K Nearest Neighbor Equality: Giving equal chance to all existing classes

B. Sierra, E. Lazkano, I. Irigoien, E. Jauregi and I. Mendialdua

Dept. of Computer Science and Artificial Intelligence
University of the Basque Country
e-mail: b.sierra@ehu.es (Basilio Sierra)
Web site: http://www.sc.ehu.es/ccwrobot

Abstract

The nearest neighbor classification method assigns an unclassified point to the class of the nearest case of a set of previously classified points. This rule is independent of the underlying joint distribution of the sample points and their classifications. An extension to this approach is the *k*-NN method, in which the classification of the unclassified point is made by following a voting criteria within the k nearest points.

The method we present here extends the *k*-NN idea, searching in each class for the *k* nearest points to the unclassified point, and classifying it in the class which minimizes the mean distance between the unclassified point and the *k* nearest points within each class. As all classes can take part in the final selection process, we have called the new approach *k* Nearest Neighbor Equality (*k*-NNE).

The experimental results we obtained empirically show the suitability of the *k*-NNE algorithm, and its effectiveness suggests that it could be added to the current list of distance based classifiers

Keywords: Nearest Neighbor, Supervised Classification, Machine Learning, Non-parametric Pattern Recognition

1. Introduction

In a supervised classification problem based on a sample of p-variate observations $\mathbf{x}_1, \ldots, \mathbf{x}_n$ classified in $\theta_1, \ldots, \theta_M$ classes or populations, and given a new observation or case $\mathbf{x}$, the aim is to classify $\mathbf{x}$ in its correct class [20]. If the modeler has complete statistical knowledge of the underlying joint distribution of the observation $\mathbf{x}$ and the category θ_m ($m = 1, \ldots, M$), a standard Bayes analysis will yield an optimal decision procedure and the corresponding minimum (Bayes) probability of error classification, R^*.

However, if the only knowledge the modeler has of the distribution is that which can be inferred from samples, then the decision to classify $\mathbf{x}$ into the category θ_m depends on the sample $\mathbf{x}_1, \ldots, \mathbf{x}_n$ along with its correct classification in categories $\theta_1, \ldots, \theta_M$, and the procedure is by no means clear. The classification problem falls then into the domain of supervised classifications, where there is no an optimal classification procedure with regards to all underlying statistics.

Assuming that the classified samples $\mathbf{x}_i$ are independently identically distributed according to the distribution of $\mathbf{x}$, certain heuristic arguments may be considered about good decision procedures. For example, it is reasonable to assume that observations which are close together (in some appropriate distance metric) will have almost the same posterior probability distributions in their respective classifications.

Thus to classify the unknown sample $\mathbf{x}$ we may choose to give a heavier weight to the nearby $\mathbf{x}_i$'s. Perhaps the simplest non-parametric decision procedure of this type is the nearest neighbor (NN) classification method, which assigns the category of its nearest neighbor to $\mathbf{x}$.

The first formulation of a rule of the NN type and primary previous contribution to the analysis of its properties is presumed to have been made by Fix and Hodges [13]. They investigated a method that is known as *k* Nearest Neighbors (*k*-NN), which assigns an unclassified point to the class most heavily represented among its *k* nearest neighbors.

In this paper we present a modification of the *k*-NN method that searches for the *k* nearest neighbors of the point to be classified in each class, and assigns the point to the class whose *k* points have the minimal mean distance to the

new point. The idea is based on the assumption that the underlying distribution of the predictor variables (components of $\mathbf{x}$) could be different in each class.

This paper is organized as follows. In section 2 we review the k-NN classification method while section 3 is devoted to Related Works in distance based classifiers; the new proposed method is introduced in section 4, in section 5 we show the experimental results obtained and concluding remarks are presented in section 6.

2. The k-NN Classification Method

Let $\mathbf{x}_1, \ldots, \mathbf{x}_n$ be a correctly classified sample in classes $\theta_1, \ldots, \theta_M$, where $\mathbf{x}_i$ takes values in a metric space upon which a distance function d is defined. We will consider the pairs $(\mathbf{x}_i, \theta^i)$, where $\mathbf{x}_i$ is the p-variate observation upon the ith individual, and θ^i is the class or category which that individual belongs to. We usually say that "$\mathbf{x}_i$ belongs to θ^i" when we mean precisely that the ith individual, upon which measurements $\mathbf{x}_i$ have been observed, belongs to category $\theta^i \in \{\theta_1, \ldots, \theta_M\}$.

Consider a new pair $(\mathbf{x}, \theta)$, where only the measurement $\mathbf{x}$ is observable, and where we estimate θ by using the information contained in the set of correctly classified points. We shall call

$$\mathbf{x}' \in \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$$

the *nearest neighbor* (NN) of $\mathbf{x}$ if

$$\min_{i=1,\ldots,n} d(\mathbf{x}_i, \mathbf{x}) = d(\mathbf{x}', \mathbf{x})\,.$$

The NN classification decision method gives to $\mathbf{x}$ the category θ^i, that is, the category of its nearest neighbor $\mathbf{x}_i$. In case of a tie between several neighbors, a modification of the decision rule is applied.

An immediate extension to this decision rule is the so called k-NN approach [7], which assigns the candidate $\mathbf{x}$ the class which is most frequently represented in the k nearest neighbors to $\mathbf{x}$. In Figure 1, for example, the 3-NN decision rule would decide that class θ_o is active because two of the three nearest neighbors of $\mathbf{x}$ belong to class θ_o.

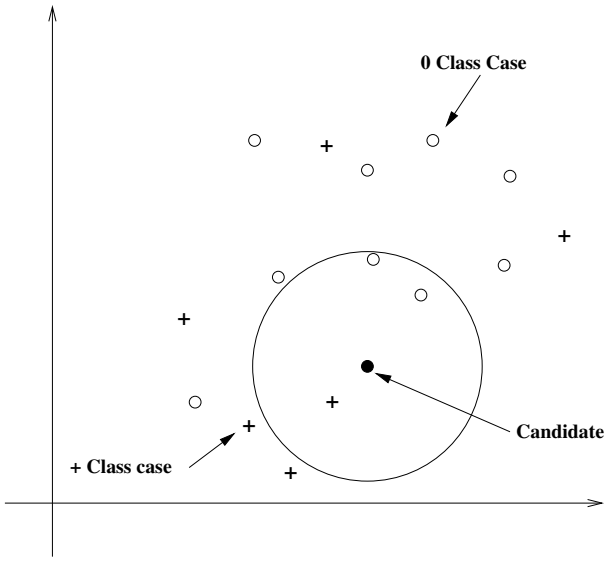

Figure 1: Third Nearest Neighbor Decision Rule

3. Related work

Much research has been devoted to the k-NN rule [8]. One of the most important results is that k-NN has a very good asymptotic performance. Broadly speaking, for a very large design set, the expected probability of incorrect classifications (error) R achievable with k-NN is bound as follows:

$$R^* < R < 2R^*$$

where R^* is the optimal (minimal) error rate for the underlying distributions. This performance, however, is demonstrated for the training set size tending to infinity, and thus, it is not really applicable to real world problems in which we usually have a training set of hundreds or thousands of cases, too few for the number of probability estimations to be performed.

