Available online at www.sciencedirect.com
ScienceDirect

Pattern Recognition 41 (2008) 1718–1731

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/pr

From dynamic classifier selection to dynamic ensemble selection

Albert H.R. Ko[a,*], Robert Sabourin[a], Alceu Souza Britto, Jr.[b]

[a] *LIVIA, École de Technologie Supérieure, University of Quebec, 1100 Notre-Dame West Street, Montreal, Que., Canada H3C 1K3*
[b] *PPGIA, Pontifical Catholic University of Parana, Rua Imaculada Conceicao, 1155, PR 80215-901, Curitiba, Brazil*

Received 5 March 2007; received in revised form 22 August 2007; accepted 9 October 2007

Abstract

In handwritten pattern recognition, the multiple classifier system has been shown to be useful for improving recognition rates. One of the most important tasks in optimizing a multiple classifier system is to select a group of adequate classifiers, known as an Ensemble of Classifiers (EoC), from a pool of classifiers. Static selection schemes select an EoC for all test patterns, and dynamic selection schemes select different classifiers for different test patterns. Nevertheless, it has been shown that traditional dynamic selection performs no better than static selection. We propose four new dynamic selection schemes which explore the properties of the oracle concept. Our results suggest that the proposed schemes, using the majority voting rule for combining classifiers, perform better than the static selection method.

Keywords: Oracle; Combining classifiers; Classifier selection; Ensemble selection; Pattern recognition; Majority voting; Ensemble of learning machines

1. Introduction

The purpose of handwritten pattern recognition systems is to achieve the best possible recognition rate. In general, a number of classifiers are tested in these systems, and the most appropriate one is chosen for the problem at hand. Different classifiers usually make different errors on different samples, which means that, by combining classifiers, we can put together an ensemble that makes more accurate decisions [1–3]. In order to have classifiers with different errors, it is advisable to create diverse classifiers and group them into what is known as an Ensemble of Classifiers (EoC). EoCs have been shown to be useful in improving handwritten recognition.

One way to define the upper limit of EoC's performance is through the concept of oracle. If one classifier from an EoC can correctly classify a given pattern, then this EoC is considered to be able to classify this pattern in oracle. Intuitively, the more diverse the EoC, the better the oracle. The objective of this work is to use the nature of oracle, so we can only select those classifiers which might be capable to correctly classify a given pattern. This is done through a dynamic fashion, since different patterns might require different ensembles of classifiers. Thus, we call our method a dynamic ensemble selection.

To understand better how we can explore the oracle to do a dynamic ensemble selection, first we have to look at how these classifiers are generated. There are several methods for creating diverse classifiers, among them Random Subspaces [4], Bagging and Boosting [5–7]. The Random Subspaces method creates various classifiers by using different subsets of features to train them. Because problems are represented in different subspaces, different classifiers develop different borders for the classification. Bagging generates diverse classifiers by randomly selecting subsets of samples to train classifiers. Intuitively, we would expect classifiers trained by different sample subsets to exhibit different behaviors. The basic Boosting [7] also uses parts of samples to train classifiers, but not randomly; difficult samples have a greater probability of being selected, and easier samples have less chance of being used for training. With this mechanism, most of the classifiers created will focus on hard samples and can be more effective.

One of the most important issues surrounding EoC creation is ensemble selection. The mechanism for doing this is designed to select adequate classifiers from a pool of different classifiers, so that the selected group of classifiers can achieve

[*] Corresponding author. Tel.: +1 514 577 9759.
E-mail addresses: albert@livia.etsmtl.ca (A.H.R. Ko), robert.sabourin@etsmtl.ca (R. Sabourin), alceu@ppgia.pucpr.br (A.S. Britto, Jr.).

0031-3203/$30.00 © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2007.10.015

