Improving a dynamic ensemble selection method based on oracle information

Leila Maria Vriesmann* and Alceu de Souza Britto Jr.

State University of Ponta Grossa (UEPG),
Av. General Carlos Cavalcanti, 4748,
Ponta Grossa (PR), 84030-900, Brazil
and
Pontifical Catholic University of Parana (PUCPR),
R. Imaculada Conceição, 1155,
Curitiba (PR), 80215-901, Brazil
E-mail: lmvriesmann@uepg.br
E-mail: alceu@ppgia.pucpr.br
*Corresponding author

Luiz Eduardo Soares de Oliveira

Federal University of Parana (UFPR),
Av. Cel. Francisco H. dos Santos, s/n,
Curitiba (PR), 81530-900, Brazil
E-mail: lesoliveira@inf.ufpr.br

Robert Sabourin and Albert Houng-Ren Ko

École de Technologie Supérieure (ÈTS) – University of Quebec,
1100, Notre-Dame West,
Montréal (QC), H3C1K3, Canada
E-mail: robert.sabourin@etsmtl.ca
E-mail: albert@livia.etsmtl.ca

Abstract: This work evaluates some strategies to approximate the performance of a dynamic ensemble selection method to the oracle performance of its pool of weak classifiers. For this purpose, we evaluated different distance metrics in the K-nearest-oracles (KNORA) method, the use of statistics related to the class accuracy of each classifier in the pool and some additional information calculated by using a clustering process in the validation dataset. Moreover, different strategies are also evaluated to combine the results of the KNORA dynamic ensemble selection method with the results of its built-in K-nearest neighbour (KNN) used to define the neighbourhood of a test pattern during the ensemble creation. A strong experimental protocol based on more than 60,000 samples of handwriting digits extracted from NIST-SD19 was used to evaluate each strategy. The experiments have shown that the fusion of the KNORA results with the results of its built-in KNN is a very promising strategy.

Keywords: dynamic ensemble selection; oracle; K-nearest-oracles; KNORA.

Reference to this paper should be made as follows: Vriesmann, L.M., de Souza Britto, A., Jr., de Oliveira, L.E.S., Sabourin, R. and Ko, A.H-R. (2012) 'Improving a dynamic ensemble selection method based on oracle information', *Int. J. Innovative Computing and Applications*, Vol. 4, Nos. 3/4, pp.184–200.

Biographical notes: Leila Maria Vriesmann received her MSc in Informatics from the Federal University of Parana, Brazil in 2006. In 2007, she started her PhD in Informatics in Pontifical Catholic University of Parana (PUC-PR), Brazil. Her research interests are ensemble classification methods and pattern recognition.

Alceu de Souza Britto Jr. received his MSc in Industrial Informatics from the Federal Centre for Technological Education of Parana, Brazil in 1996, and PhD in Computer Science from Pontifical Catholic University of Parana (PUC-PR), Brazil in 2001. In 1989, he joined the Computer Science Department of the Ponta Grossa University, Brazil. In 1995, he also joined the Computer Science Department of the PUC-PR. His research interests are in the areas of pattern recognition, document analysis and handwriting recognition.

Luiz Eduardo Soares de Oliveira received his BS in Computer Science from UnicenP, Curitiba, PR, Brazil, MSc in Electrical Engineering and Industrial Informatics from the Centro Federal de Educacao Tecnologica do Parana (CEFET-PR), Curitiba, PR, Brazil, and PhD in Computer Science from Ecole de Technologie Superieure, Universite du Quebec in 1995, 1998, and 2003, respectively. His current interests include pattern recognition, neural networks, image analysis, and evolutionary computation.

Robert Sabourin received his BIng, MScA and PhD in Electrical Engineering from the Ecole Polytechnique de Montreal in 1977, 1980 and 1991, respectively. In 1977, he joined the Physics Department of the Universite de Montreal. In 1983, he joined the staff of the Ecole de Technologie Superieure, Universite du Quebec, Canada. In 1995, he also joined the Computer Science Department of the Pontificia Universidade Catolica (PUC-PR), Brazil. Since 1996, he has been a senior member of the Centre for Pattern Recognition and Machine Intelligence (CENPARMI). His research interests are in the areas of handwriting recognition and signature verification.

Albert Houng-Ren Ko received his MScA in Artificial Intelligence and Pattern Recognition from the Universite Pierre et Marie Curie in 2002. In 2007, he received his PhD in Pattern Recognition in Ecole de Technologie Superieure, Universite du Quebec. His research interests are ensemble classification methods, small world structure and neural networks.

This paper is a revised and expanded version of a paper presented at the 17th International Conference on Systems, Signals, and Image Processing (IWSSIP), Rio de Janeiro, 17–19 June 2010.

1 Introduction

The classification task is used in different applications in image processing (Silva et al., 2009; Nicoletti and Bertini, 2007). The objective of a classification algorithm is to assign a specific class among different possible ones to a given instance not previously seen (unknown sample). To allow a classifier to perform this non-trivial task with high accuracy, different aspects must be considered during the classifier construction, such as: the complexity of the problem; the feature extraction, the quality of the training set; the dimensionality of the feature space; the number of involved classes; and the amount of training samples (Ko, 2007).

Ranawana (2006) cited that, often, the construction of a perfect classifier for any task is almost impossible. Thus, an alternative has been to construct an ensemble of them. This is based on the idea that different classifiers make errors on different samples (Ko et al., 2008). It has been investigated by Brown et al. (2005), Kittler et al. (1998), Kuncheva and Whitaker (2003), Opitz and Maclin (1999), Pekalska et al. (2004), Webb and Zheng (2004) and Zouari et al. (2004). The use of more than one classifier is named on the literature as multiple classifier system (MCS) or ensemble of classifiers (EoC). The researches have shown that the diversity among the classifiers of an ensemble contributes to improve the general classification accuracy. The diversity may be obtained by using different machine learning algorithms, different organisation of the training data, or even different configuration of the feature space for the same problem.

In EoC, from an initial pool of trained classifiers, it is possible to select the classifier, or a subset of classifiers, that will be used to classify a given unknown sample. Two strategies have been used:

a the static selection, where the best classifier (or subset of classifiers) for all samples is selected from the initial pool

b the dynamic selection, where for each unknown sample a specific classifier (or subset of classifiers) that seems to be more appropriated is selected.

In this paper, we focus on the dynamic selection. Different approaches have been proposed for dynamic classifier selection in the literature (Cao et al., 1995; Didaci et al., 2005; Didaci and Giacinto, 2004; Giacinto and Roli, 1999; Woods et al., 1997). In a dynamic selection, the classifier is chosen and assigned to the sample based in different features or different decision regions. Popular examples of these methods are a priori selection, a posteriori selection, overall local accuracy (LCA) and local class accuracy (OCA) (Didaci et al., 2005; Didaci and Giacinto, 2004; Giacinto and Roli, 1999; Woods et al., 1997). Recently, we can find interesting methods for dynamic selection of EoCs, as follows:

- The dynamic overproduce-and-choose strategy (Santos et al., 2008), which is divided into two phases: a phase of overproduction where it is generated a pool of candidate classifiers and the selection phase, where different combinations of subsets removed from the pool are tested.
- The use of accuracy and diversity to build ensembles (Santana et al., 2006), where the classifiers (in a group of neighbours) are ordered in decreasing order of accuracy (rank) and in increasing order of diversity. It is selected the classifiers more diverse among the classifiers more accurate.

- The adaptive classifier ensemble selection based on group method of data handing (Xiao and He, 2009), in which a classifier ensemble is selected for each test pattern from the initial pool of classifiers and also the combination weights among the classifiers in the ensemble is determined.
- The linear random oracle (Kuncheva and Rodrígues, 2007), where each classifier in the set has a subset with two classifiers and an oracle, which is a random linear function. When a new object comes to be classified, the oracle to a respective classifier decides which subclassifier use. Then, the class chosen for each subclassifier is subjected to the rule of combining results.
- The *K*-nearest-oracles (KNORA) method (Ko et al., 2008), which considers the neighbourhood of the test patterns in the validation set to select the classifiers to compose the ensemble.

The aforementioned methods select a subset of classifiers from an initial pool. In such schemes, the ideal selection should always provide the correct subset of classifiers, i.e., the subset of classifiers that recognises the class of the unknown sample, if any. This is related to the oracle concept, which can be used as the possible upper limit of classification accuracy, defined as the ratio of samples that is correctly classified by at least one classifier in the pool to all samples. Thus, methods of dynamic selection usually have their performances compared with the oracle accuracy.

In this paper, we focus in the investigation of some strategies to approximate the performance of a dynamic ensemble selection method to the oracle performance of its pool of weak classifiers. Proposed in Ko et al. (2008), the KNORA uses a *K*-nearest neighbour (KNN) to find the *K* nearest neighbours of a test pattern (to be recognised) in a feature space, where we know the classifiers of the pool that correctly classify each sample. Then, different strategies are used to select these classifiers to compose an ensemble that will be used to classify the test pattern. The selection strategies are related with the ability of each classifier in recognising the neighbours of the test sample.

The results reported in Ko et al. (2008) were very promising. Therefore, our investigation considers the KNORA method applied in a handwriting numeral recognition problem. The objective is to evaluate different strategies trying to improve the recognition rates of the KNORA method, based on oracle information. The following strategies are investigated:

- Since the KNORA method uses the Euclidian distance to define the neighbourhood of a test sample, the first question to be answered is: How the use of different distance metrics in KNORA method may affect the recognition rates? To answer this question the use of different distance metrics will be investigated.
- Since a classifier in the initial pool is chosen to compose the ensemble to recognise a specific sample based on its accuracy in recognise the sample neighbourhood, another point that can be investigated is related to the class recognition accuracy of each classifier in the pool. Here, the questions are:
  - a Are the statistics related to the class recognition accuracy calculated for the pool of classifiers important in the classification task of KNORA?
  - b May the use of a clustering scheme, where each test sample may be addressed to a cluster, for which we know the frequency of each class, be used as additional information to improve the KNORA performance?
- Finally, may the use of additional information provided by the KNN used to select the *K* nearest neighbours of the test pattern in the KNORA method be interesting to improve the KNORA final results? To answer this question, we will evaluate different strategies to combine the KNN used to select the *K* nearest neighbours of the test pattern in the KNORA method with the final KNORA results.

This paper is organised into five sections. Section 2 presents the KNORA ensemble selection method and its schemes. Section 3 describes the four strategies evaluated: different distance metrics; additional information about the classifier accuracy; additional information about the class frequency inside the clusters; and the fusion of KNN and KNORA. The experimental results are presented in Section 4, while the conclusion and further works may be found in Section 5.