Some distance based approaches, such as that of Weinberger et al. [31] try to increase the obtained accuracy in distance based classification by looking for a specific distance, in an automatic way, for each classification problem. The proposed approach could be used to deal with unbalanced or biased databases; a similar idea can be found in other distance based methods [29]. Alternatively, *PEBLS* instance based inducer (Cost and Salzberg [6]) incorporates MVDM distance metric to deal with symbolic features, a modification of Stanfill and Waltz's VDM metric [27].

Improvements in classification can also be achieved by selecting and/or weighting features (see [28] for an example). Probabilistic voting approaches have also been used ([21], [26]); the main idea here is that each case among the k nearest ones makes a weighted vote in favour of the class it belongs to, being the weight the probability each case has of belonging to its own class. A very well kown approach is the so called Instance Based Learning (IBL), based on the work of Aha [2] and Wettschereck [32]; there are several versions of the algorithm [1].

Another problem that arises with the distance based classifier systems is the management of large database. Studies devoted to data reduction [12] show interesting approaches which could also be used in combination with any other distance based algorithm when the size of the database is huge; there are as also studies that try to accelerate the execution of the distance based algorithm ([30], [18]) to obtain faster classifications.

There are other distance based classifiers which aim to deal with the so called multi labeling problem [33], classifying each case in several categories. For instance, and taking as an example the document categorization area, a newspaper article reporting the wedding of the president of a certain country would obtain "Politics" and "Society" as category labels, both being adequate for the document.

The next section is devoted to the new approach we present in this paper.

4. The *k* Nearest Neighbor Equality Method

The new approach can be seen as a new distance based classifier within the k-NN family of classifiers. Two are the main reasons of this new version: to decrease the influence of the outliers in the voting process, and to include all the classes as candidates in the final selection process.

4.1. Motivation to extend k-NN

Whereas k-NN rule offers good results in general, there are some situations where it could be improved. Let us consider the following situation with 2 classes where the observations are drawn from bivariate normal populations:

$$\mathbf{X}_{\theta_1} \sim N(\mu_1, \Sigma) \quad \text{and} \quad \mathbf{X}_{\theta_2} \sim N(\mu_2 \Sigma),$$

with $\mu_1 = (3, 0)'$, $\mu_1 = (-3, 0)'$ and variance-covariance matrix $\Sigma = \begin{pmatrix} 2^2 & 0 \\ 0 & 3^2 \end{pmatrix}$. We generated 40 cases from each class (see Figure 2). Let us consider case $\mathbf{x} = (1, -1.5)$ which is more likely belongs to class θ_1 than to class θ_2. Under 3-NN, the closest cases are indicated as shaded symbols and $\mathbf{x}$ would be classified in class θ_2. However, we can see, in general, cases from θ_1 (circles) are closer from $\mathbf{x}$ than cases from θ_2 (triangles). Summarizing, there are situations where some few cases from one class can "contaminate" the natural surroundings of the other class and the method we propose aims to overcome this situation. The main idea is to diminish the importance of outlier cases in the classification process.

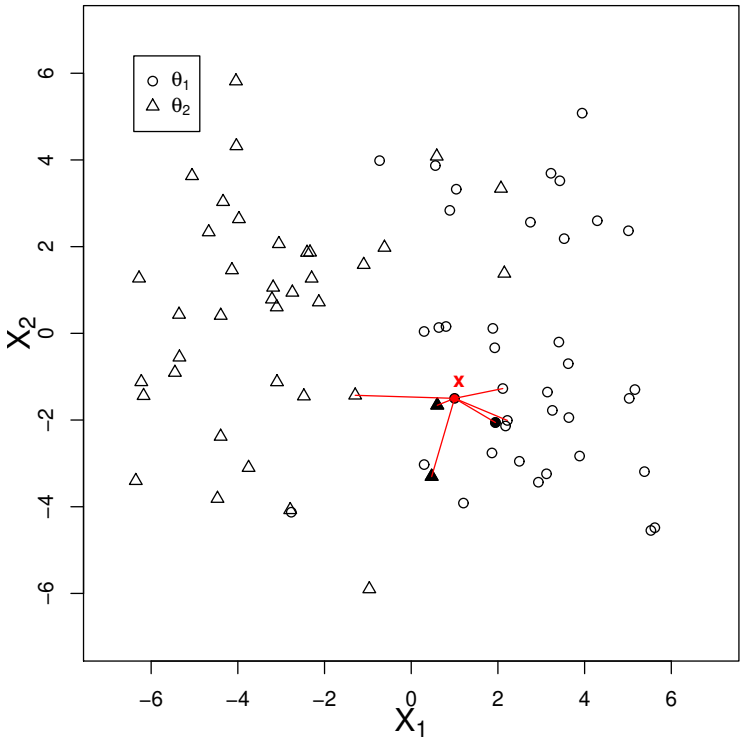

Figure 2: Synthetic data to show situations where k-NNE can overcome some shortcomings of k-NN. The 3 closest cases are indicated as shaded symbols. The 3 closest cases in each class are indicated with connecting segments.

We propose a modification of the k-NN algorithm that make good use of the information about the real class given by the values of the components of the observation $\mathbf{x}$. In view of the interest in k-NN, it is surprising that –to the best of the authors knowledge– the following generalization of the rule has not been investigated: given k, search for the k nearest neighbor cases to $\mathbf{x}$ in each class $\theta_1, \ldots, \theta_M$, and classify the case $\mathbf{x}$ to the class θ_{m^*} whose k nearest neighbor cases have the *minimum mean* distance to $\mathbf{x}$. That is, considering individuals $\mathbf{x}_{i_m^1}, \ldots, \mathbf{x}_{i_m^k}$ are the k NN for $\mathbf{x}$ in class θ_m, $m = 1, \ldots, M$,

$$\text{classify } \mathbf{x} \text{ in } \theta_{m^*} \text{ if } \quad \min_{m=1,\ldots,M} \left\{ \frac{1}{k} \sum_{l=1,\ldots,k} d(\mathbf{x}, \mathbf{x}_{i_m^l}) \right\} = \frac{1}{k} \sum_{l} d(\mathbf{x}, \mathbf{x}_{i_{m^*}^l}) \, .$$

We have called this rule k Nearest Neighbor Equality (k-NNE).