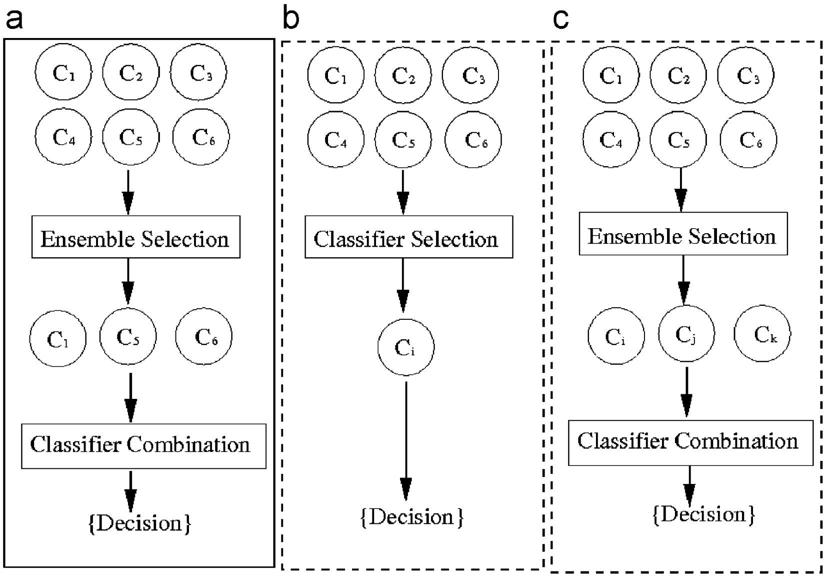

Fig. 1. Three different schemes for selection and combining classifiers: (a) static ensemble selection; (b) dynamic classifier selection; (c) proposed dynamic ensemble selection. The solid line indicates a static process carried out only once for all patterns, and the dash lines indicate dynamic process repeated each time for a different test pattern.

optimum recognition rates. We can perform this task either by static selection, i.e. selecting an EoC for all test patterns, or by dynamic selection, i.e. selecting different EoCs for different test patterns.

The process of static ensemble selection can be further divided into two steps: (a) find a pertinent objective function for selecting the classifiers; and (b) use a pertinent search algorithm to apply this criterion. Obviously, the choice of a pertinent objective function is one of the most crucial elements in selecting pertinent classifiers [1,3,8]. It has been shown that the simple majority voting error (MVE) is one of the best objective functions for this purpose. While there is still no one search algorithm that is universally regarded as better than others, the Genetic Algorithm (GA) is considered to have somewhat of an advantage because of its population-based approach.

However, since different test patterns are, in general, associated with different classification difficulties, it is reasonable to assume that they might be better if they are fit to different classifiers rather than to a single static EoC. This may give us reason to believe that dynamic classifier selection is better than static ensemble selection. The dynamic scheme explores the use of different classifiers for different test patterns [9–15]. Based on the different features or different decision regions of each test pattern, a classifier is selected and assigned to the sample. Some popular methods are A Priori selection, A Posteriori selection, overall local accuracy and local class accuracy [10–12,15], hereafter referred to as the A Priori, A Posteriori, OLA and LCA methods respectively. In general, their performances are compared with that of the oracle, which assigns the correct class label to a pattern if at least one individual classifier from an ensemble produces the correct class label for this pattern. Against all expectations, however, it has been shown that there is a large performance gap between dynamic classifier selection and the oracle [10], and, moreover, dynamic classifier selection does not necessarily give better performance than static ensemble selection [12].

A critical point in dynamic classifier selection is that our choice of one individual classifier over the rest will depend on how much we trust the estimate of the generalization of the classifiers [14]. The advantage of dynamic ensemble selection is that we distribute the risk of this over-generalization by choosing a group of classifiers instead of one individual classifier for a test pattern. So far, this scheme seems to work well.

We note that most dynamic classifier selection schemes use the concept of classifier accuracy on a defined neighborhood or region, such as the local accuracy A Priori or A Posteriori methods [10]. These classifier accuracies are usually calculated with the help of K-nearest neighbor classifiers (KNN), and its use is aimed at making an optimal Bayesian decision. However, KNN could be still outperformed by some static ensemble selection rule, such as the MVE. This poses a dilemma in the estimation of these local accuracies, because their distribution might be too complicated for a good result. Interestingly, dynamic classifier selection is regarded as an alternative to EoC [10,11,15], and is supposed to select the best single classifier instead of the best EoC for a given test pattern. The question of whether or not to combine dynamic schemes and EoC in the selection process is a debate being carried out [14]. But, in fact, the two are not mutually exclusive. Hybrid methods have been shown to be useful, in that they apply the methods for different patterns [13,14]. However, we are interested in exploring another type of approach here, because we believe that ensemble selection can be dynamic as well. This means that, instead of performing dynamic classifier selection, we will perform dynamic ensemble selection (Fig. 1).

We also note that the oracle is usually regarded as a possible upper bound for EoC performances. As far as we know, no effort has been made to explore the appropriateness of the properties of the oracle for dynamic selection. We believe that the complicated process of local classifier accuracy estimation can actually be carried out by the oracle on a validation data set, and a simple KNN method can allow the test data set to obtain the approximate local classifier accuracy from the validation data set. Here are the key questions that need to be addressed:

(1) Can the concept of the oracle be useful in dynamic ensemble selection?
(2) Can dynamic ensemble selection outperform dynamic classifier selection?
(3) Can dynamic ensemble selection outperform static ensemble selection?