2 KNORA dynamic ensemble selection method

The KNORA is a dynamic ensemble selection method proposed by Ko et al. (2008). It considers the neighbourhood of the test pattern in the validation set to select the classifiers to compose the ensemble. For a given test pattern, it locates the *K* neighbours in the validation set. Since we know which classifiers in the pool can recognise each sample in the validation set, an EoCs can be dynamically selected to label the given test pattern. Different schemes have been proposed:

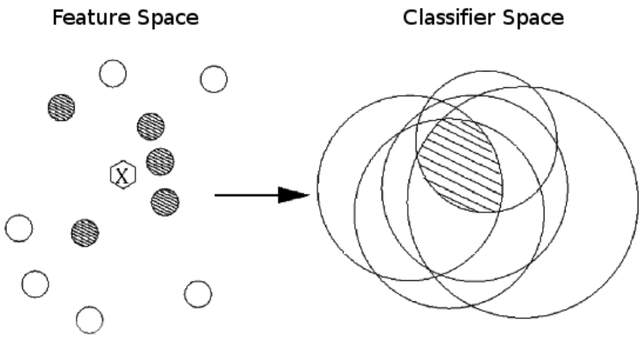
- KNORA-Eliminate: given *K* neighbours x_j, $1 \leq j \leq K$, of a test pattern *X*, and supposing $C(j)$, $1 \leq j \leq K$, a set of classifiers that correctly classifies all its KNNs, then every classifier $c_i \in C(j)$ should submit a vote on the sample *X*. In the case where no classifier can correctly classify all the KNNs of the test pattern, find the classifier that correctly classifies more neighbours in *K*. Then, only use the classifiers that recognise this number of neighbours. On the left side in Figure 1, test pattern is shown as a hexagon, validation data points are shown as circles and the five nearest validation points are darkened. On the right side, the used classifiers in the intersection of correct classifiers are darkened.
- KNORA-Union: given *K* neighbours x_j, $1 \leq j \leq K$, of a test pattern *X*, and supposing that the *j*-nearest neighbour has been correctly classified by a set of

classifiers $C(j)$ $1 \leq j \leq K$, then every classifier $c_i \in C(j)$ should submit a vote on the sample X. Note that, since all the KNNs are considered, a classifier can have more than one vote if it correctly classifies more than one neighbour. The more neighbours a classifier recognise, the more votes this classifier will have for a test pattern. On the left side in Figure 2, test pattern X is shown as a hexagon, validation data points are shown as circles, and the five nearest validation points are darkened. On the right side, the used classifiers in the union of correct classifiers are darkened.

- KNORA-Eliminate-W: this scheme is the same as KNORA-Eliminate, but each vote is weighted by the Euclidean distance between the neighbour pattern x_j and the test pattern X.
- KNORA-Union-W: this scheme is the same as KNORA-Union, but each vote is weighted by the Euclidean distance between the neighbour pattern x_j and the test pattern X.

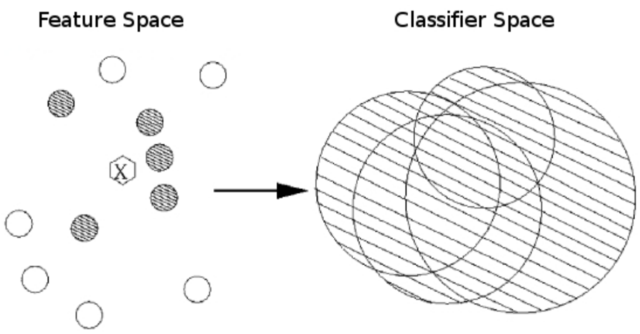
The KNORA method apparently gives better performances than static ensemble selection schemes such as genetic algorithm (GA) with the majority voting error (MVE) as the objective function and also perform slightly better than other dynamic selection methods as reported in Ko et al. (2008).

Figure 1 KNORA-Eliminate



Source: Ko et al. (2008)

Figure 2 KNORA-Union



Source: Ko et al. (2008)

3 Proposed strategies

Four strategies to improve the KNORA method were evaluated. The idea is to achieve the oracle performance. The first strategy uses different distance metrics (Section 3.1) to define the neighbourhood of a test sample in the KNORA method. The distance metrics evaluated are: Camberra distance, Cosine distance and Pearson distance. The second strategy uses additional information about the class accuracy of each classifier in the pool (Section 3.2), while the third strategy uses clustering and the corresponding statistics about the class frequency at each cluster as additional information in the KNORA method (Section 3.3). Finally, Section 3.4 describes the fusion of the KNORA and its build-in KNN method.

3.1 The use of different distance metrics

As described before, the KNORA method applies the Euclidian distance to find the K neighbours of a test sample. Then, the neighbours are used to select the corresponding classifiers that will compose the ensemble. The idea is that different distance measures may define different neighbourhoods, then, different ensembles will be selected. This section describes the distance measures evaluated: Camberra distance (Section 3.2.1), Cosine distance (Section 3.2.2) and Pearson distance (Section 3.2.3). The obtained results are shown in Section 4.3. These distance metrics were selected since they are frequently used in the literature and also represent distinct approaches to calculate the distance between two feature vectors.

3.1.1 Camberra

The Camberra distance (d_C) between two elements, A and B, can be expressed by means of equation (1).

$$d_C(A,B) = \sum_{i=1}^{n} \frac{|a_i - b_i|}{|a_i| - |b_i|} \tag{1}$$

where

A is a test sample and B is a validation sample

n is the number of features

a_i is the value of the i^{th} feature of the test sample A

b_i is the value of the i^{th} feature of the validation sample B.

3.1.2 Cosine

The Cosine distance calculates the angle between the test sample and its neighbour. It is defined by equation (2).

$$d_{COS}(A,B) = 1 - \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2 \cdot \sum_{i=1}^{n} b_i^2}} \tag{2}$$

where

A is a test sample and B is a validation sample

n is the number of features

a_i is the i^{th} feature of the test sample

b_i is the i^{th} feature of the validation sample.

3.1.3 Pearson

The Pearson distance is derived from the Pearson correlation coefficient and it is measured by equation (3).

$$d_{pcc}(A,B) = 1 - |p| \tag{3}$$

where $|p|$ is the Pearson correlation coefficient, defined by means of equation (4).

$$p = \frac{n\sum_{i=1}^{n} a_i b_i - \left(\sum_{i=1}^{n} a_i\right)\left(\sum_{i=1}^{n} b_i\right)}{\sqrt{\left[n\sum_{i=1}^{n} a_i^2 - \left(\sum_{i=1}^{n} a_i\right)^2\right]\left[n\sum_{i=1}^{n} b_i^2 - \left(\sum_{i=1}^{n} b_i\right)^2\right]}} \tag{4}$$

where

n is the number of features

a_i is the i^{th} feature of the test sample

b_i is the i^{th} feature of the validation sample.

3.2 The use of classifier accuracy

In this strategy, we calculate the classifier accuracy for each class on the validation set. It means the probability that each element of the pool of classifiers has to correctly recognise each class. In fact, this gives us a rank of classifiers per class accuracy. This accuracy value is used in KNORA-Union and in KNORA-Eliminate as additional information. After the classifiers have been selected for the ensemble, their votes are weighed by their corresponding class accuracy value, i.e., the value used to create the rank of classifiers per class. Section 4.4 presents the experiments where the classifier accuracy is used to weight the vote in the KNORA method.

3.3 The use of clustering

The clustering is an unsupervised classification of samples into groups (Jain et al., 1999). Clustering techniques aim to train, automatically, data groups according to some criterion of similarity, which depends on the algorithm and the problem to be treated.

The *K*-means (MacQueen, 1967; Hartigan, 1975) is a clustering technique by simply partitions that try to find *K* different clusters. The clusters are represented by their centroids. Usually, the centroid is the average of the points in the cluster. Since in this work we already use a *K* variable in the KNORA method, we renamed the *K*-means method for *U*-means, where *U* is the number of clusters. By applying this clustering method to the validation set, we expect to obtain some additional information related to the groups or clusters inside the KNORA method. The information that will be used is the frequency of each class observed inside the clusters (see Algorithm 1). The idea is that: since inside the cluster where the unknown sample belongs to, it is possible to observe the predominance of some classes, we may consider it inside the selected ensemble. For this purpose, the class frequency inside the cluster is used as weight in the voting scheme of the selected ensemble.

Algorithm 1 Compute the frequency of each class per cluster

```
Input:  A validation dataset V, where E_n is the sample
        number n in the dataset
        A cluster i for each sample E_n
Output: CFC(i, cl) = class frequency CFC in each clustering
        i, for each class cl
Begin
  1  Initialisation step:
     For each cluster i Do
        For each class cl Do
           CFC(i, cl) = 0
           Endfor
        Endfor
  2  For each cluster i Do
     For each example E_n Do
        If example E_n is in cluster i then
           CFC(i, class(E_n)) = CFC(i, class(E_n)) + 1
        Endif
     Endfor
     Endfor
  3  Calculation of frequency CFC:
     For each cluster i Do
        SUM = Sum all class cl in CFC(i, cl)
        For each class cl Do
           CFC(i, cl) = CFC (i, cl) / SUM
        Endfor
     Endfor
End
```

In the KNORA method, after the selection of the *K* neighbours, and consequently the classifiers that will compose the ensemble, we multiply the number of votes for the class *cl* by the relative frequency of the class *cl* inside the cluster to which the test sample belongs to. Thus, when selecting a class, instead of selecting the class that has more votes, it is chosen the class that has more votes weighted by the relative frequency of class in the cluster obtained by *U*-means that owns the test instance to be recognised. Such kind of weight may also be quantised as: {0.25; 0.50; 0.75; 1.0}. Thus, the class frequency could be defined considering these possible weights. For instance, values from 0 to 0.25 are defined as 0.25; values from 0.26 to 0.50 are defined as 0.50 and so on. Section 4.5 presents the results for the KNORA using this strategy.

3.4 The fusion of KNN and KNORA

As described in the previous section, all KNORA schemes take into account a built-in KNN. In this section, we investigate different strategies to combine the KNN (already used to select the classifiers for the KNORA method) with the final KNORA results, in order to approximate the recognition performance of that estimated as the oracle of our pool of weak classifiers. In fact, five different schemes were implemented:

- KNORA conditional use (CU): execute the KNN for the test pattern. If less than Y% (Y is a predetermined value) of the all neighbours of the current test pattern belong to the same class, then execute KNORA (eliminate or union), and use only the KNORA final outputs. Otherwise, use the KNN outputs. Figure 3 presents the KNORA (eliminate) CU scheme.
- KNN + KNORA conditional fusion (CF): execute the KNN for the test pattern. If less than Y% of the all neighbours of the current test pattern belongs to the same class, then combine the KNN outputs with the outputs of the KNORA (eliminate or union). Otherwise, use the KNN outputs. Figure 4 presents the KNN + KNORA (eliminate) CF scheme.
- KNN CU: execute KNORA (eliminate or union). If less than Y% of the classifiers in the ensemble selected for the current test pattern has the same vote, then use only the KNN outputs. Otherwise, use the KNORA outputs. Figure 5 presents the KNN (after KNORA-Eliminate) CU scheme.
- KNORA + KNN CF: execute KNORA. If less than Y% of the classifiers of the current test pattern has the same vote, then combine the KNORA (eliminate or union) outputs with the KNN outputs. Otherwise, use the KNORA outputs. Figure 6 presents the KNORA (eliminate) + KNN CF scheme.
- KNN + KNORA unconditional fusion (UF): combine the KNN outputs with the KNORA (eliminate or union) outputs. Figure 7 presents the KNN + KNORA (eliminate) UF scheme.