This method will be introduced and empirically investigated below. It will be shown that its application has better improvement than k-NN, especially for multi-class problems ($M > 2$ classes).

The new proposed method, k-NNE is shown in its algorithmic form in Figure 3. It is a simple method that obtains good results in multi-class problems. Although the computational cost seems to be expensive, it is very similar to the original k-NN computational costs, as it has to calculate the distances with respect to all the cases in the data base as well. Moreover, the obtained results are better than those obtained using other classification techniques (Classification Trees, Rule Induction, Instance Based Learning,...) used in the literature.

As it can be seen in Figure 4, the k-NNE approach works as follows: given a set of n correctly classified cases $(\mathbf{x}_1, \theta^1), (\mathbf{x}_2, \theta^2), \ldots, (\mathbf{x}_n, \theta^n)$ in a classification problem with M classes $\theta_1, \ldots, \theta_M$, given a new case to be classified $(\mathbf{x}, \theta)$ in which the class θ is unknown, and once a number k is fixed, the following classification process is performed: For each class θ_m, search for the k nearest neighbors to $\mathbf{x}$ among the cases belonging to class θ_m: $\mathbf{x}_{i_m^1}, \ldots, \mathbf{x}_{i_m^k}$ $(m = 1, \ldots, M)$.

begin k-NNE
As input we have the samples file, containing n cases $(\mathbf{x}_i, \theta^i)$, $i = 1, ..., n$,
the value of k and a new case $\mathbf{x}$ to be classified
FOR each class value θ_m DO
BEGIN
Select the k *nearest neighbors* to $\mathbf{x}$ that belong to θ_m from the sample file
Compute the mean distance from these k points to $\mathbf{x}$, $\bar{d}_{\theta_m}$
END
Output the class m^* with the minimal mean distance $\bar{d}_{\theta_{m^*}}$ among all the classes
end K-NNE

Figure 3: The pseudo-code of the k Nearest Neighbor Equality Algorithm.

For the k nearest neighbor cases of $\mathbf{x}$ in each class θ_m, compute the mean distance $\bar{d}_{\theta_m}$:

$$\bar{d}_{\theta_m} = \frac{1}{k} \sum_{l=1,\ldots,k} d(\mathbf{x}, \mathbf{x}_{i_m^l}), \quad m = 1, \ldots, M.$$

Assign case $\mathbf{x}$ to the class θ_{m^*} which has the minimum mean distance $\bar{d}_{\theta_{m^*}}$, that is,

$$\hat{\theta} = \theta_{m^*} \text{ such that } \theta_{m^*} = \arg\min \left\{ \bar{d}_{\theta_1}, \ldots, \bar{d}_{\theta_M} \right\}.$$

In principle, it would be possible to use another measure among the distances between the k NN in each class and case $\mathbf{x}$, for instance, the median. If the mean distance has been selected here, is because it can identify scenarios with outliers. That is to say, when some outliers of a class θ_m are close to cases of any other class θ_l, $(l \neq m)$ and far from cases of the class they belong to (i. e., θ_m), $\bar{d}_{\theta_m}$ could become big, and therefore, the influence of the outlier in the voting process is diminished.

As in the k-NN classification technique, a tie-break method must be implemented. In this first approach to the method, we break ties by using the prior probabilities of the classes, that is, selecting the most probable class.

Returning to the situation shown in Figure 2, the 3 NN to case $\mathbf{x}$ in each class θ_1 and θ_2 are indicated with connecting segments and the corresponding mean distances are $\bar{d}_{\theta_1} = 1.18$ and $\bar{d}_{\theta_2} = 1.27$. Hence, with 3-NNE case $\mathbf{x}$ will be classified in class θ_1 whereas by 3-NN it would be classified in class θ_2.

Figure 4 shows the behavior of the 3-NNE algorithm in comparison with the 6-NN for two two-class example problems. Although the computational cost is similar, the behavior of the 3-NNE method could make good use of the discriminant information provided by the predictor variables. This can be better seen in multi-class problems.

In Figure 4 we can see the difference between the two methods applied, although the result given in this case example is the same (Figure 4a), it is seen that the methods follow different procedures in the classification process (Figure 4b).

5. Experimental Results

The characteristics of the experimental files are given in Table 1. These domains are public at the Statlog project WEB page [19], and we have searched for some multi-class problems to compare the behavior of our algorithm.

5.1. Classifiers

Supervised classifiers [10], that come from different families are chosen. Seven well known inducers are used in the experiments:

- *ID3* decision tree algorithm (Quinlan [22]). It does not prune the expanded tree.
- *C4.5* decision tree algorithm (Quinlan [23]). Instead of *ID3*, it makes a *post-pruning* phase, based on *error based pruning* algorithm.

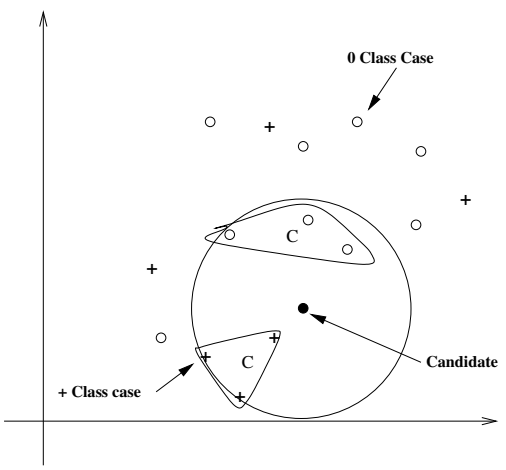

Figure 4a

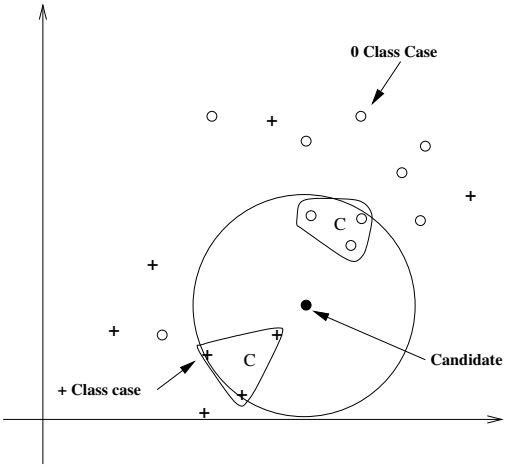

Figure 4b

Figure 4: 3-NNE Decision Rule compared with 6-NN. In Figure 4a, for a first classification problem, both the 3-NNE and 6-NN result in a tie, while in Figure 4b, for a second classification problem, the result of 3-NNE is the + class but the 6-NN has a tie.

