New neighborhood based classification rules for metric spaces and their use in ensemble classification

Jose-Norberto Mazón, Luisa Micó, and Francisco Moreno-Seco

Departamento de Lenguajes y Sistemas Informáticos
Universidad de Alicante
P.O. box 99, E-03080 Alicante, Spain
{jnmazon,mico,paco}@dlsi.ua.es
http://www.dlsi.ua.es

Abstract. The k-nearest-neighbor rule is a well known pattern recognition technique with very good results in a great variety of real classification tasks. Based on the neighborhood concept, several classification rules have been proposed to reduce the error rate of the k-nearest-neighbor rule (or its time requirements). In this work, two new geometrical neighborhoods are defined and the classification rules derived from them are used in several real data classification tasks. Also, some voting ensembles of classifiers based on these new rules have been tested and compared.

1 Introduction

Several scientific fields like pattern recognition, information retrieval, or data mining frequently use the same techniques for different purposes. For example, the k-Nearest Neighbor rule (k-NN) is often used in pattern recognition [1] for classification tasks. Also, the k-NN is used to obtain high performance data mining [2] [3], or efficient similarity retrieval of information [4].

Given a set T of n points that are labelled with J different labels $(\omega_1, \ldots, \omega_J)$, and given an unlabelled sample x, the k-NN rule R assigns to the sample x the most frequent label among the k points closest to x, i.e., if $K_i(x)$ is the number of points that are labelled with ω_i among the k nearest points to x, this rule can be defined as:

$$R(x) = \omega_i \quad \text{if} \quad K_i(x) = \max_{c=1\ldots J}\{K_c(x)\}$$

From a theoretical point of view, the k-NN rule error rate is low (and bounded by as much as twice the Bayes error), and usually the classification time of a k-NN based classifier is small (by using a fast k-NN search algorithm). However, in real data tasks the behavior of the k-NN rule is not usually as good. In the last years, a number of alternative neighborhood definitions have been proposed in the literature in order to reduce the error rate of the k-NN rule, or to speed up the classification [5, 6]. Some of the new alternative rules to reduce the error rate are

based on the use of the Gabriel and the Relative Neighborhood graphs [7]. There are also some surrounding rules, as the k Nearest Centroid Neighborhood rule, k-NCN, that classifies the sample using the neighbors whose centroid (mean) is closest to the sample. This rule looks for points that are not only close enough but also symmetrically distributed around a sample [5].

The k-NCN rule has been shown to give significantly better results than the classical k-NN approach in many real data tasks. However, this rule cannot be used in the general case, where objects are represented by data structures such as strings. For example, in the k-means clustering algorithm, the median[1] can be used instead of the mean when strings are used and the number of strings belonging to a cluster is high enough [2]. However, in the k-NCN rule the mean is computed for a relatively small number of points (between 1 and k, with $k << n$). In this case, the use of the median instead of the mean is a wrong option.

In this paper some alternatives to the k-NN and k-NCN rules have been defined, compared and tested experimentally with a database where the objects are represented as strings. The edit distance between strings [8] has been used to compare the objects. The proposed classification rules (based on new neighborhood definitions) are suitable for any classification task where a dissimilarity measure is defined. As a complement to this work, some classifier ensembles (some of them based on the rules proposed) have been tested.

In the following section, two different alternatives to the k-NN rule based on the concept of surrounding neighborhood are presented. Section 3 describes the different ensemble schemes for combining classifiers. Next, the results for the proposed classifiers and ensembles are presented and compared in real data tasks. Finally, the conclusions drawn from the results are discussed, pointing the research to further work lines.

2 New geometrical neighborhood definitions for metric spaces

The k-NN rule uses the neighborhood defined by the k closest points to an unlabelled sample to classify it. The k-NCN rule defines the neighborhood using the k neighbors whose mean is closest to the sample. Thus, a neighborhood definition has a corresponding classification rule that classifies the sample using the points belonging to the neighborhood. In this section, we propose two alternative neighborhood definitions based on the same type of information used in k-NCN rule: the distances and the geometrical distribution of points. The neighborhood definitions are proposed to overcome the problem of the representation of data in non vectorial spaces (i.e., metric spaces in general), that is, to select the surrounding points without computing the mean.

[1] Given a set of n points and a distance function, the median is defined as the point in the set that minimizes the sum of distances to the remaining points in the set.

[2] As the number of points increases, the diference between the median and the mean decreases.