The fusion or combination of outputs is always done by the majority voting scheme (Huang and Suen, 1995; Kuncheva, 2004). The experimental results are reported in Section 4.6.

Figure 3 KNORA (eliminate) CU scheme

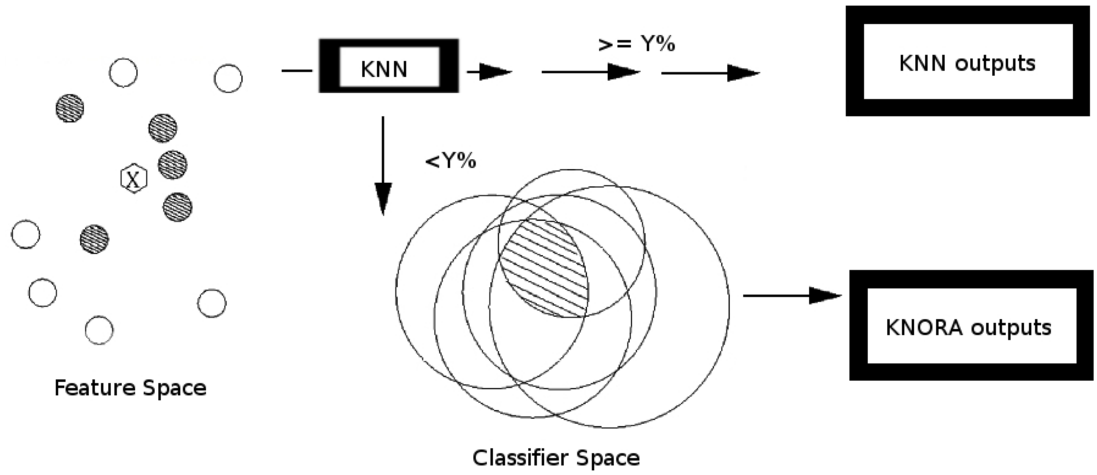


Figure 4 KNN + KNORA (eliminate) CF scheme

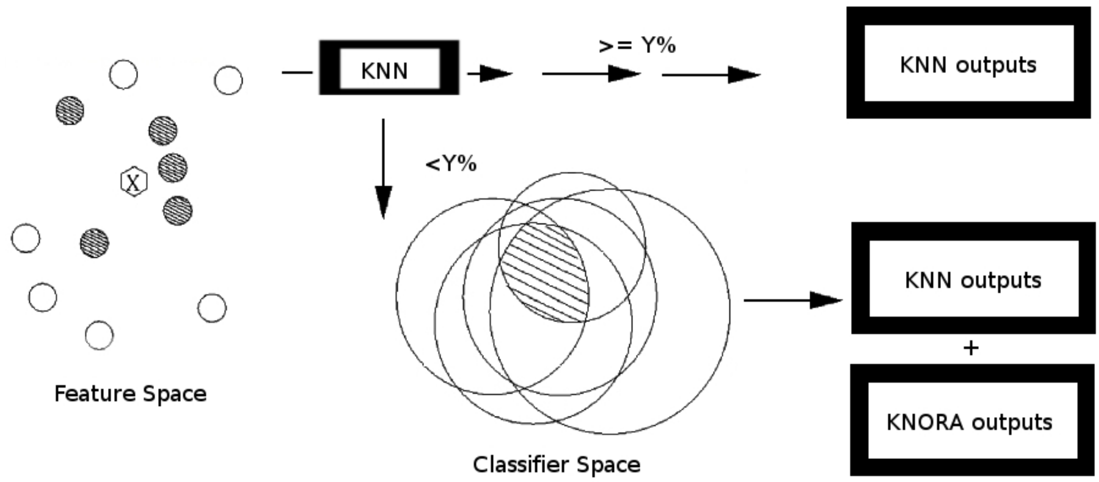


Figure 5 KNN (after KNORA-Eliminate) CU scheme

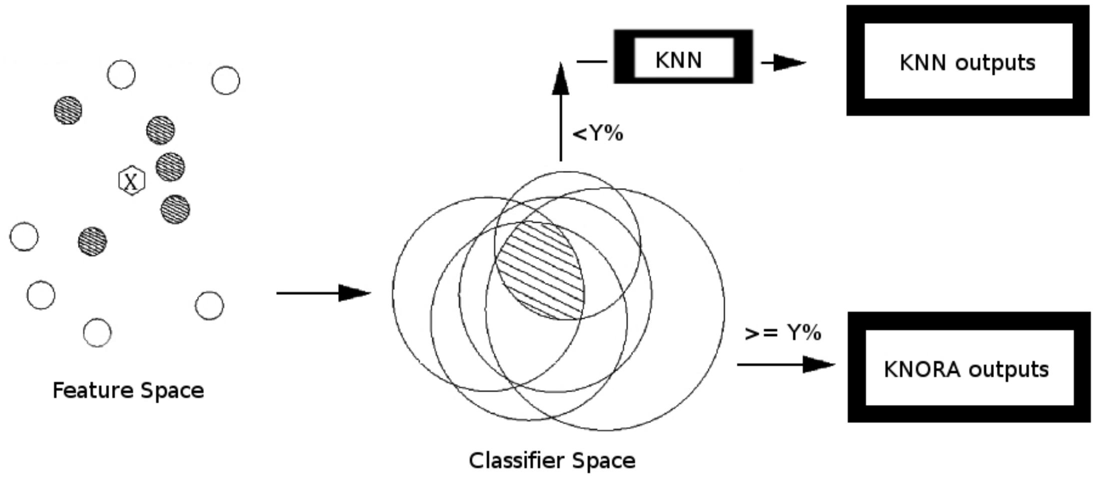


Figure 6 KNORA (eliminate) + KNN CF

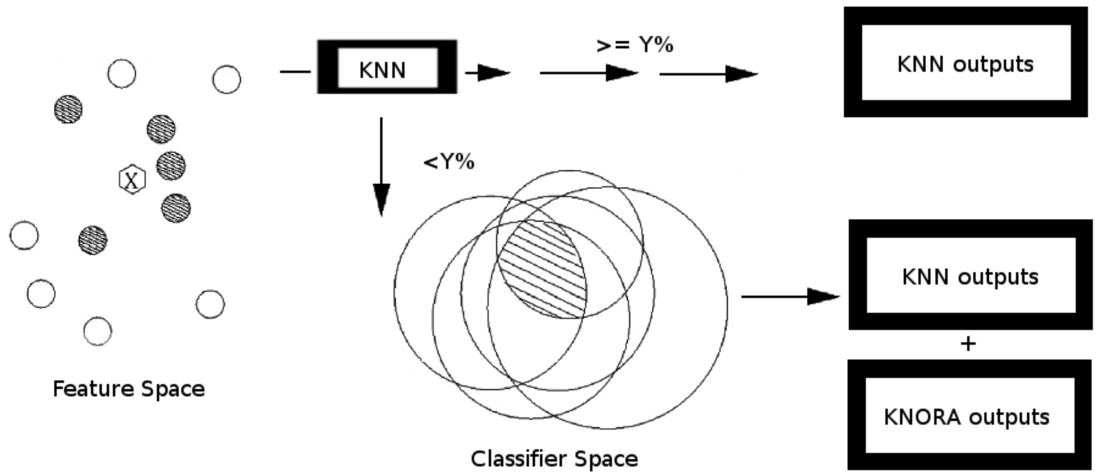
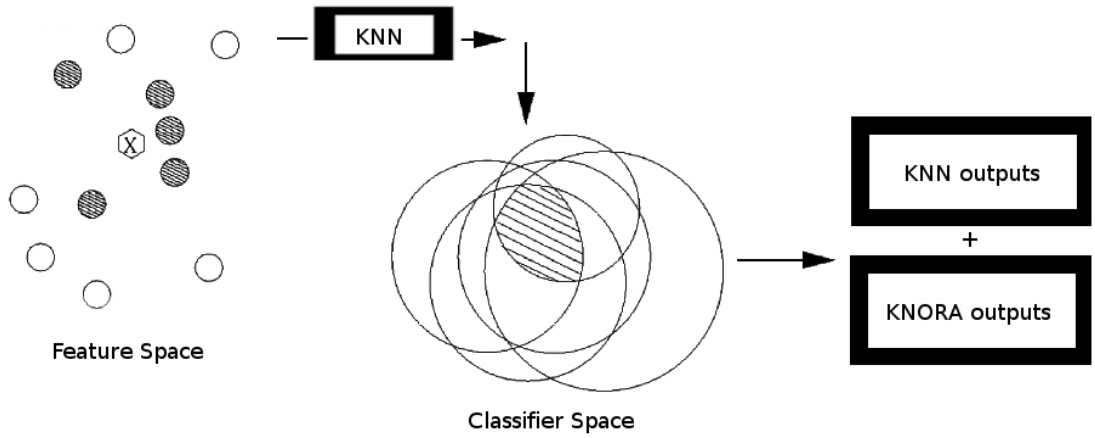


Figure 7 KNN + KNORA (eliminate) UF



4 Results and discussions

This section presents the experimental results and discussions of the strategies proposed in Section 3. First, in Section 4.1, it is shown the database and the pool of weak classifiers used in the experiments. Then, in Section 4.2, the benchmark parameters with the same database reported by Ko et al. (2008) are presented. These parameters are important to compare the achieved results by means of the proposed strategies.

The results of the evaluation using different distance metrics is presented in Section 4.3. The experiments using classifier accuracy and clustering are reported in Section 4.4 and 4.5, respectively. Finally, Section 4.6 shows the results obtained by means of the fusion strategies. All strategies were tested by considering KNORA-Eliminate and KNORA-Union.

4.1 Database and pool of weak classifiers

The experiments undertaken to evaluate the proposed strategies are based on the same experimental protocol described in Ko et al. (2008). It was selected a large scale pattern recognition problem related to the recognition of handwritten numerals from NIST SD19, with ten class (0 to 9). Three datasets were used: the training set with 5,000 samples (hsf_{0 – 3), the validation set containing 10,000 samples (hsf_{0 – 3) and the test set containing 60,089 samples (hsf_{7). The final accuracies were obtained evaluating the samples of the test set.