- *Naive Bayes (NB)* algorithm (Cestnik [5]). It is based on Bayesian rules and, given that the value of the class is known, it assumes independence between the occurrences of feature values to predict the class.
- *Naive Bayes Tree (NBTree)* algorithm (Kohavi [16]). It runs *Naive Bayes* at the leaves of a induced decision tree. It normally improves *Naive Bayes* in large databases.
- *OneR* is a simple classifier that induces a set of rules, where each rule is based on only one predictor variable (the variable could be different in different rules) , i.e., a rules based on the value of a single attribute (Holte [14]).
- *CN2* rule induction classifier, based on the work of Clark and Nibblet [4]. It uses statistical tests to expand classification rules.
- *Neural Network* classifier based on the back-propagation algorithm[20]. Three layers have been used and a different number of intermediate neurons.

The class distribution of the training databases can be seen in Figure 5. Different distributions appear, some of them are uniformly distributed, while in other cases the distribution is biased.

5.2. Datasets

Ten databases are used to test our hypothesis. Most of them are obtained from the *UCI Machine Learning Repository* [3]. The *Nettalk* database was obtained from *MLC++* main repository [17]. All databases have a separate set of training data and testing data. The characteristics of the databases are given in Table 1.

We use the Training files as case examples for all the applied techniques, and the Test cases in order to estimate the error of the approach being used. Table 2 shows the experimental results obtained using standard Machine Learning approaches.

5.3. Distance Based Classifier comparison: Training and Test databases

In our experiments, we have run the k-NNE with different k values, and have compared the results obtained with those obtained with other methods implemented in standard Machine Learning software packages. In Table 3 we can see the results achieved with the k-NN method, while Table 4 shows the results obtained by the new proposed k-NNE method for some values of k. Obviously, the $k = 1$ case is equivalent to the NN method (or to the *1*-NN method).

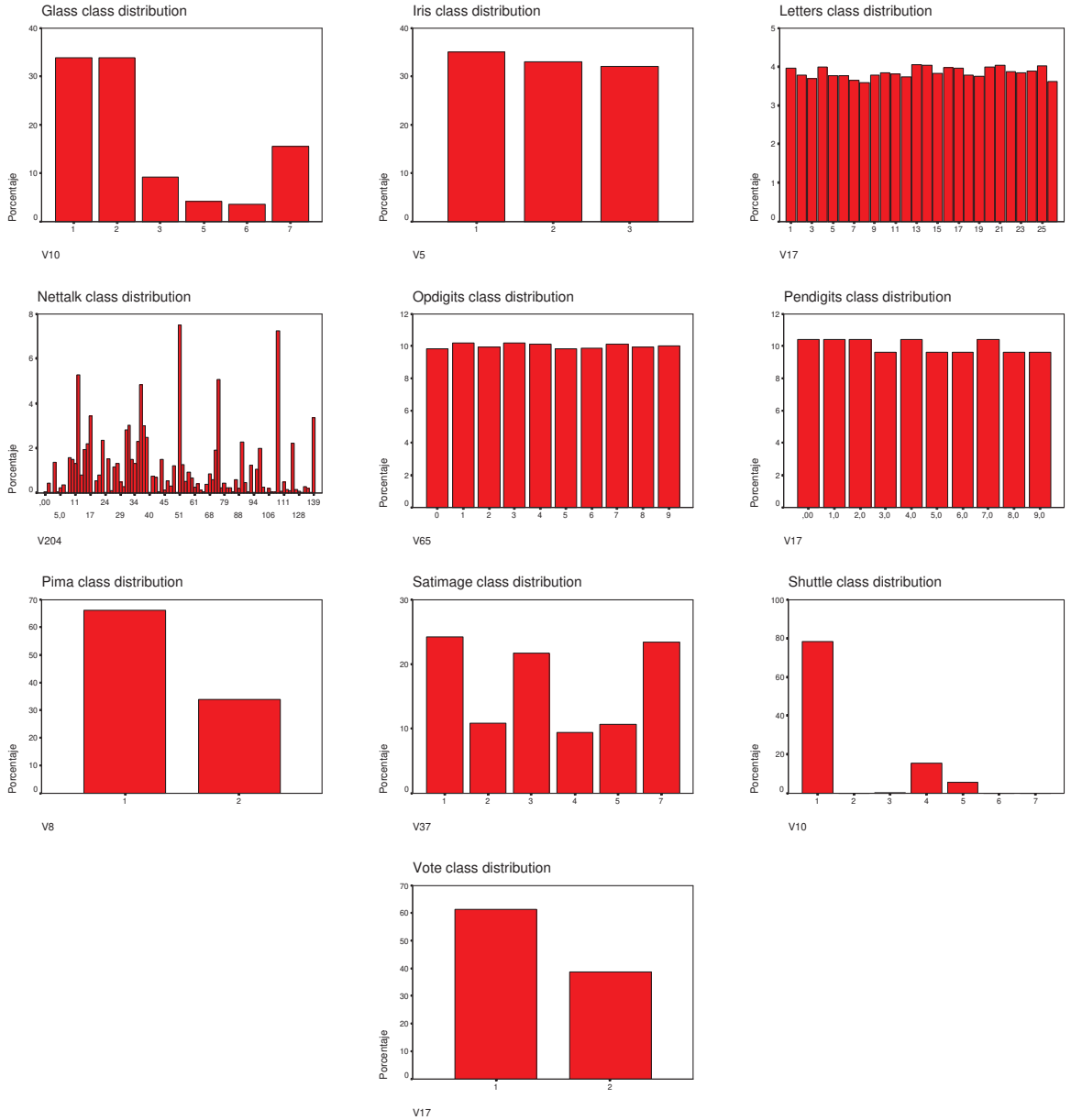

Figure 5: Class distributions of the training databases.

