{TN}{TN + FP} \quad (6)$$

Where, TP (true positive), TN (true negative), FP (false positive), and FN (false negative) are on the 2×2 confusion matrix as follows.

$$\begin{matrix} & \begin{matrix} Positive(P) & Negative(N) \end{matrix} \\ \begin{matrix} True(T) \\ False(F) \end{matrix} & \begin{pmatrix} TP & TN \\ FP & FN \end{pmatrix} \end{matrix}$$

These performance measures aid to further understand the performance of the the proposed method (or any method). Illustrating this issue, let us look at a simple example: suppose that a confusion matrix exists, where $TP = 0$, $TN = 95$, $FP = 0$, and $FN = 5$. In this matrix accuracy= 95% and specificity= 100%, but the precision and the sensitivity are zero. It is easy to understand that any model with zero precision is too weak to be used in any real-world purposes and this highlights the importance of including also other

performance measures than prediction accuracy, when model performance is reported. To compute these performance measures for the binary class problems (Vehicle and Ionosphere), equations (4), (5), and (6) were used. For the multi-class problems (Car and Thyroid) we applied performance measure computations, such as they are proposed in [15]. Accordingly, TN , TP , FP , and FN first and then sensitivity and specificity were computed for each class. The average of sensitivities and specificities for each class were considered to the final performance measures of the classifier exactly as suggested in [6].

A paired t-test was used, in vein with [2] to test, whether the difference of the (best) mean accuracies achieved with 1) MLPM-FKNN and FKNN and 2) MLPM-FKNN and KNN significantly differs from zero under a 0.05 significance level. This is to say, we tested for whether a statistically significant difference in the classification accuracy of the different methods can be found.

5 Results and discussions

The first results presented here are from the training & validation step of testing the method. The values of the accuracy and performance measures achieved in the training & validation step are presented in Table 2. In the table "Max.Acc." refers to the the maximum of means of accuracy and "sensitivity" and "specificity" the means of these values. As mentioned earlier, optimal values for the parameters (Opt.param.val.) k and p were being picked for when the mean accuracy was at maximum.

Dataset	Max Acc.	Sensitivity	Specificity	Opt.param.val.
Car	0.9230	0.7903	0.9654	$p = 0, k = 25$
Vehicle	0.9412	0.8723	0.9628	$p = 1, k = 11$
Ionosphere	0.8881	0.8744	0.9265	$p = 1, k = 16$
Thyroid	0.9229	0.8813	0.9367	$p = 3, k = 7$

Table 2: Accuracy and related results for MLPM-FKNN in the training & validation step.

We can see from the Table 2, that using arithmetic mean, $p = 1$ has produced the maximum accuracy with the Vehicle and Ionosphere data sets, while geometric and cubic mean where optimal w.r.t. accuracy with the Car and Thyroid data sets respectively. It can be noted that the specificity is higher than the sensitivity in all cases considered.

By observing Figures 1 and 2 one can visually inspect the impact that different combinations of the parameters p and k have on selected performance measures for the Vehicle data set in the training & validation step.

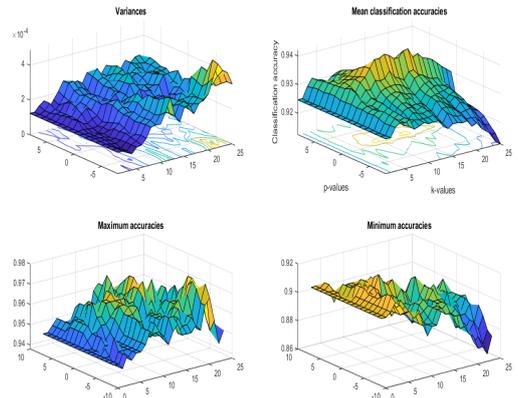


Figure 1: Variance and accuracies for different (p, k) parameter combinations with the Vehicle data