We need to address the fact that the pool of the KNORA method is composed of 1-NN (*KNN where* $K = 1$) classifiers generated with feature subsets having only 32 features out of 132 by using a random subset selection scheme. The same pool of weak classifiers proposed in Ko et al. (2008) is used in our experiments. This pool contains 100 1-NN classifiers. It is important clarify that the variable K in this paper is related to the KNORA method and all experiments were executed considering values from 1 to 30.

4.2 Benchmark parameters

In Ko et al. (2008), the authors have reported that with the 132-feature-based 1-NN (*KNN where* $K = 1$), the performance on the testing set is 93.34%. The combination of all 32-feature-based 1-NN classifiers available in our pool (100 elements) by simple majority voting gives 96.28% of classification accuracy. In addition, the best KNORA recognition rates for the same database were reported as: 97.25% for KNORA-Union ($K = 1$) and 97.52% for KNORA-Eliminate ($K = 7$ and $K = 8$), as shown in Table 1. The K parameter of the KNORA method had been evaluated from 1 to 30.

Table 1 Best recognition rates (in %) and the corresponding K values, plus the oracle performance

Scheme	*Recognition rates(K)*
KNN	93.34(1)
KNORA-Eliminate	97.52(7,8)
KNORA-Union	97.25(1)
Oracle performance	99.95

As one can see in Table 1, the oracle of our pool of weak classifiers is 99.95% of recognition rate. This was calculated by looking down at the pool of classifiers if there is some classifier that well-recognise each test sample.

4.3 Evaluation using different distance metrics

The experiments in this section are related to the use of Camberra, Cosine and Pearson distances as alternative to the Euclidian distance used in the original KNORA method. The first line of Table 2 presents the best results (and the corresponding K values) in terms of recognition rates obtained by using the Camberra distance to select the neighbours. In the KNORA-Eliminate scheme, the best recognition rate observed was 97.13%, with $K = 6$ to $K = 30$ and in the KNORA-Union was 96.99%, with $K = 8$ to $K = 10$. These results were worst than the best result obtained by means of the KNORA-Eliminate with Euclidian distance (97.52%). The second line in Table 2 presents the best results by using the Cosine distance. The best result for the KNORA-Eliminate was 97.49%, which is slightly worst than the recognition rate obtained using the Euclidean distance. In KNORA-Union, the recognition rate was 97.25%, with $K = 1$ (the same obtained using Euclidean distance).

Table 2 Best recognition rates (in %) for different distance metrics and the corresponding K values

Distance metric	*KNORA-Eliminate(K)*	*KNORA-Union(K)*
Camberra	97.13(6–30)	96.99(8–10)
Cosine	97.49(9, 10, 12–15)	97.25(1)
Pearson	97.44(4, 5, 7)	97.25(1)

The best results using Pearson distance for the KNORA-Eliminate was 97.44%, with $K = 4$, $K = 5$ and $K = 7$ (worst than the recognition rate obtained using Euclidean distance) and for KNORA-Union was 97.25%, with $K = 1$ (the same obtained using Euclidean distance).

The Camberra, Cosine and Pearson metrics did not provide any improvement in terms of recognition rates when compared with the results obtained using the Euclidian distance to select the test neighbourhood in the KNORA method. In fact, we may conclude that small changes in the selected neighbourhood do not provide significant changes in the results. A deeply analysis of the selected neighbourhoods shows that most of the time they are very similar.

4.4 Experiments using the classifier accuracy

Here, the classifiers that compose the initial pool are ranked by class recognition accuracy. First, we calculate the classifier class accuracy in the validation set. Thus, when the classifier is selected to give a vote for the predicted class, the vote is weighted by the classifier class accuracy already estimated. This was evaluated in the KNORA-Eliminate and KNORA-Union methods.

The worst and best accuracies per class of the ranking of classifiers are presented in Table 3. The worst observed result is 62.8% of recognition rate by the classifier number 73 for the digit class 8. The best result in the rank is 96.1% of accuracy for the class 0 provided by the classifier number 12.

The best results in terms of recognition rates when using the KNORA-Eliminate and KNORA-Union schemes were 97.48% ($K = 8$) and 97.17% ($K = 1$), respectively. As one can see in Table 1, these results do not contribute to improve the performance of the KNORA method. In fact, the experiments have shown that the information obtained from the created rank of classifiers is not appropriate to define the importance of a specific classifier inside of the selected ensemble. The reason is that the classifiers in the pool are weak, and then, it is not possible to observe a clear contribution of each classifier for specific classes and also a significant difference in terms of class accuracy among them.

Table 3 The worst and best recognition accuracies (in %) per class and the corresponding classifier number

Digit class	*Worst result*	*Best result*
0	85.0(29)	96.1(12)
1	69.5(73)	94.0(91)
2	73.5(96)	93.4(45)
3	80.8(85)	92.5(87)
4	79.7(52)	94.8(81)
5	74.4(70)	92,8(8)
6	83.2(24)	97.1(49)
7	77.7(95)	94.9(40)
8	62.8(73)	88.0(91)
9	71.9(73)	91.8(7)

4.5 Experiments using clustering

The clustering process of the validation set was done by using the *U*-means method, implemented in the Weka tool (Hall et al., 2009). The *U* value, i.e., the number of clusters was evaluated from 2 to 30. Table 4 shows the class distribution at each cluster when using $U = 10$. The rows indicate the class of the samples in the validation set, and the columns show the cluster (from 0 to 9). For instance, in the first column (cluster 0) of Table 4, the predominant class is 2 with 460 samples; while in the second column (cluster 1), the predominant class is 6 with 937 samples. Note that in clusters 7 and 8, despite the existence of a dominant class, the difference is not significant for the second option. Cluster 8 has only seven samples of the predominant class. Probably this cluster contains more difficult samples for classification.

The cluster of the test sample is defined by using the Euclidian distance. The class frequency is used as weight for the votes of the KNORA-Eliminate and KNORA-Union schemes. Table 5 presents the best recognition rates (and the respective *K* value) for some values of the *U* parameter. For KNORA-Eliminate, the best recognition rate was 96.46%, obtained with $U = 2$ and with $K = 10$, while for the KNORA-Union, the best recognition rate was 95.56%, obtained with $U = 2$ and with $K = 1$.

In Table 5, it is possible to see that for KNORA-Union the best recognition rate was always obtained with $K = 1$. The KNORA-Union considers one vote per classifier for each well-recognised neighbour. Thus, when the *K* value increases, some classifiers may provide more votes (one for each recognised neighbour), and it can, probably, be the reason of the worst recognition results. With $K = 1$, the neighbour selected is more similar to the test sample and the classifier can give more accurate votes.

Table 4 The absolute class frequencies considering ten clusters

Class	*Cluster*									
	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
0	3	30	0	7	922	8	1	0	0	29
1	24	25	4	40	0	29	6	55	3	814
2	460	43	2	0	0	471	7	4	2	11
3	371	2	0	15	4	535	21	34	4	14
4	0	16	861	47	0	4	13	21	4	34
5	151	16	2	742	0	74	1	1	7	6
6	1	937	5	24	1	2	0	0	5	25
7	3	0	0	8	0	2	624	342	3	18
8	5	29	17	403	244	7	112	63	2	118
9	3	0	56	20	6	1	532	377	0	5

Table 5 The best recognition rates (%) by KNORA-Eliminate and KNORA-Union for some *U* values and the corresponding *K* values

U value	*KNORA-Eliminate(K)*	*KNORA-Union(K)*
2	96.46(10)	95.56(1)
3	95.30(21)	93.92(1)
4	95.64(26, 27)	94.14(1)
5	94.62(30)	92.42(1)
6	94.17(29, 30)	91.59(1)
7	94.17(29, 30)	90.04(1)
8	92.90(30)	89.60(1)
9	92.89(30)	89.62(1)
10	92.03(29, 30)	88.16(1)
15	90.25(30)	85.66(1)
20	89.18(30)	84.22(1)
30	88.05(30)	83.23(1)

The best recognition rates was obtained with $U = 2$. The reason for this can be that some classes have their weights with low values in the cluster. Then, when $U = 2$ we had an interesting frequency distribution between classes in the clusters.

The observed 96.46% of recognition rate of the KNORA-Eliminate ($U = 2$, $K = 10$) and 95.56% of the KNORA-Union ($U = 2$, $K = 1$) were both better results than 93.34% obtained by the 132-feature-based 1-NN. In addition, the recognition rate of 96.46% of the KNORA-Eliminate ($U = 2$, $K = 10$) was better than 96.28% observed when all classifiers of the pool were combined (100 elements) by simple majority voting. However, KNORA-Eliminate using additional information about the class frequency in the clustering provided worst results than the original KNORA-Eliminate. The same was observed for the KNORA-Union.

Another way of using the clustering information was evaluated as described in Section 3.3. It was implemented only for U = 10. It does not use directly the frequency distribution of classes in the cluster as weight, but one of the four possible values: 0.25, 0.5, 0.75 or 1. The weight 0.25 would be assigned to all classes that have relative frequency of 0.25 or less than that in the cluster. Similarly, the weight 0.5 would be assigned to all classes that have relative frequency from 0.26 to 0.5 in the cluster, and so on. Table 6 presents the best performances obtained considering U = 10. Special attention was given to U = 10, since the digit recognition is a 10-class problem. The best recognition rate for the KNORA-Eliminate in this scheme was 96.51% (with K = 10) and it outperforms the KNORA-Eliminate using normal weight (second line in Table 6) and every test showed in Table 5. For the KNORA-Union, the recognition rate was 95.56% (K = 1), the same observed as the best result to KNORA-Union in Table 5. The best results were obtained when the class frequency was quantised. Thus, we may conclude that some classes had very low frequency values and as weights they become insignificant.

Table 6 Best recognition rates by using the class frequency in the clusters for U = 10 and the corresponding K values

Scheme	*KNORA-Eliminate(K)*	*KNORA-Union(K)*
Normal weight	92.03(29, 30)	88.16(1)
Four values of weight	96.51(10)	95.56(1)

As one can see, by using statistics about the class frequency at each cluster as additional information in the KNORA method, we did not observe any improvement in the results of the original KNORA method. In fact, even when inside of the cluster where the test pattern belongs to, there is a significant predominance of certain class. This is not evidence that the test pattern has high probability of being of the same class.