Table 1: Details of experimental domains

Domain	*Training cases*	*Test cases*	*Num. of classes*	*Num. of attributes*
Glass	142	72	7	9
Iris	100	50	3	4
Letters	15,000	5,000	26	16
Nettalk	7,229	7,242	324	203
Optdigit	3,823	1,797	10	64
Pendigit	7,494	3,493	10	16
Pima	200	332	2	7
satimage	4,435	2,000	7	36
shuttle	43,500	14,500	7	9
Vote	300	135	2	16

Table 2: Details of accuracy level percentages for the databases

Inducer	*Glass*	*Iris*	*Letters*	*Nettalk*	*Opdigit*	*Pendigit*	*Pima*	*Satimage*	*Shuttle*	*Vote*
ID3	62.50	94.00	76.65	72.52	54.14	91.51	71.71	84.80	99.99	94.17
C4.5	62.50	92.00	87.02	71.50	56.93	92.02	75.39	85.40	99.95	97.04
NB	50.00	96.00	63.20	60.99	82.63	82.22	78.01	79.65	87.61	91.85
NBTree	63.89	96.00	84.30	65.85	89.93	92.74	78.91	81.65	99.98	95.53
oneR	48.61	94.00	16.52	12.48	23.20	36.60	73.82	58.80	94.67	97.04
CN2	76.40	94.00	64.00	75.40	67.40	86.40	75.80	71.30	99.40	95.60
Neural Net	63.89	98.00	89.24	71.42	97.11	93.25	80.42	82.20	98.87	97.04
Best result	**76.40**	**98.00**	**89.24**	**75.40**	**97.11**	**93.25**	**80.42**	**88.80**	**99.99**	**97.04**

Table 3: Accuracy level percentage of the *k*-NN method for the databases using different *K* numbers

k	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
Glass	**81.94**	73.61	69.44	68.06	66.67	69.44	62.50	69.44	65.28	66.67
Iris	96.00	90.00	**98.00**	96.00	94.00	96.00	94.00	92.00	92.00	94.00
Letter	**95.62**	94.80	95.20	95.00	95.04	95.08	95.00	94.64	94.84	94.66
Nettalk	55.59	52.18	55.40	57.21	58.20	59.02	59.13	**59.67**	59.80	59.75
Opdigit	**98.00**	97.44	97.77	97.61	97.83	97.72	97.61	97.61	97.66	97.50
Pendigit	97.74	97.34	**97.77**	97.54	97.37	97.34	97.34	97.34	97.28	97.28
Pima	68.37	71.99	71.11	76.20	**78.92**	78.31	78.01	78.01	78.01	77.41
Satimage	89.45	88.95	**90.35**	90.25	90.35	89.80	89.85	89.25	89.40	89.50
Shuttle	**99.88**	99.81	99.83	99.81	99.80	99.78	99.79	99.77	99.79	99.75
Vote	**93.33**	92.59	91.85	92.59	93.33	93.33	93.33	91.85	92.59	92.59

Table 4: Accuracy level percentage of the *k*-NNE method for the seven databases using different *k* numbers

k	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
Glass	**81.94**	79.17	70.83	69.44	68.06	66.67	63.89	63.89	63.89	62.50
Iris	96.00	96.00	**98.00**	98.00	96.00	96.00	96.00	92.00	92.00	92.00
Letter	95.62	96.08	**96.28**	96.28	96.26	95.96	95.80	95.76	95.42	95.16
Nettalk	55.59	60.66	64.24	65.30	66.21	**66.53**	66.43	66.16	65.66	65.53
Opdigit	98.00	98.33	**98.39**	98.22	98.11	98.00	97.89	97.83	97.77	97.66
Pendigit	97.74	98.00	**98.03**	97.97	97.83	97.74	97.74	97.54	97.54	97.48
Pima	68.37	74.10	**76.51**	76.51	75.60	75.90	76.20	76.20	75.90	76.20
Satimage	89.45	90.65	**90.80**	90.60	90.75	90.55	90.15	90.00	89.95	90.00
Shuttle	**99.88**	99.86	99.85	99.85	99.85	99.82	99.79	99.79	99.77	99.77
Vote	**93.33**	92.59	92.59	93.33	93.33	93.33	92.59	93.33	93.33	93.33

As can be seen in Table 3 and Table 4, the new proposed method obtains a better accuracy than the k-NN method, especially in multi-class problems, i.e., in those problems in which the class number is high. For example, when applying the new method to the nettalk database (324 classes), the best accuracy result obtained is 66.53 in correctly classified percentage, while the best result obtained by the k-NN paradigm is 59.80. Although some other paradigms outperform this accuracy (ID3, 72.52, Neural Net 71.42), we are interested in how our proposed method compares with other distance based paradigms. When looking to the accuracy obtained with the letter database (26 classes), the k-NNE approach obtains a performance of 96.28%, while the k-NN best performance is 95.62%. For this database, both distance based paradigms outperform the results obtained with the other paradigms (as shown in Table 2).

This better performance of the presented approach with respect to the k-NN does not seem to hold when working with the databases whose class number is low (iris, vote); in both of them, the obtained performance is equivalent for the k-NN and k-NNE approaches.

In Table 5 the best results obtained for each database in each of the three previous experiments is presented in a summarized way.

Table 5: Best percentages obtained for the databases in each of the three classifier subsets considered.

Classifier	*Glass*	*Iris*	*Lett.*	*Nett.*	*Opd.*	*Pend.*	*Pima*	*Satim.*	*Shut.*	*Vote*
Standard ML	76.40	**98.00**	89.24	**75.40**	97.11	93.25	**80.42**	88.80	**99.99**	**97.04**
k-NN	**81.94**	**98.00**	95.62	59.80	98.00	97.77	78.92	90.35	99.88	93.33
k-NNE	**81.94**	**98.00**	**96.28**	66.53	**98.39**	**98.03**	76.51	**90.80**	99.88	93.33

We have applied classifier comparison statistical tests as recommended by Demsar [9], and no significance differences are obtained when either applying the Fisher-Snedecor test for the three rows or using the Wilcoxon rank test for each pair.

Another value we can compare is the mean accuracy obtained for all the ten values of the k parameter. These are presented in Table 6. As can be seen, on average the k-NNE method outperforms the standard algorithm in 8 of the 10 databases.

Table 6: Mean percentages of the k-NN and k-NNE methods for the ten values of k in each of the databases

Classifier	*Glass*	*Iris*	*Letters*	*Nettalk*	*Opdigit*	*Pendigit*	*Pima*	*Satimage*	*Shuttle*	*Vote*
k-NN	69.30	94.2	94.99	57.60	97.68	97.43	75.63	89.71	99.80	92.74
k-NNE	69.03	95.2	95.86	64.23	98.02	97.76	75.15	90.29	99.82	93.10

This comparison is made because selecting the best k value for the test databases is not considered sound in the Machine Learning classifier selection process.