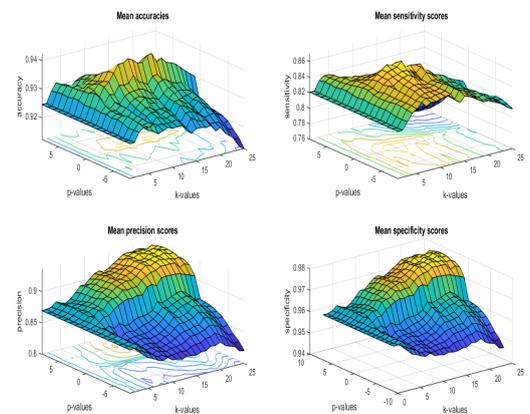


Figure 2: Mean accuracy and performance measures for different (p, k) parameter combinations with the Vehicle data

To understand how well the proposed new method performs in comparison with the two benchmark methods during the training & validation step, we show classification results for the three classifiers with all four data sets in Table 3. In addition to the maximum of the mean accuracy (Mean Acc.), variance, confidence interval (CI), and the optimal parameter k value (Op. K) is presented. As observed above, the mean accuracy and the variance are the result of the hold out method with 30 repetition.

It is visible from Table 3 that the proposed MLPM-FKNN method can outperform the FKNN and the KNN in mean classification accuracy, while the confidence intervals remain narrow. It seems that using high values for the parameter k significantly increases the performance of the classification. This is interesting, because low values of k seem to be working better

Data	Measure	MLPM-FKNN	FKNN	KNN
Car	Mean Acc.	0.9230	0.8823	0.8740
	Variance	1.13e-04	6.19e-05	6.09e-05
	CI	[0.9191, 0.9270]	[0.8794, 0.8853]	[0.8711, 0.8769]
	Op. K	$k = 25$	$k = 4$	$k = 3$
Vehicle	Mean Acc.	0.9412	0.9291	0.9274
	Variance	1.46e-04	1.92e-04	1.98e-04
	CI	[0.9367, 0.9457]	[0.9239, 0.9343]	[0.9222, 0.9327]
	Op. K	$k = 11$	$k = 4$	$k = 1$
Ionosphere	Mean Acc.	0.8881	0.8443	0.8431
	Variance	5.57e-04	4.24e-04	4.66e-04
	CI	[0.8793, 0.8969]	[0.8366, 0.8520]	[0.8350, 0.8512]
	Op. K	$k = 16$	$k = 4$	$k = 3$
Thyroid	Mean Acc.	0.9229	0.9167	0.9101
	Variance	5.92e-04	6.16e-04	6.05e-04
	CI	[0.9138, 0.9320]	[0.9074, 0.9259]	[0.9009, 0.9193]
	Op. K	$k = 7$	$k = 3$	$k = 1$

Table 3: Classification results for all three classifiers for the four datasets. CI = confidence interval

Dataset	Paired with	MLPM-FKNN (p-value)	test-statistic
Car	FKNN	$5.06e^{-24}$	significant
	KNN	$4.15e^{-28}$	significant
Vehicle	FKNN	$6.28e^{-04}$	significant
	KNN	$1.42e^{-04}$	significant
Ionosphere	FKNN	$2.26e^{-10}$	significant
	KNN	$1.91e^{-10}$	significant
Thyroid	FKNN	0.333	not-significant
	KNN	0.043	significant

Table 4: T-test results on the statistical significance in classification accuracy of the proposed new method.

for the benchmark methods and $k = 1$ has been the optimum for the KNN method with two of the data sets used. This finding corroborates what was found previously by Derrac et al.[4].

In Table 4, the results from paired t-test for testing the statistical significance of the classification accuracies between the proposed new classifier and the benchmarks are presented. The test results show that the claim that MLPM-FKNN has higher performance over the benchmarks finds statistical corroboration. One can observe that only in the case of the thyroid data the proposed new classifier cannot be said to produce statistically significantly better results in comparison with the result obtained from FKNN.