1. The first neighbor of x is also its nearest neighbor, q_1.
2. To obtain the $i <= k$ point:
   (a) among the unselected points, the k' nearest neighbors to x are obtained;
   (b) among these k' points, the one whose sum of distances to the previously selected $i-1$ points is maximum is selected as a new neighbor, q_i
3. return to step 2 (increasing i) until k points are selected

Fig. 1. k-MMS neighborhood.

Given a set T of points, two different approaches have been defined. Both are incremental methods that use a new parameter (k', with $k' : 1 \dots k$); each new k_i surrounding point to a test sample x is selected in two steps:

1. a set $B \subset T$ with the k' nearest points to x is obtained;
2. k_i is selected among the points belonging to B.

2.1 *k*-Min Max Sum (*k*-MMS) neighborhood

In a neighborhood based classifier, the unlabelled sample x is classified using k neighbors that should be very close to x. However, in a surrounding neighborhood definition, each of the k neighbors should be far away from the previously selected neighbors, while at the same time they should be close to the sample.

The first surrounding neighborhood definition (see figure 1) is based on the incremental selection of the k nearest surrounding points using the ideas mentioned above. This rule is called *k-MinMaxSum* because the points whose sum of distances is maximum among the k' nearest points (minimum distance) to the sample are selected as new candidates to belong to the neighborhood.

2.2 *k*-Min Ranking Sum (*k*-MRS) neighborhood

In the second neighborhood definition, two vectors are used to store the $k' < k$ nearest points to the sample x (see figure 2):

1. $\boldsymbol{K_{min}}$ stores the k' points in increasing value of the distances to the sample x
2. $\boldsymbol{K_{max}}$ stores the points in decreasing value of the sum of distances to the previously selected points.

Then, the prototype whose sum of indexes in both vectors is minimum is selected to be included in the neighborhood. For example, if $\boldsymbol{K_{min}} = \{k_1, k_2, k_3, k_4\}$ and $\boldsymbol{K_{max}} = \{k_3, k_1, k_4, k_2\}$, the first selected point would be k_1, the second k_3, etc.

1. The first neighbor of x is also its nearest neighbor, q_1.
2. To obtain the $i <= k$ point:
   (a) the k' nearest neighbors to x are selected and ordered by their distances from the test sample (the nearest the first), $\boldsymbol{K_{min}}$;
   (b) the same selected k' points are ordered by the sum of distances to the previously selected $i-1$ points (the largest the first), $\boldsymbol{K_{max}}$;
   (c) the point whose sum of its indexes in both vectors is the lower, is selected as new neighbor, q_i.
3. return to step 2 (increasing i) until k points are selected

Fig. 2. k-MRS neighborhood.

3 Combining schemes

In order to increase the performance of single classifiers, a combination may be used instead [9]. In this paper, some known alternatives have been explored based on confidence methods and ranked voting methods using the proposed rules.

Confidence voting approaches. Confidence methods are based on the use of the confidence of classifiers about their preference for a candidate. In this case, a confidence value 1 means the higher preference in the decision. This preference is put into practice by assigning a confidence value to every possible class for each point. A wide range of different confidence methods can be used, based on distances or probabilities.

The confidence methods associated to each class c_i used in k selected points are defined to obtain a value in the range $[0, 1]$. If m_i is the number of neighbors among the selected k that belong to the class c_i ($m_i \leq k$):

$$conf(c_i)_{PROP} = \frac{m_i}{k}$$

$$conf(c_i)_{SD} = \frac{1}{1+\sum_{j=1}^{m_i} d_j/m_i}$$

where $\sum_{j=1}^{m_i} d_j$ is the sum of distances of the m_i neighbors belonging to the class c_i among the k selected neighbors.

Based on these definitions of confidence, three well known ensembles have been used in this work:

1. Pandemonium. Each single classifier gives a confidence value for each class. Then, the class whith the highest confidence value among the single classifiers is returned [10, 11].

2. Sum rule. As in the previous method, each classifier assigns a confidence value to each class. All confidence values associated to each class are added, and the class whose sum of confidences is the highest is assigned to the sample [13].

3. Product rule. Unlike the sum rule, in this case confidence values associated to each class are multiplied [13].

Ranked voting methods. In these approaches, single classifiers have to give a preference ranking of the class assigned to each point.

4. Borda count. This method was originally developed by Jean-Charles de Borda [12]. It has been adapted to classification problems in [11]. The only prerequisite is that each single classifier must return a complete preference ranking list of the possible classes. Then, the class with minor mean rank among all single classifiers is returned.

4 Experiments

Some experiments have been developed in order to study the performance of the new rules and the performance of the voting schemes presented in the previous section.