In order to better understand the results obtained, and following Demsar [9], a comparison among all k values is performed; in order to compare k-NN and k-NNE, we choose the 20 different results given by 20 different classifiers. As shown in Table 7, the best rank mean (4.65) is obtained by the k-NNE algorithm with a k value of 4. It can also be seen that the mean ranks of k-NNE are better in general, and in fact five of them are better than the best mean rank obtained by k-NN (9.60 for $k = 1$). The obtained Fisher-Snedecor $F_{9,9}$ value is 4.53, which does allow to consider k-NNE, with k equal to 4, as the best algorithm over all the others.

5.4. Distance Based Classifier comparison: cross-validation approach

In order to make a better comparison between the k-NN algorithm and the new proposed k-NNE paradigm, we have carried out a more complete experiment dividing the data-files (which came as fixed training and test data files in the repository) into ten randomly generated training and test files, each of the same size as shown in Table 1.

The results obtained by the two algorithms are showed in Figure 6. As can be seen, the k-NN algorithm results are not as good as those obtained by the new proposed paradigm. The standard deviation is very low, and thus, the statistical testing (using Wilcoxon range testing) indicates, in almost all the cases, that the new paradigm is more accurate than the standard k-NN.

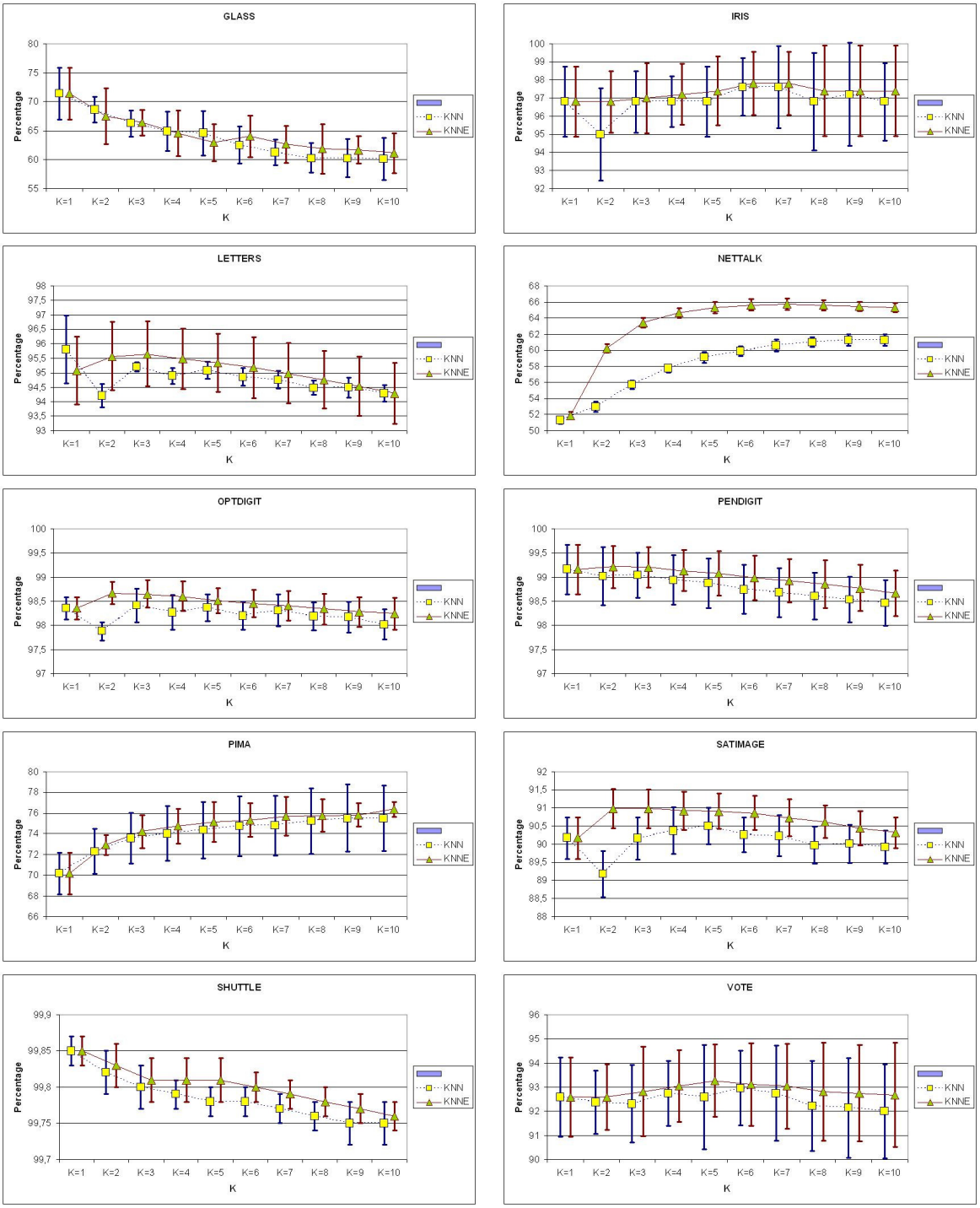

Figure 6: Mean Accuracy level percentage and standard deviation obtained by the *k*-NN and *k*-NNE methods for the databases using different *k* numbers.

Table 7: Mean percentages of the k-NN and k-NNE methods for the ten values of k in each of the databases