4.6 Evaluation of the fusion strategies

The Y parameter for the combination schemes proposed in this paper was evaluated from 10% to 100%. Figures 8 to 16 present the recognition rates obtained by considering different K and Y values, for each fusion strategy compared to the original KNORA-Eliminate scheme. Table 7 summarises these experimental results. The following considerations may be done for each strategy:

- KNORA CU: Figure 8 shows the recognition rates using the KNORA-Eliminate CU strategy. It was observed that KNORA-Eliminate CU strategy with high values of Y provided better results than the original KNORA-Eliminate. Similar results were observed for the KNORA-Union CU strategy (Figure 9). The best recognition rates are 97.66% (with Y > = 80%) when using KNORA-Eliminate, and 97.54% (with Y > = 70%) when using KNORA-Union (see Table 7). As we can observe, both cases provided better results than the original KNORA (eliminate 97.52% and union 97.25%) or even than the KNN (see Table 1).
- KNN + KNORA CF: in this scheme we have also observed better results than using the original KNORA-Eliminate and KNORA-Union. Figure 10 shows the recognition rates using the KNN + KNORA-Eliminate CF strategy. It was observed that KNN + KNORA-Eliminate CF strategy with high values of Y obtained better results than KNORA-Eliminate. In KNN + KNORA-Union CF strategy (Figure 11), we observed better recognition rates than that obtained by the original KNORA-Eliminate, but only for few values of K. The best recognition rates are 97.73% (with Y> = 80%) when using KNORA-Eliminate, and 97.54% (with Y> = 70%) when using KNORA-Union (see Table 7).
- KNN CU: Figure 12 shows the recognition rates by using the KNN CU after KNORA-Eliminate. It was observed that KNN CU after KNORA-Eliminate with Y = 50% and Y = 60% provided better results than the original KNORA-Eliminate. However, for KNN CU after KNORA-Union (Figure 13), we observed worst recognition rates than that of the original KNORA-Eliminate. The best recognition rate is 97.58% (with Y = 50%) when using KNN (after KNORA-Eliminate) CU, which is better than KNORA-Eliminate alone (97.52%). The best recognition rate is 97.48% (with Y = 60%) when using KNN (after KNORA-Union) CU, which is better than KNORA-Union alone (97.25%).
- KNORA + KNN CF: Figure 14 shows the recognition rates using the KNORA (eliminate) + KNN CF. It was observed that KNORA (eliminate) + KNN (CF) in the most of the cases obtained equal or better results than KNORA-Eliminate. In KNORA (union) + KNN (CF) (Figure 15), we observed worst recognition rates than that of the original KNORA-Eliminate. The best recognition rates are 97.70% (with Y> = 70%) when using KNORA-Eliminate, and 97.30% (with Y> = 60%) when using KNORA-Union.
- KNN + KNORA UF: in this scheme, we always combine the KNN with KNORA (eliminate or union) for all samples to be classified. Thus, there is no Y parameter to evaluate. The best recognition rate (Table 8) observed for KNN + KNORA (eliminate) UF was 97.74% (K = 6), while for the KNN + KNORA (union) UF was 97.30% (K = 1). Figure 16 presents the recognition rate obtained for each value of the parameter K (of 1 up to 30) in this scheme, as well as the recognition rates obtained for KNORA-Eliminate and KNORA-Union in the work of Ko et al. (2008). It is observed that KNN + KNORA-Union UF reached better results than KNORA-Union of Ko et al. (2008), and that KNN + KNORA-Eliminate UF reached better results than KNORA-Eliminate of Ko et al. (2008).

Figure 8 Recognition rates using the KNORA-Eliminate CU strategy (see online version for colours)

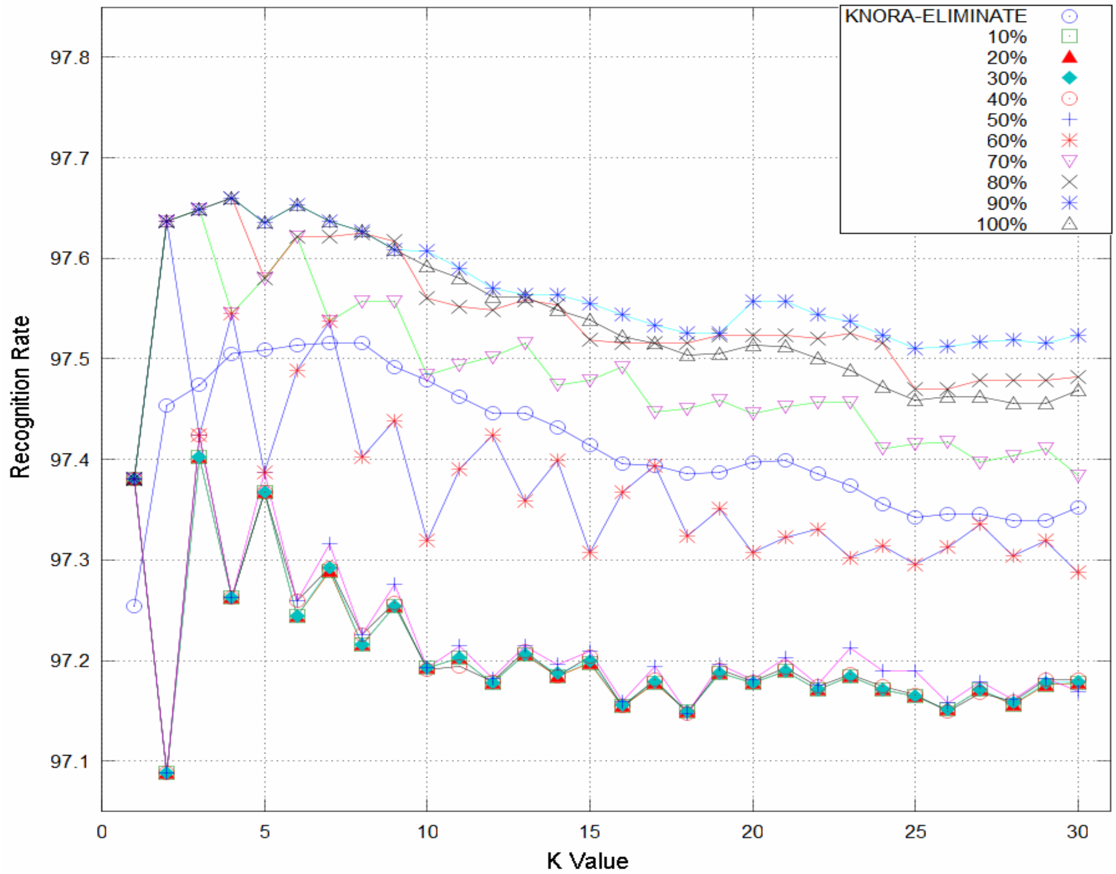


Figure 9 Recognition rates using the KNORA-Union CU strategy (see online version for colours)

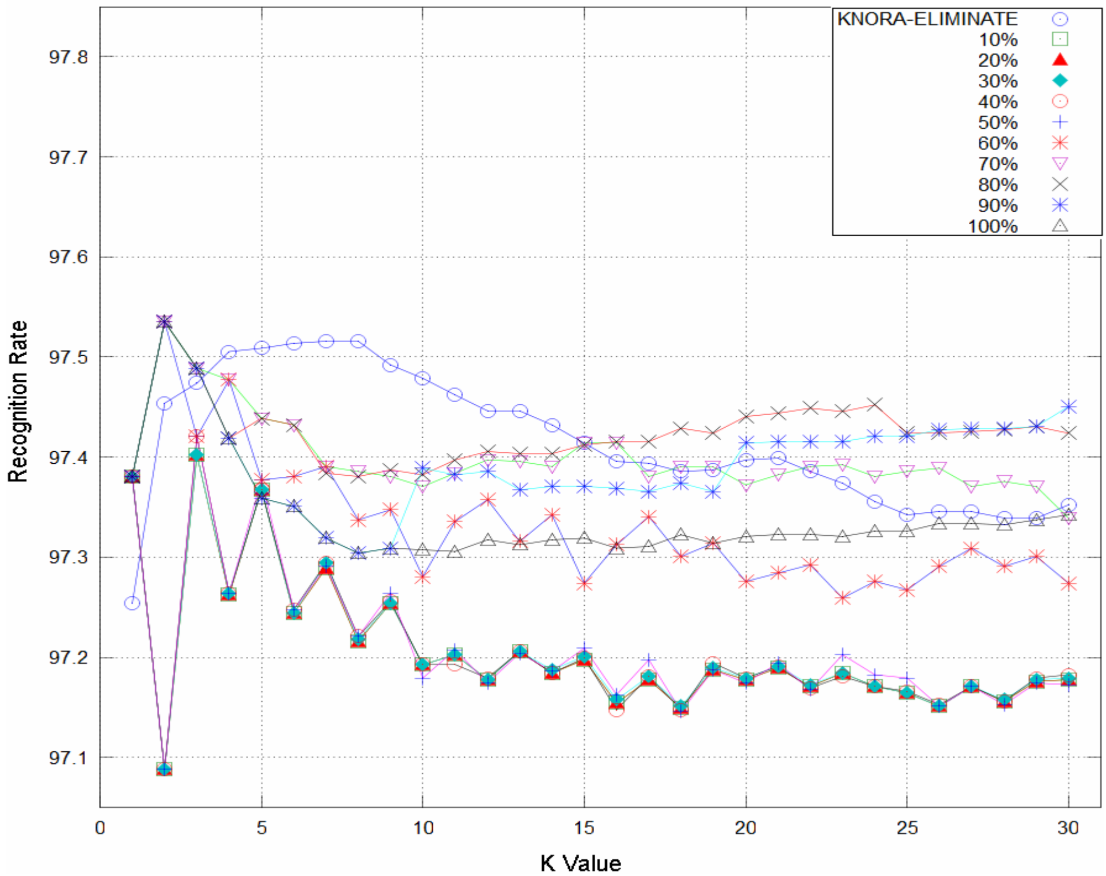


Figure 10 Recognition rates using the KNN + KNORA-Eliminate CF strategy (see online version for colours)