5.1 Performance with the testing data

This section presents the results obtained with the testing data (constituting of 20% of the data set size) on the classifiers, whose parameter values have been optimized in the training & validation phase. In this testing phase, the test sample set was evaluated with training set samples which were saved during the hold-out crossvalidation to get optimal parameter values

from validation results. Then mean accuracies and other performance measures were computed. Table 5 shows the results obtained for the proposed new classifier and for the benchmark classifiers for mean classification accuracies (Mean Acc.), sensitivity (Mean Sen.), specificity (Mean Spe.) and confidence interval (CI) and variance for all the four data sets. The results from the test set show that MLPM-FKNN classifier has a higher classification accuracy in all cases compared to the benchmarks.

Data	Measure	MLPM-FKNN	FKNN	KNN
Car	Mean Acc.	0.9246	0.8742	0.8632
	Mean Sen.	0.8105	0.6704	0.6085
	Mean Spe.	0.9667	0.9119	0.9014
	CI	[0.9204, 0.9288]	[0.8688, 0.8796]	[0.8577, 0.8687]
	Variance	1.26e-04	2.08e-04	2.15e-04
Vehicle	Mean Acc.	0.9294	0.9051	0.9006
	Mean Sen.	0.8568	0.7907	0.7910
	Mean Spe.	0.9521	0.9432	0.9355
	CI	[0.9241, 0.9347]	[0.8966, 0.9136]	[0.8932, 0.9080]
	Variance	2.03e-04	5.18e-04	3.95e-04
Ionosphere	Mean Acc.	0.8381	0.7957	0.7952
	Mean Sen.	0.8212	0.7686	0.7694
	Mean Spe.	0.8925	0.9215	0.9113
	CI	[0.8281, 0.8481]	[0.7875, 0.8039]	[0.7869, 0.8036]
	Variance	7.237e-04	4.81e-04	4.97e-04
Thyroid	Mean Acc.	0.9434	0.9279	0.9233
	Mean Sen.	0.9256	0.8520	0.8449
	Mean Spe.	0.9584	0.9254	0.9214
	CI	[0.9316, 0.9552]	[0.9171, 0.9387]	[0.9138, 0.9327]
	Variance	9.95e-04	8.34e-04	6.40e-04

Table 5: Classification results for all three classifiers for the four datasets. CI = confidence interval

Also mean sensitivity and specificity values show improved results compared to FKNN and KNN. Using local means in this manner also seem to lower variances a bit compared to the benchmark methods. Table 6 illustrates the results of the statistical t-test on the accuracies of the proposed method and benchmarks over the test sample. From Table 6, it is also confirmed that the MLPM-FKNN method has produced significantly higher accuracies in classification for the test set than accuracies of FKNN and KNN classifiers.

Dataset	Paired with	MLPM-FKNN (p-value)	test-statistic
Car	FKNN	$1.64e^{-18}$	significant
	KNN	$2.88e^{-25}$	significant
Vehicle	FKNN	$6.73e^{-06}$	significant
	KNN	$2.41e^{-08}$	significant
Ionosphere	FKNN	$4.95e^{-08}$	significant
	KNN	$1.06e^{-12}$	significant
Thyroid	FKNN	0.051	not-significant
	KNN	0.008	significant

Table 6: T-test results on the statistical significance for test set in classification accuracy of the proposed new method with benchmarks.

6 Conclusions

The aim of this paper was to present a new classification method in the k -nearest neighbor family of classification algorithms that is based on using the power mean to calculate a multi-local mean that serves as a representative for the representative classes near the to-be-classified sample. The proposed new classifier includes also elements of the previously presented fuzzy KNN algorithm in that the strength of class membership of the to-be-classified sample is also taken into consideration in the classification. The results of this research show that the proposed MLPN-FKNN classifier offered improved classification accuracy compared to the KNN and the FKNN classifiers with the benchmark data sets. The performance was tested by using four different data sets and the improvement in classification accuracy was found to be statistically significant.