Cl	*k*	*Glass*	*Iris*	*Letters*	*Nettalk*	*Opdigit*	*Pendigit*	*Pima*	*Satimage*	*Shuttle*	*Vote*	*Mean*
k-NN	1	1.50	7.50	8.50	17.50	6.0	7.50	19.50	16.50	1.50	6.0	*9.20*
	2	4.0	20.0	18.0	20.0	20.0	16.50	17.0	20.0	9.50	15.0	16.0
	3	7.50	2.0	11.0	19.0	11.50	5.0	18.0	6.50	7.0	19.50	10.70
	4	10.50	7.50	15.50	16.0	17.0	11.0	10.50	8.0	9.50	15.0	12.05
	5	13.0	13.0	14.0	15.0	9.50	14.0	1.0	6.50	11.0	6.0	10.30
	6	7.50	7.50	13.0	14.0	13.0	16.50	2.0	14.0	16.0	6.0	10.95
	7	19.50	13.0	15.50	13.0	17.0	16.50	4.0	13.0	13.50	6.0	13.10
	8	7.50	17.0	20.0	12.0	17.0	16.50	4.0	19.0	18.0	19.50	15.05
	9	15.0	17.0	17.0	10.0	14.50	19.50	4.0	18.0	13.50	15.0	14.35
	10	13.0	13.0	19.0	11.0	19.0	19.50	6.0	15.0	20.0	15.0	15.05
k-NNE	1	1.50	7.50	8.50	17.50	6.0	7.50	19.50	16.50	1.50	6.0	9.20
	2	3.0	7.50	4.0	9.0	2.0	2.0	16.0	3.0	3.0	15.0	*6.45*
	3	5.0	2.0	1.50	8.0	1.0	1.0	7.50	1.0	5.0	15.0	*4.70*
	4	7.50	2.0	1.50	7.0	3.0	3.0	7.50	4.0	5.0	6.0	**4.65**
	5	10.50	7.50	3.0	3.0	4.0	4.0	15.0	2.0	5.0	6.0	*6.0*
	6	13.0	7.50	5.0	1.0	6.0	7.50	13.50	5.0	8.0	6.0	*7.25*
	7	17.0	7.50	6.0	2.0	8.0	7.50	10.50	9.0	13.50	15.0	9.60
	8	17.0	17.0	7.0	4.0	9.50	11.0	10.50	10.50	13.50	6.0	10.60
	9	17.0	17.0	10.0	5.0	11.50	11.0	13.50	12.0	18.0	6.0	12.10
	10	19.50	17.0	12.0	6.0	14.50	13.0	10.50	10.50	18.0	6.0	12.70

The results obtained for the data bases in which the problems have more classes indicate that the approach presented here could work better when classifications among more than two classes have to be carried out:

- Letters: 26 classes, k-NN obtains 95.19 ± 00.16 mean accuracy and k-NNE 95.65 ± 01.12.
- Nettalk: 324 classes, k-NN obtains 61.24 ± 00.72 mean accuracy and k-NNE 65.72 ± 00.66.
- Optdigit: 10 classes,k-NN obtains 98.41 ± 00.34 mean accuracy and k-NNE 98.67 ± 00.23.
- Pendigit: 10 classes, k-NN obtains 99.16 ± 00.51 mean accuracy and k-NNE 99.21 ± 00.44.

In the experiment carried out, it can also be seen that the two algorithms offer a similar accuracy, but that the one presented in this paper outperforms the accuracy in all the multi-class problems.

Once again, as proposed by Demsar in [9], a comparison among all k values is performed. As shown in Table 8, the best rank mean (5.60) is obtained by the k-NNE algorithm with a k value of 6. It can also be seen that the mean ranks of k-NNE are better in general than the mean rank obtained by k-NN (10.60 for $k = 3$). Moreover, the obtained Fisher-Snedecor $F_{9,9}$ value is 4.92, which leads us to consider k-NNE with k equal to 6 as the best algorithm among the rest.

6. Conclusion and Future Work

A new method extending the k-NN idea is presented in this paper. The new method, called k-NNE, is based on the idea that predictor variables could have a different probability distribution in each class, and consequently, in the classification process it searches for elements in the proximity of the new case to be classified belonging to each class.

This new method is used in different multi-class problems, and its final results are compared with those obtained by using the standard ML paradigms. We do not expect our new method to be better than that of k-NN in all the classification problems, but it works better in almost all the experiments we have performed, and the difference is more evident when the problem has more than two classes.

Also we would like point out that the new method has been presented in its simplest distance calculation approach, and compared with the same version of the original k-NN. A lot of extensions could be applied to the algorithm: the distance metric, the weight of the neighbors depending on its distance, the weighing of the variables in the distance calculation, and so on. Different versions can be designed to try to decrease the number of calculations to be performed to obtain the k nearest neighbors of a given case [25], [11]. Different techniques of prototype selection or/and attribute selection could also be used in some extensions of the new algorithm, as has be done with the k-NN algorithm [24]

Table 8: Mean percentages of the k-NN and k-NNE methods for the ten values of k in each of the databases

Cl	*k*	*Glass*	*Iris*	*Letters*	*Nettalk*	*Opdigit*	*Pendigit*	*Pima*	*Satimage*	*Shuttle*	*Vote*	*Mean*
k-NN	1	1.50	15.50	8.0	19.50	9.50	3.50	19.50	15.50	1.50	13.50	10.75
	2	3.0	20.0	20.0	18.0	20.0	8.0	18.0	20.0	4.0	16.0	14.70
	3	6.0	15.50	5.0	17.0	6.0	7.0	16.0	8.0	8.50	17.0	*10.60*
	4	7.0	15.50	11.0	16.0	14.0	10.0	15.0	11.0	10.50	9.0	11.90
	5	8.50	15.50	8.0	15.0	8.0	12.0	13.0	9.0	13.0	13.50	11.55
	6	13.0	3.50	12.0	14.0	16.0	15.0	11.50	13.0	13.0	5.0	11.60
	7	16.0	3.50	14.0	12.0	12.0	16.0	10.0	14.0	15.50	9.0	12.20
	8	18.50	15.50	17.0	11.0	17.0	18.0	8.0	18.0	17.50	18.0	15.85
	9	18.50	9.50	16.0	9.0	18.0	19.0	5.0	17.0	19.50	19.0	15.05
	10	20.0	15.50	18.50	10.0	19.0	20.0	6.0	19.0	19.50	20.0	16.75
k-NNE	1	1.50	15.50	8.0	19.50	9.50	3.50	19.50	15.50	1.50	13.50	10.75
	2	4.0	15.50	2.0	13.0	1.0	1.0	17.0	1.50	3.0	13.50	*7.15*
	3	5.0	11.0	1.0	8.0	2.0	2.0	14.0	1.50	6.0	6.50	*5.70*
	4	8.50	9.50	3.0	7.0	3.0	5.0	11.50	3.0	6.0	3.50	*6.0*
	5	11.0	6.50	4.0	5.0	4.0	6.0	9.0	4.0	6.0	1.0	*5.65*
	6	10.0	1.50	6.0	2.0	5.0	9.0	7.0	5.0	8.50	2.0	**5.60**
	7	12.0	1.50	10.0	1.0	7.0	11.0	4.0	6.0	10.50	3.50	*6.65*
	8	14.0	6.50	13.0	3.0	11.0	13.0	3.0	7.0	13.0	6.50	*9.0*
	9	15.0	6.50	15.0	4.0	13.0	14.0	2.0	10.0	15.50	9.0	*10.40*
	10	17.0	6.50	18.50	6.0	15.0	17.0	1.0	12.0	17.50	11.0	12.15

[15] [26]. The goal of these extensions would be to decrease the computation order of the algorithm in the distance calculation.