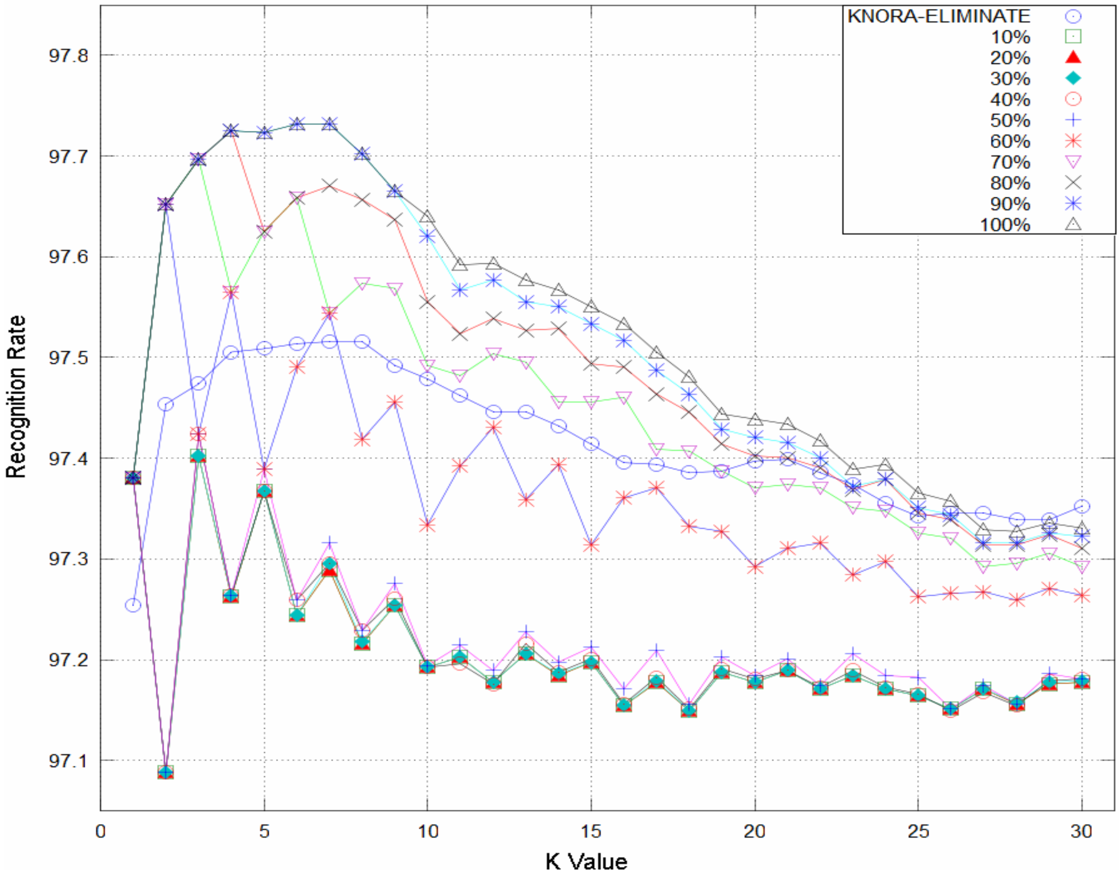


Figure 11 Recognition rates using the KNN + KNORA-Union CF strategy (see online version for colours)

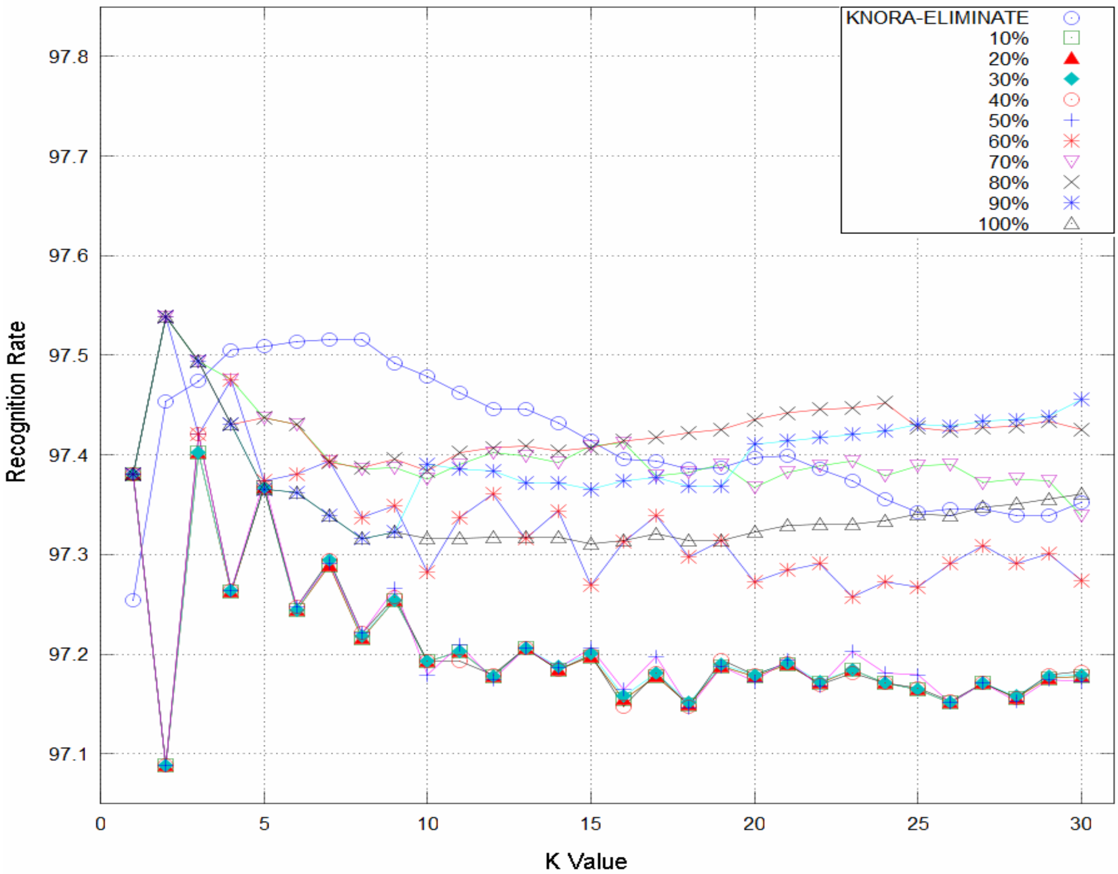


Figure 12 Recognition rates using the KNN CU after KNORA-Eliminate (see online version for colours)

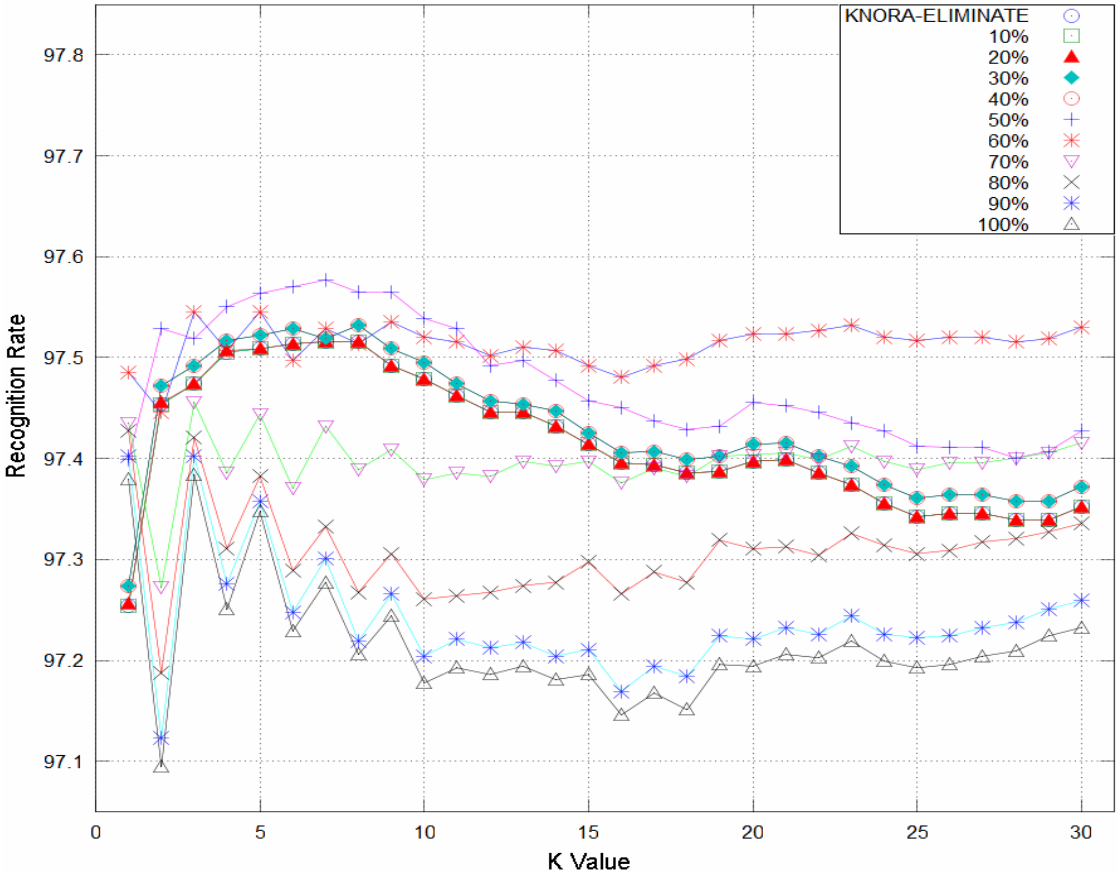


Figure 13 Recognition rates using the KNN CU after KNORA-Union (see online version for colours)

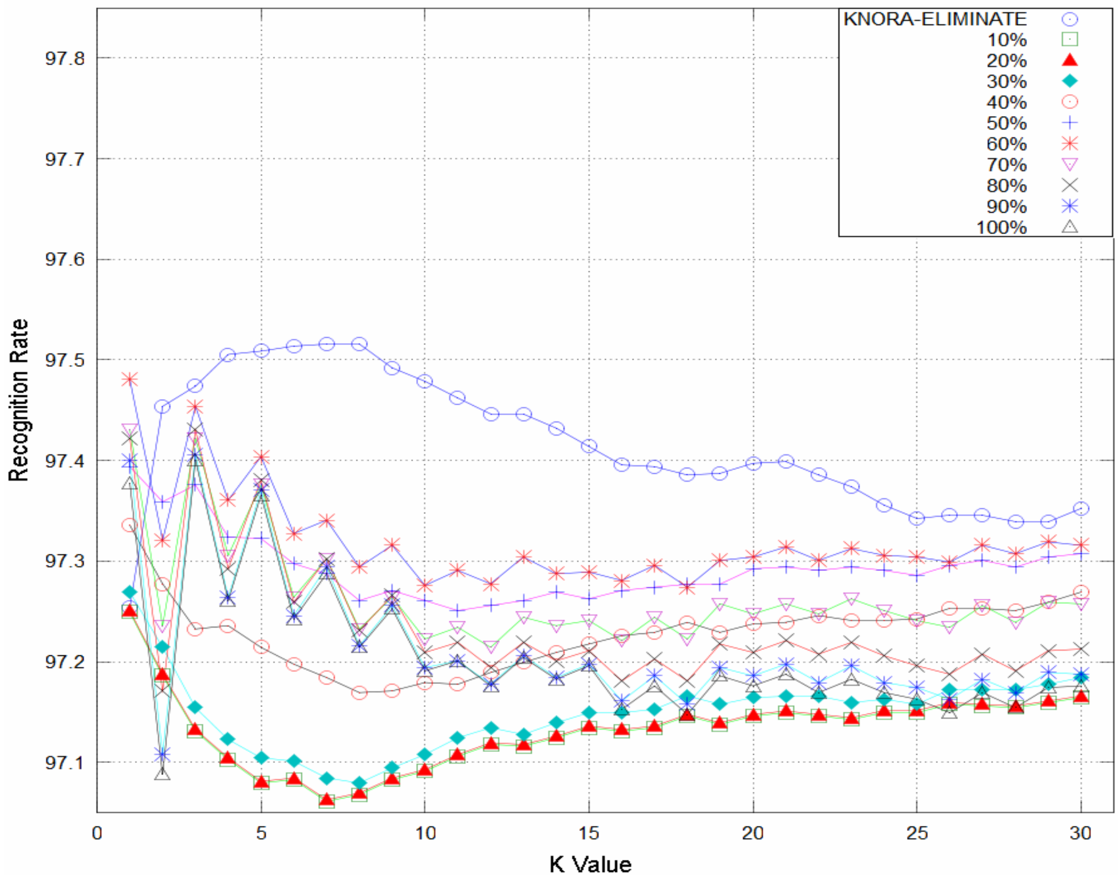


Figure 14 Recognition rates using the KNORA (eliminate) + KNN CF (see online version for colours)