To answer these questions, we propose a dynamic ensemble selection scheme which explores the properties of the oracle concept, and compare the scheme with static ensemble selection guided by different objective functions. All the approaches are evaluated on small-scale pattern recognition problems taken from the UCI machine learning repository, and on a large-scale pattern recognition problem related to the recognition of handwritten numerals from NIST SD19. It is important to state that the purpose of this work is not to achieve the best handwritten pattern recognition rate using dynamic selection, but to explore a potential advantage of dynamic selection which might suit the nature of the dynamic environment in machine learning, such as incremental learning. In order to gain a better understanding of the impact of dynamic selection, we use weak classifiers in our experiment.

2. Dynamic classifier selection methods

2.1. Overall local accuracy (OLA)

The basic idea of this scheme is to estimate each individual classifier's accuracy in local regions of the feature space surrounding a test sample, and then use the decision of the most locally accurate classifier [15]. Local accuracy is estimated as the percentage of training samples in the region that are correctly classified.

2.2. Local class accuracy (LCA)

This method is similar to the OLA method, the only difference being that the local accuracy is estimated in respect of output classes [15]. In other words, we consider the percentage of the local training samples assigned to a class cl_i by this classifier that have been correctly labeled.

2.3. A Priori selection method (A Priori)

The classifier accuracy can be weighted by the distances between the training samples in the local region and the test sample. Consider the sample $x_j \in \omega_k$ as one of the K-nearest neighbors of the test pattern X. The $\hat{p}(\omega_k|x_j, c_i)$ provided by classifier c_i can be regarded as a measure of the classifier accuracy for the test pattern X based on its neighbor x_j. If we suppose that we have N training samples in the neighborhood, then the best classifier C_* for classifying the sample X can be selected by [10,12]:

$$C_* = \arg_i \max \frac{\sum_{j=1}^{N} \hat{p}(\omega_k|x_j \in \omega_k, c_i) W_j}{\sum_{j=1}^{N} W_j}, \tag{1}$$

where $W_j = 1/d_j$ is the weight, and d_j is the Euclidean distance between the test pattern X and the its neighbor sample x_j.

2.4. A Posteriori selection method (A Posteriori)

If the class assigned by the classifier c_i is known, then we can use the classifier accuracy in the aspect of the known class. Suppose that we have N training samples in the neighborhood and let us consider the sample $x_j \in \omega_k$ as one of the K-nearest neighbors of the test pattern X. Then, the best classifier $C_*(\omega_k)$ with the output class ω_k for classifying the sample X can be selected by [10,12]:

$$C_*(\omega_k) = \arg_i \max \frac{\sum_{x_j \in \omega_k} \hat{p}(\omega_k|x_j, c_i) W_j}{\sum_{j=1}^{N} \hat{p}(\omega_k|x_j, c_i) W_j}, \tag{2}$$

where $W_j = 1/d_j$ is the weight, and d_j is the Euclidean distance between the test pattern x and the its neighbor sample x_j.

3. K-nearest-oracles (KNORA) dynamic ensemble selection

All the above dynamic selection methods are designed to find the classifier with the greatest possibility of being correct for a sample in a pre-defined neighborhood. We, however, are proposing another approach: Instead of finding the most suitable classifier, we select the most suitable ensemble for each sample.

The concept of the K-nearest-oracles (KNORA) is similar to the concepts of OLA, LCA, and the A Priori and A Posteriori methods in their consideration of the neighborhood of test patterns, but it can be distinguished from the others by the direct use of its property of having training samples in the region with which to find the best ensemble for a given sample. For any test data point, KNORA simply finds its nearest K neighbors in the validation set, figures out which classifiers correctly classify those neighbors in the validation set and uses them as the ensemble for classifying the given pattern in that test set.