As further work we are going to apply different k-NN extensions to the k-NNE: weighting techniques, condensation methods, and editing approaches will be combined with the new proposed method in order to compare its behavior with that of the k-NN.

We are also collecting data from the Basque Country Weather Service in order to apply supervised classification techniques, including k-NNE, to the weather prediction task.

Acknowledgments

References

[1] D. Aha, D. Kibler and M.K. Albert (1991): "Instance-Based learning algorithms", *Machine Learning* **6**, 37-66

[2] D. Aha (1992): "Tolerating, irrelevant and novel attributes in instance-based learning algorithms", *International Journal of Man-Machine Studies* **36** (1), 267-287

[3] B.L. Blake and C.J. Merz (1998): "UCI Repository of Machine Learning databases", *[http://www.ics.uci.edu/~mlearn/MLRepository.html].* Irvine, CA: University of California, Department of Information and Computer Science

[4] P. Clark and T. Nibblet (1989): "The CN2 Induction Algorithm", *Machine Learning* **3** (4), 261-283

[5] B. Cestnik (1990): "Estimating Probabilities: A Crucial Task in Machine Learning", *Proceedings of the European Conference on Artificial Intelligence*, 147-149

[6] S. Cost and S. Salzberg (1993): "A Weighted Nearest Neighbor Algorithm for Learning with Symbolic Features", *Machine Learning* **10**(1), 57-78

[7] T. M. Cover and P. E. Hart (1967): "Nearest Neighbor Pattern Classification", *IEEE Trans. IT-13* **1**, 21-27

[8] B.V. Dasarathy (1991): *Nearest Neighbor (NN) Norms: NN Pattern Recognition Classification Techniques*, IEEE Computer Society Press

[9] J. Demsar (2006): "Statistical Comparisons of Classifiers over Multiple Data Sets", *Journal of Machine Learning Research* **7**, 1-30

[10] T.G. Dietterich (1997): "Machine Learning Research: four current directions", *AI Magazine* **18** (4), 97-136

[11] C. Domeniconi and D. Gunopulos (2001): "An Efficient Approach for Approximating Multi-dimensional Range Queries and Neares Neighbor Classification in Large Datasets", *Proceedings of the Eighteenth International Conference on Machine Learning ICML01*, 98-105

[12] H.A. Fayed and A.F. Atiya (2009): "A Novel Template Reduction Approach for the K-Nearest Neighbor Method", *IEEE Transactions on Neural Networks* **20** (5), 890-896

[13] E. Fix and J. L. Hodges Jr (1951): "Discriminatory Analysis, Nonparametric Discrimination", *USAF school of Aviation Medicine, Randolf field, Project 21-49-004, Rept 4*

[14] R.C. Holte (1993): "Very Simple Classification Rules Perform Well on Most Commonly Used Databases", *Machine Learning*, **11**, 63-90

[15] I. Inza, P. Larrañaga, R. Etxeberria and B. Sierra (2000): "Feature Subset Selection by Bayesian Network-Based Optimization", *Artificial Intelligence* **123**, 157-184.

[16] R. Kohavi (1996): "Scaling up the Accuracy of Naive-Bayes Classifiers: a Decision-Tree Hybrid", *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*

[17] R. Kohavi, D. Sommerfield and J. Dougherty (1997): "Data mining using MLC++, a Machine Learning Library in C++", *International Journal of Artificial Intelligence Tools* **6** (4), 537-566. *[http://www.sgi.com/Technology/mlc/]*

[18] Y.C. Liaw, M.L. Leou and C.M. Wu (2010): "Fast Exact K Nearest Neighbors Search Using an Orthogonal Search Tree", *Pattern Recognition* **43**, 2351–2358
[19] D. Michie, D.J. Spiegelhalter and C.C. Taylor (eds) *Machine Learning, Neural and Statistical Classification* http://www.amsta.leeds.ac.uk/~charles/statlog/
[20] T. Mitchell (1997): *Machine Learning*. McGraw-Hill.
[21] J. M. Martínez-Otzeta and B. Sierra (2004): "Analysis of the Iterated Probabilistic Weighted k-Nearest Neighbor Method, a New Distance-Based Algorithm",*6th International Conference on Enterprise Information Systems (ICEIS)* **2**, 233–240
[22] J.R. Quinlan (1986): "Induction of Decision Trees", Machine Learning **1**, 81-106
[23] J.R. Quinlan (1993): *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, Inc. Los Altos, California
[24] A. Rozsypal and M. Kubat (2001): "Using the Genetic Algorithm to Reduce the Size of a Nearest-Neighbor Classifier and to Select Relevant Attributes", *Proceedings of the Eighteenth International Conference on Machine Learning ICML01*, 449-456
[25] J. Schumacher and R. Bergmann (2000): "An Effective Approach for Similarity-Based Retrieval on Top of Relational Databases". 5th European Workshop on Case-Based Reasoning. Springer.
[26] B. Sierra and E. Lazkano (2002): "Probabilistic-Weighted K Nearest Neighbor Algorithm: a New Approach for Gene Expression Based Classification", *KES02 proceedings (IOS press)*, 932-939.
[27] C. Stanfill and D. Waltz (1986): "Toward Memory-Based Reasoning", *Communications of the ACM*, **29** (12), 1213-1228
[28] M.A. Tahir, A. Bouridane and F. Kurugollu (2007): "Simultaneous Feature Selection and Feature Weighting Using Hybrid Tabu Search/K-Nearest Neighbor Classifier", *Pattern Recognition Letters archive* **28**, 438-446
[29] S. Tan(2005): "Neighbor-Weighted K-Nearest Neighbor for Unbalanced Text Corpus", *Expert Systems with Applications* **28**, 667-671
[30] Q. Wang, S.R. Kulkarni and S. Verdu (2009): "Divergence Estimation for Multidimensional Densities Via K-Nearest-Neighbor Distances", *IEEE Transactions on Information Theory* **55**, 2392-2405
[31] K.Q. Weinberger and L.K. Saul (2009): "Distance Metric Learning for Large Margin Nearest Neighbor Classification", *The Journal of Machine Learning Research* **10**, 207-244
[32] D. Wettschereck (1994): *A Study of Distance-Based Machine Learning Algorithms*, Ph.D. Thesis, Oregon State University
[33] Z. Younes, F. Abdallah and T. Denœux (2008): "Multi-Label Classification Algorithm Derived from K-Nearest Neighbor Rule with Label Dependencies", *16th European Signal Processing Conference*