It can be observed in connection with the new proposed method that the best classification accuracy is found with high k values (number of nearest neighbors considered). This we attribute to the use of multi-local power mean. We note that the computation time required for using the proposed MLPN-FKNN method is also longer than that of the benchmarks, it is clear that this is a result of the more numerous calculations needed caused by the increase in parameters and thus testing of more possible parameter combinations to find optimal parameters.

Future research possibilities include, e.g., testing the effect including power means would have together with other variants of KNN algorithms, such as IV-KNN [4], MLM-KHNN [11], and PTVPSO-FKNN [2].

7 Acknowledgment

This research was partly supported by the Academy of Finland project Manufacturing 4.0.

References

- [1] 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, F. Herrera, KEEL: a software tool to assess evolutionary algorithms for data mining problems, *Soft Computing* 13 (3) (2009) 307–318. URL <https://doi.org/10.1007/s00500-008-0323-y>
- [2] 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, *Lecture Notes in Computer Science* 6634 LNAI (PART 1) (2011) 249–264.
- [3] T. Cover, P. Hart, Nearest neighbor pattern classification, *IEEE Transactions on Information Theory* 13 (1) (1967) 21–27.
- [4] J. Derrac, F. Chiclana, S. García, F. Herrera, An Interval Valued K -Nearest Neighbors Classifier, in: *Proceedings of the International Joint Conference IFSA-EUSFLAT-2015*, 2015.
- [5] D. Dheeru, E. Karra Taniskidou, UCI machine learning repository (2017). URL <http://archive.ics.uci.edu/ml>
- [6] C. Ferri, J. Hernández-Orallo, R. Modroiu, An experimental comparison of performance measures for classification, *Pattern Recognition Letters* 30 (1) (2009) 27–38. URL <http://linkinghub.elsevier.com/retrieve/pii/S0167865508002687>
- [7] J. Gou, Y. Zhan, Y. Rao, X. Shen, X. Wang, W. He, Improved pseudo nearest neighbor classification, *Knowledge-Based Systems* 70 (2014) 361375.
- [8] J. M. Keller, Michal R. Gray, J. R. James A. Givens, A fuzzy k -nearest neighbor algorithm, *IEEE Transactions on Systems, Man, and Cybernetics* 15 (4) (1985) 580–585.
- [9] D. Koloseni, J. Lampinen, P. Luukka, Differential evolution based nearest prototype classifier with optimized distance for the features in the data sets, *Expert Systems with Applications* 40 (10) (2013) 4075–4082.
- [10] C. Liu, L. Cao, P. S. Yu, A hybrid coupled k -nearest neighbor algorithm on imbalance data, in: *Proceedings of the International Joint Conference on Neural Networks* (2014) 2011–2018.
- [11] Z. Pan, Y. Wang, W. Ku, A new k -harmonic nearest neighbor classifier based on the multi-local means, *Expert Systems with Applications* 67 (115-125) (2017) 21–27.
- [12] F. C. Rhee, C. Hwang, An interval type-2 fuzzy k -nearest neighbor, in: *The 12th IEEE International Conference on Fuzzy Systems*, 2003. FUZZ '03., Vol. 2, 2003, pp. 802–807 vol.2.
- [13] J. S. Sanchez, R. Barandela, A. I. Marques, R. Alejo, J. Badenas, Analysis of new techniques to obtain quality training sets, *Pattern Recognition Letters* 24 (2003) 1015–1022.

- [14] J. S. Sanchez, F. Pla, F. Ferri, Prototype selection for the nearest neighbour rule through proximity graphs, *Pattern Recognition Letters* 18 (6).
- [15] A. Tharwat, Classification assessment methods, *Applied Computing and Informatics*, (2018) accepted.
URL <https://doi.org/10.1016/j.aci.2018.08.003>
- [16] K. Toduka, Y. Endo, Fuzzy K-nearest neighbor and its application to recognize of the driving environment, in: *IEEE International Conference on Fuzzy Systems*, 2006.
- [17] S. Watanabe, *Pattern Recognition, Human and Mechanical.*, Trends in Logic, New York (N.Y.): Wiley, 1985.