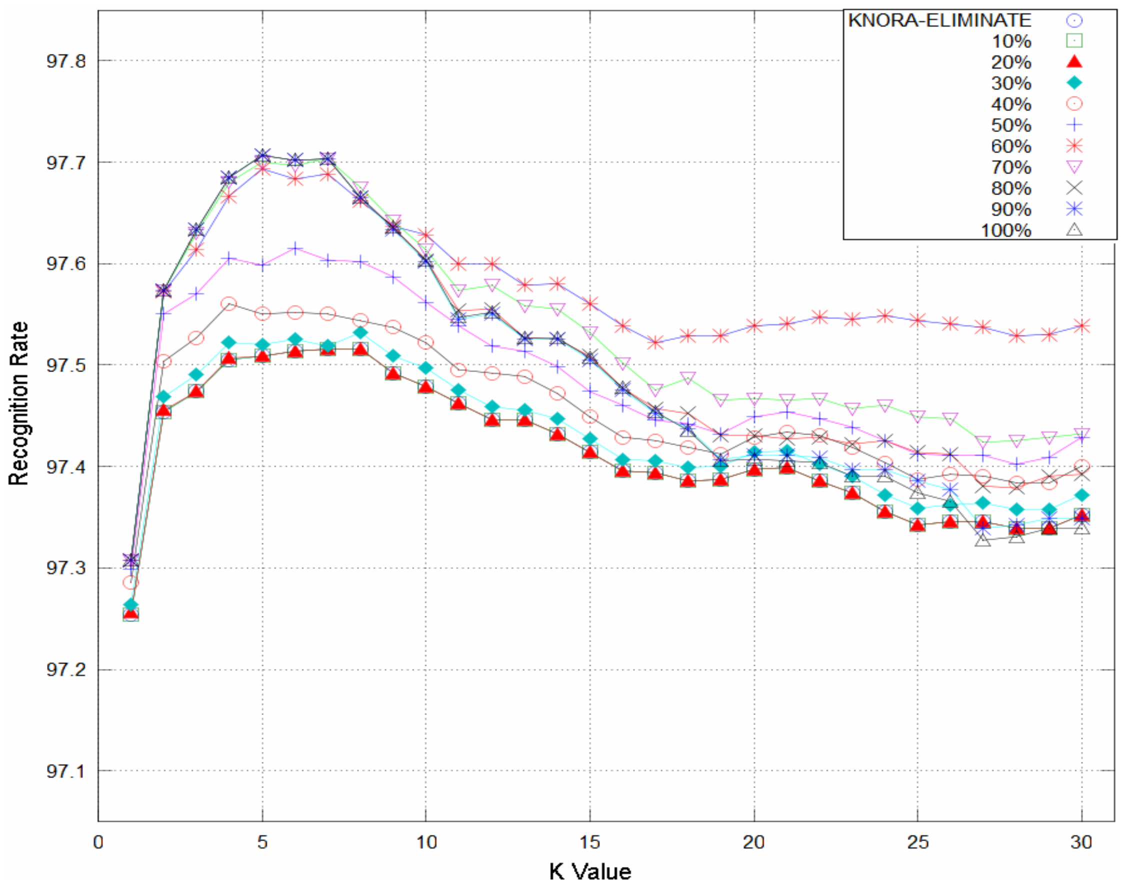


Figure 15 Recognition rates using the KNORA (union) + KNN CF (see online version for colours)

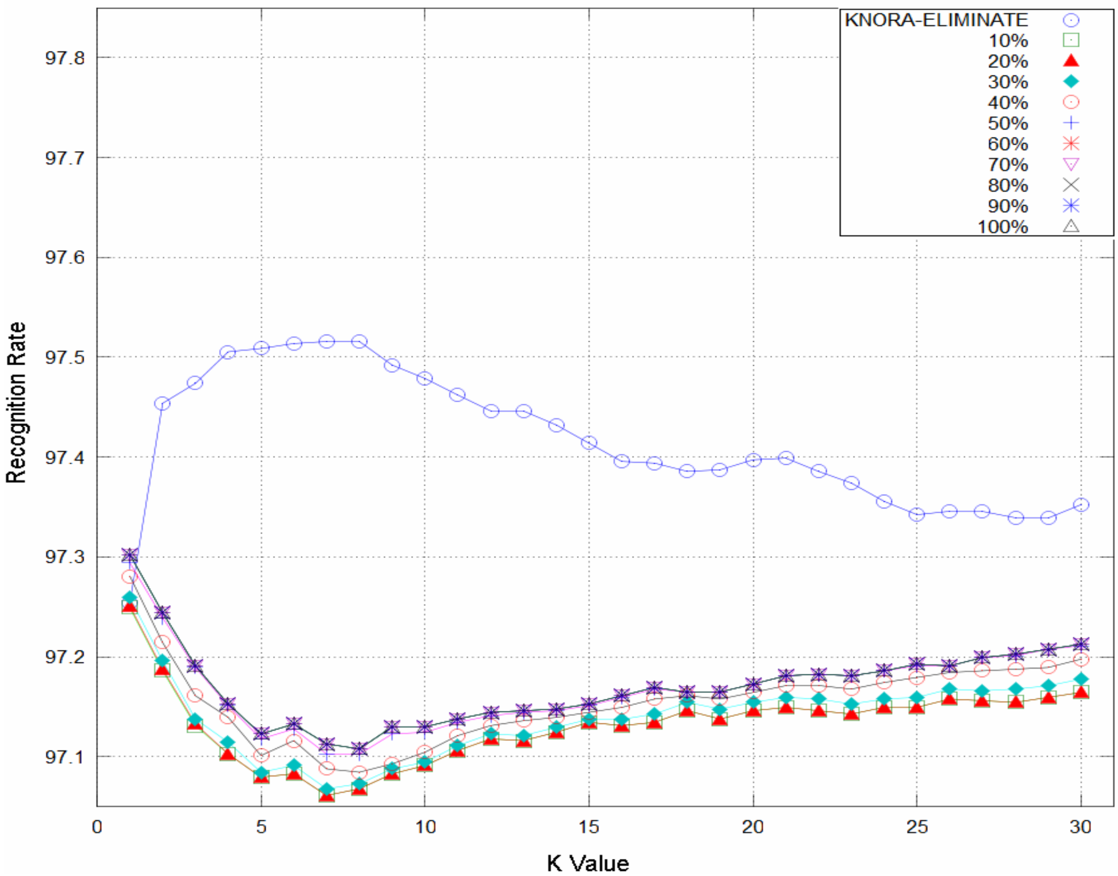


Figure 16 Recognition rates for KNORA-Union, KNORA-Eliminate, KNN + KNORA (U) UF and KNN + KNORA (E) UF (see online version for colours)

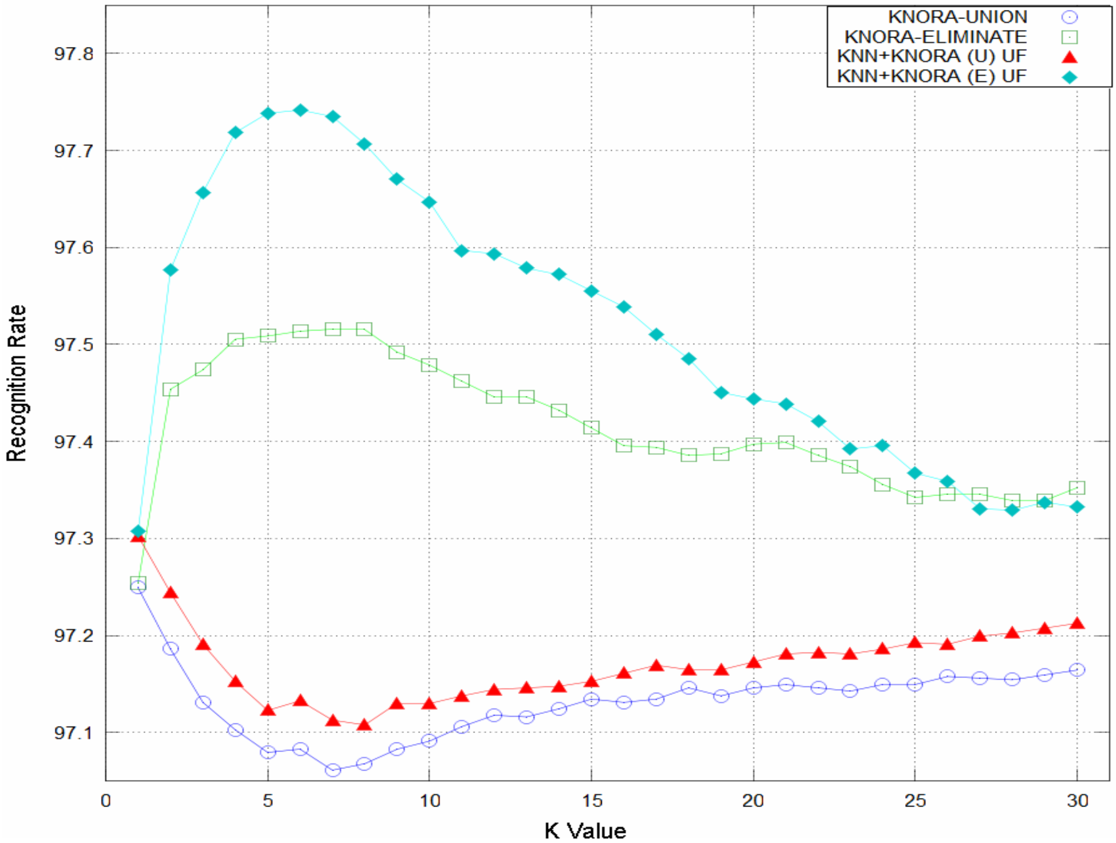


Table 7 Recognition rates by using the fusion of KNN and KNORA considering different *Y* and (*K*) values