4.1 Neighborhood based rules

The experiments consist on a human chromosome classification task, where the objects (chromosomes) are represented as strings.

The Chromosome database [14–16] used for experiments contains 4400 samples (22 classes with 200 samples per class) coded as strings. The Levenshtein distance [8] has been used to measure the distance between chromosomes. The whole set has been divided into two sets of 2200 samples each, and the experiments have been performed using one of them for training and the other one for test. In a first experiment, the training set has been used to build training sets of different sizes. Table 1 shows the error rates for the proposed rules and the k-NN rule when different training set sizes have been used. This experiment was performed using different values of k, from 1 to 15. Due to the lack of space, only results for $k = 11$ are presented. After some tests, the value of k' used in k-MMS and k-MRS methods was set to 3 for all the experiments.

These experiments show that the k-MMS and k-MRS rules obtain results very similar to those of the k-NN rule. In particular, the k-MMS rule and the k-MRS rules outperform slightly the k-NN rule when the training set is not very small. Though the improvements are not very important, it can be observed that for increasing sizes of the training set, even if the error rate decreases quickly with the size of the training set, the proposed rules reduce the error rate.

In the following experiment, a fixed training set size was used (2200 chromosomes) for different k-values in the classifiers that use the rules (see table 2). As the previous experiment, the proposed rules reduce (slightly) the error rate.

Training set size	k-**NN**	k-**MMS**	k-**MRS**
220	37.81	39.18	**37.50**
660	20.40	20.40	**20.13**
1100	11.90	13.09	**11.54**
1540	8.27	8.31	**8.18**
1980	6.77	**6.40**	6.72
2200	6.59	**6.18**	6.27

Table 1. Error rates (in %) of the different classification rules with the Chromosome data sets using different training set sizes.

k	k-**NN**	k-**MMS**	k-**MRS**
1	10.09	10.09	10.09
3	8.23	9.59	**8.13**
5	6.90	7.40	**6.72**
7	6.68	7.40	**6.54**
9	6.77	7.00	**6.54**
11	6.59	**6.18**	6.27
13	6.45	6.09	**6.00**
15	6.68	**6.27**	6.36

Table 2. Error rates (in %) of the different classification rules with the Chromosome data sets using different values of k for a fixed training set size of 2200 chromosomes.

4.2 Ensemble classification

The combinations tested were:

- two combinations of 2 classifiers: k-NN and k-MMS rules (C1), and k-NN and k-MRS rules (C2)
- two combinations of 6 classifiers: 3 k-NN and 3 k-MMS rules (C3), and 3 k-NN and 3 k-MRS rules (C4) (varying the value of k).

Table 3 shows the results of these experiments using the confidence voting approaches $conf(c_i)_{PROP}$ and $conf(c_i)_{SD}$. As the best results were obtained by the C1 and C3 combinations, the experiments with the confidence $conf(c_i)_{SD}$ were only developed for these two combinations.

Finally, an experiment using C1 and C3 (with the sum rule) has been performed using different sizes of the training set and the confidence method $conf(c_i)_{SD}$. The comparison with k-NN is presented in figure 3. This figure shows that better results can be obtained in general for different training set sizes, but the best results are achieved when the training set is small.

	$C1_{PROP}$	$C2_{PROP}$	$C3_{PROP}$	$C4_{PROP}$	$C1_{SD}$	$C3_{SD}$
Pandemonium	6.18	6.31	6.09	6.18	8.13	6.18
Sum rule	5.90	6.27	5.50	5.86	6.68	**5.36**
Product rule	5.95	6.27	8.13	8.45	6.63	8.00
Borda count	6.81	6.45	6.27	6.50	6.82	6.27
k-**NN**	6.59					

Table 3. Error rates (in %) of the different ensembles with the Chromosome data sets using the confidence voting method $conf_{PROP}$ and $conf_{SD}$.

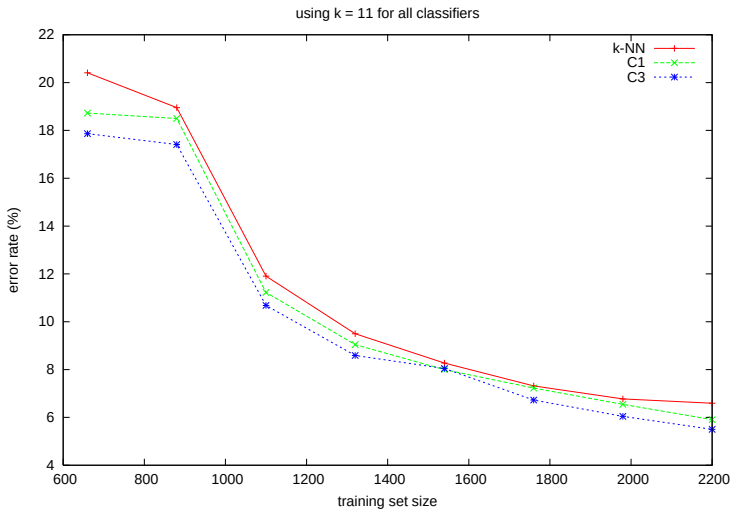

Fig. 3. Error rates (in %) of the different ensembles with the Chromosome data sets using different sizes of the training set and the confidence method $conf_{SD}$.

5 Conclusions

The k-NN rule is often used in classification tasks. However, sometimes the results may be improved if another neighborhood definition is used instead, as for instance the k-NCN rule. The main drawback of this rule is that it requires a vector space representation of data. In this work, two alternative neighborhood definitions that do not require a vector space are presented and the corresponding classification rules are tested in a real data task. The experimental results with the Chromosome database show that the proposed rules outperform the k-NN rule.

Moreover, the experiments with several ensembles of classifiers show that the proposed classification rules may perform adequately in a combination scheme. Future work includes a more exhaustive study of the rules with other real data tasks to know better their possibilities.

6 Acknowledgements

This work has been supported in part by grant DPI2006-15542-C04-01 from the Spanish CICYT (Ministerio de Ciencia y Tecnología), GV06/166 from Generalitat Valenciana, and the IST Programme of the European Community, under the Pascal Network of Excellence, IST-2002-506778.

References

1. Duda, R., Hart, P., Stork, D.: Pattern Classification, second edition. Wiley (2001)
2. Bohm, C., Krebs, F.: High performance data mining using the nearest neighbor join. Second IEEE International Conference on Data Mining (2002) 43–50.
3. Dasarathy, B.V.: Data mining tasks and methods: Classification: nearest-neighbor approaches. In Handbook of data mining and knowledge discovery, Oxford University Press (2002) 288–298.
4. Katayama, N., Satoh, S.: Distinctiveness-Sensitive Nearest-Neighbor search for efficient similarity retrieval of multimedia information. In proceedings of the 17th International Conference on Data Engineering (2001) 493–502.
5. Sanchez, J.S.; Pla, F., Ferri, F.J.: On the use of neighbourhood-based non-parametric classifiers. Pattern Recognition Letters 18, pp.1179–1186, 1997.
6. Moreno-Seco, F., Micó, L., Oncina, J.: Extending fast nearest neighbour search algorithms for approximate k-NN classification. Pattern Recognition and Image Analysis. Lecture Notes in Computer Science, F.J. Perales et al (Eds.) vol. 2652, Springer-Verlag (2003) 589–597.
7. Jamonczyk, J.W., Toussaint, G.T.: Relative neighbourhood graphs and their relatives. Proceedings IEEE, vol 80 (1992), 1502-1517.
8. Wagner, R.A., Fischer, M.J.: The String-to-String Correction Problem. Journal of the Association for Computing Machinery (1974) **21**(1) 168–173.
9. Kuncheva, L.: Combining Pattern Classifiers: Methods and Algorithms. Wiley InterScience, 2004.
10. Selfridge, O.: Pandemonium: a paradigm for learning in mechanisation of thought processes. In Proceedings of a Symposium Held at the National Physical Laboratory (1958), 513–526.
11. Van Erp, M., Vuurpijl, L.G., Schomaker, L.R.B.: An overview and comparison of voting methods for pattern recognition. In Proceedings of the 8th International Workshop on Frontiers in Handwriting Recognition (2002), 195–200.
12. J.-C. d. Borda. Memoire sur les Elections au Scrutin. Histoire de l'Academie Royale des Sciences, Paris, 1781.
13. Duin, R.: A theoretical study of six classifier fusion strategies. IEEE Transactions on Pattern Analysis and Machine Intelligence, **24**(2) (2002), 281–286.
14. Lundsteen, C., Phillip, J., Granum, E.: Quantitative analysis of 6985 digitized trypsin G-banded human metaphase chromosomes. Clinical Genetics (1980) **18** 355–370
15. Granum, E., Thomason, M.G., Gregor, J.: On the use of automatically inferred Markov networks for chromosome analysis. In Automation of Cytogenetics, C. Lundsteen and J. Piper, eds., Springer-Verlag (1989) 233–251
16. Granum, E., Thomason, M.G.: Automatically inferred Markov network models for classification of chromosomal band pattern structures. Cytometry (1990) **11** 26–39