Fusion scheme	*Y%*									
	10%	*20%*	*30%*	*40%*	*50%*	*60%*	*70%*	*80%*	*90%*	*100%*
KNORA(E) CU	97.40(3)	97.40(3)	97.40(3)	97.42(3)	97.40(3)	97.64(2)	97.64(3)	97.66(4)	97.66(4)	97.66(4)
KNORA(U) CU	97.40(3)	97.40(3)	97.40(3)	97.42(3)	97.42(3)	97.48(4)	97.54(2)	97.54(2)	97.54(2)	97.54(2)
KNN + KNORA(E) CF	97.40(3)	97.40(3)	97.40(3)	97.42(3)	97.42(3)	97.65(2)	97.70(3)	97.73(4)	97.73(6,7)	97.73(6,7)
KNN + KNORA(U) CF	97.40(3)	97.40(3)	97.40(3)	97.42(3)	97.42(3)	97.48(4)	97.54(2)	97.54(2)	97.54(2)	97.54(2)
KNN (After KE) CU	97.52(7, 8)	97.52(7, 8)	97.53(8)	97.53(8)	97.58(7)	97.55(3, 5)	97.46(3)	97.43(1)	97.40(1, 3)	97.38(3)
KNN(After KU) CU	97.25(1)	97.25(1)	97.27(1)	97.34(1)	97.39(1)	97.48(1)	97.43(1)	97.43(3)	97.41(3)	97.40(3)
KNORA(E) + KNN CF	97.52(7, 8)	97.52(7, 8)	97.53(8)	97.56(4)	97.62(6)	97.69(5)	97.70(7)	97.70(7)	97.70(7)	97.70(7)
KNORA(U) + KNN CF	97.25(1)	97.25(1)	97.26(1)	97.28(1)	97.29(1)	97.30(1)	97.30(1)	97.30(1)	97.30(1)	97.30(1)

Table 8 Best recognition rates and the corresponding *K* values by using the KNN + KNORA UF

Scheme	*Recognition rates (K)*
KNN + KNORA (E) UF	97.74(6)
KNN + KNORA (U) UF	97.30(1)

As one can see, the best result of all experiments was 97.74% (K = 6), achieved by KNN + KNORA (eliminate) UF. We observed better results than 93.34% (132-feature-based 1-NN), than 96.28% (combination of all 32-feature-based 1-NN classifiers available in our pool), than 97.25% (KNORA-Union with K = 1) and than 97.52% (KNORA-Eliminate with K = 7 and K = 8). However, even observing some improvement in the KNORA results, the recognition rates still far from the oracle performance (99.95%).

5 Conclusions and future work

This work evaluated different strategies to improve a dynamic ensemble selection method trying to approximate its performance to the oracle accuracy. Table 9 summarises the best results obtained by each proposed strategy and the respective K values.

Table 9 Best recognition rates and the corresponding strategies

Strategy	*Using KNORA-Eliminate(K)*	*Using KNORA-Union(K)*
Different distance metrics	Pearson	Cosine and Pearson
	97.44(4, 5, 7)	97.25(1)
The use of classifier accuracy	97.48(8)	97.17% (1)
The use of clustering	10-means with quantised weights	2-means
	96.51(10)	95.56(1)
Fusion of KNN and KNORA	KNN + KNORA (E) UF	KNORA(U) CU and KNN + KNORA(U) CF
	97.74(6)	97.54(2)

The evaluation of different metrics in the KNORA method did not provide any improvement in terms of recognition rate when compared with the results obtained using the original Euclidian distance. From the experiments, we may conclude that the change of the distance metric has a small impact in the neighbourhood selection, and, consequently, it does not provide any significant change in the results.

Since the classifiers in the KNORA method are selected to compose the ensemble for a test pattern based on their accuracy in recognising the test neighbourhood, two strategies to improve the KNORA method using information about class accuracy were evaluated:

a statistics related to the classifier accuracy (ranking of classifiers) for each class

b statistics related to a clustering scheme, where each test sample were addressed to a cluster for which we knew the frequency of each class.

The recognition rates obtained when using the ranking of classifiers in the pool by their class accuracy were worst than the recognition rates obtained without ranking. It seems that the information obtained from the created rank of classifiers is not appropriate to define the importance of a specific classifier inside of the selected ensemble. The reason is that the classifiers in the pool are weak, and then, it is not possible to observe a clear contribution of each classifier for specific classes and also a significant difference in terms of class accuracy among them. Similar behaviour was observed during the experiments using the clustering strategy. However, here, we observed that even when inside of the cluster where the test pattern belongs to, there is the predominance of certain class. This is not evidence that the test pattern has high probability of being of the same class.

Finally, from the last experiments, we can see some recognition improvements. The additional information provided by the KNN built in the KNORA method, originally used to define the neighbourhood of the test pattern, allow us to improve the recognition performance in most of the strategies evaluated. The best results were achieved when an UF were used. It means that the neighbourhood additional information plays an important role in the classification process provided by the KNORA method.

As future works, we plan to model the oracle properties, aiming to replace the KNN used in the KNORA process by a classifier, which the objective will be to select the classifiers that will be part of the ensemble for a specific test pattern.

References

Brown, G., Wyatt, J., Harris, R. and Yao, X. (2005) 'Diversity creation methods: a survey and categorisation', *International Journal of Information Fusion*, Vol. 6, No. 1, pp.5–20.

Cao, J., Ahmadi, M. and Shridhar, M. (1995) 'Recognition of handwritten numerals with multiple feature and multistage classifier', *Pattern Recognition*, Vol. 28, No. 2, pp.153–160.

Didaci, L. and Giacinto, G. (2004) 'Dynamic classifier selection by adaptative k-nearest-neighbourhood rule', *International Workshop on Multiple Classifier Systems (MCS 2004)*, pp.174–183.

Didaci, L., Giacinto, G., Roli, F. and Marcialis, G.L. (2005) 'A study on the performances of dynamic classifier selection based on local accuracy estimation', *Pattern Recognition*, Vol. 38, No. 11, pp.2188–2191.

Giacinto, G. and Roli, F. (1999) 'Methods for dynamic classifier selection', *International Conference on Image Analysis and Processing (ICIAP 1999)*, pp.659–664.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I.H. (2009) 'The WEKA data mining software: an update', *SIGKDD Explorations*, Vol. 11, No. 1, pp.10–18.

Hartigan, J.A. (1975) *Clustering Algorithms*, John Wiley & Sons, New York.

Huang, Y.S. and Suen, C.Y. (1995) 'A method of combining multiple experts for the recognition of unconstrained handwritten numerals', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 1, pp.90–93.

Jain, A.K., Murthy, M.N. and Flynn, P.J. (1999) 'Data clustering: a review', *ACM Computing Reviews*, Vol. 31, No. 3, pp.265–323.

Kittler, J., Hatef, M., Duin, R. and Matas, J. (1998) 'On combining classifiers', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 3, pp.226–239.

Ko, A.H.R. (2007) 'Static and dynamic selection of ensemble of classifiers', PhD thesis, École de Technologie Supérieure, Université Du Québec, Canada, 246p.

Ko, A.H.R., Sabourin, R. and Britto, A.S., Jr. (2008) 'From dynamic classifier selection to dynamic ensemble selection', *Pattern Recognition*, Vol. 41, No. 5, pp.1718–1731.

Kuncheva, L.I. (2004) *Combining Pattern Classifiers. Methods and Algorithms*, Wiley-Interscience, Hoboken, NJ.

Kuncheva, L.I. and Rodrígues, J.J. (2007) 'Classifier ensembles with a random linear oracle', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 19, No. 4, pp.500–508.

Kuncheva, L.I. and Whitaker, C.J. (2003) 'Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy', *Machine Learning*, Vol. 51, No. 2, pp.181–207.

MacQueen, J.B. (1967) 'Some methods for classification and analysis of multivariate observations', in Cam, L.M.L. and Neyman, J. (Eds.): *Proc. of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 1, pp.281–297.

Nicoletti, M.C. and Bertini, J.R., Jr. (2007) 'An empirical evaluation of constructive neural network algorithms in classification tasks', *International Journal of Innovative Computing and Applications,* Vol. 1, No. 1, pp.2–13.

Opitz, D. and Maclin, R. (1999) 'Popular ensemble methods: an empirical study', *Journal of Artificial Intelligence Research*, Vol. 11, pp.169–198.

Pekalska, E., Skurichina, M. and Duin, R.P.W. (2004) 'Combining dissimilarity-based one-class classifiers', *International Workshop on Multiple Classifier Systems (MCS 2004)*, pp.122–133.

Ranawana, R. (2006) 'Multi-classifier system – review and a roadmap for developers', *International Journal of Hybrid Intelligent Systems*, Vol. 3, No. 2, pp.35–61.

Santana, A., Soares, R.G.F., Canuto, A.M.P. and Souto, M.C.P. (2006) 'A dynamic classifier selection method to build ensembles using accuracy and diversity', in *Proceedings of the Ninth Brazilian Symposium on Neural Networks (SBRN'06)*, pp.36–41.

Santos, E.M.D., Sabourin, R. and Maupin, P. (2008) 'A dynamic overproduce-and-choose strategy for the selection of classifier ensembles', *Pattern Recognition*, Vol. 41, No. 10, pp.2993–3009.

Silva, L.A., Del-Moral-Hernandez, E., Moreno, R.A. and Furuie, S.S. (2009) 'Cluster-based classification using self-organising maps for medical image databases', *International Journal of Innovative Computing and Applications*, Vol. 2, No.1 pp.13–22.

Webb, G.I. and Zheng, Z. (2004) 'Multistrategy ensemble learning: reducing error by combining ensemble learning techniques', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 16, No. 8, pp.980–991.

Woods, K., Kegelmeyer, W.P., Jr. and Bowyer, K. (1997) 'Combination of multiple classifiers using local accuracy estimates', *IEEE Transactions of Pattern Analysis and Machine Intelligence*, Vol. 19, No. 4, pp.405–410.

Xiao, J. and He, C. (2009) 'Dynamic classifier ensemble selection based on GMDH', *2009 International Joint Conference on Computational Sciences and Optimization*, Vol. 1, pp.731–734.

Zouari, H., Heutte, L., Lecourtier, Y. and Alimi, A. (2004) 'Building diverse classifier outputs to evaluate the behavior of combination methods: the case of two classifiers', *International Workshop on Multiple Classifier Systems (MCS 2004)*, pp.273–282.