We propose four different schemes using KNORA:

(1) KNORA-ELIMINATE
Given K neighbors x_j, $1 \leqslant j \leqslant K$ of a test pattern X, and supposing that a set of classifiers $C(j)$, $1 \leqslant j \leqslant K$ correctly classifies all its K-nearest neighbors, then every classifier $c_i \in C(j)$ belonging to this correct classifier set $C(j)$

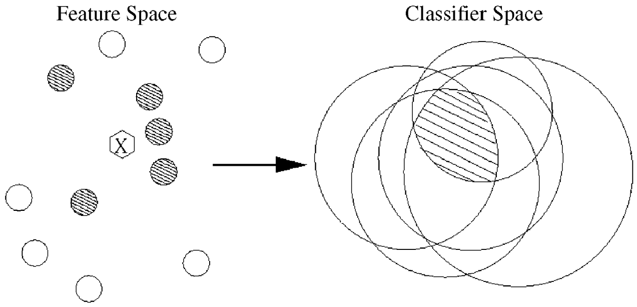

Fig. 2. The KNORA-ELIMINATE only uses classifiers that correctly classify all the K-nearest patterns. On the left side, test pattern is shown as a hexagon, validation data points are shown as circles and the 5 nearest validation points are darkened. On the right side, the used classifiers—the intersection of correct classifiers—are darkened.

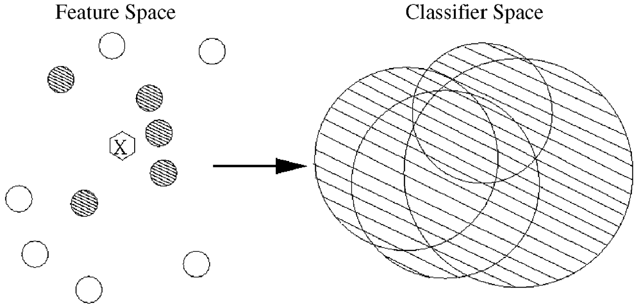

Fig. 3. The KNORA-UNION uses classifiers that correctly classify any of the K-nearest patterns. On the left side, test pattern is shown as a hexagon, validation data points are shown as circles, and the 5 nearest validation points are darkened. On the right side, the used classifiers—the union of correct classifiers—are darkened.

should submit a vote on the sample X. In the case where no classifier can correctly classify all the K-nearest neighbors of the test pattern, then we simply decrease the value of K until at least one classifier correctly classifies its neighbors (Fig. 2).

(2) KNORA-UNION
Given K neighbors x_j, $1 \leqslant j \leqslant K$ of a test pattern X, and supposing that the j-nearest neighbor has been correctly classified by a set of classifiers $C(j)$, $1 \leqslant j \leqslant K$, then every classifier $c_i \in C(j)$ belonging to this correct classifier set $C(j)$ should submit a vote on the sample X. Note that, since all the K-nearest neighbors are considered, a classifier can have more than one vote if it correctly classifies more than one neighbor. The more neighbors a classifier classifies correctly, the more votes this classifier will have for a test pattern (Fig. 3).

(3) KNORA-ELIMINATE-W
This scheme is the same as KNORA-ELIMINATE, but each vote is weighted by the Euclidean distance between the neighbor pattern x_j and the test pattern X.

(4) KNORA-UNION-W
This scheme is the same as KNORA-UNION, but each vote is weighted by the Euclidean distance between the neighbor pattern x_j and the test pattern X.

We should note that these four strategies are relatively simple and intuitive. Nevertheless, these initial strategies will give us a glimpse on whether or not a dynamic ensemble selection might be better than a dynamic classifier selection in some circumstances. It is not our intention to carry out a complete and detailed optimization on the dynamic ensemble selection, and it still remains a question how to select the optimal subset.

3.1. Comparison of dynamic selection schemes on UCI machine learning repository

To ensure that KNORA is useful for dynamic ensemble selection, we tested it on problems extracted from a UCI machine learning repository. There are several requirements for the selection of pattern recognition problems. First, to avoid identical samples being trained in Random Subspace, only databases without symbolic features are used. Second, to simplify the problem, we do not use databases with missing features. In accordance with the requirements listed above, we carried out our experiments on 6 databases selected from a UCI data repository (see Table 1). In general, among the available samples, 50% are used as a training data set and 50% are used as a test data set, except for the Image Segmentation data set, the training data set and test data set of which have been defined on the UCI data repository. Of the training data set, 70% of the samples are used for classifier training and 30% are used for validation.

Three ensemble creation methods have been used in our study: Random Subspaces, Bagging and Boosting [7]. The Random Subspaces method creates various classifiers by using different subsets of features to train them. Bagging generates diverse classifiers by randomly selecting subsets of samples to train classifiers. Similar to Bagging, Boosting uses parts of samples to train classifiers as well, but not randomly. Difficult samples have a greater probability of being selected, and easier samples have less chance of being used for training. The cardinality of Random Subspaces is set under the condition that all classifiers have recognition rates of more than 50%.

The three different classification algorithms used in our experiments are KNN, Parzen windows classifiers (PWC) and quadratic discriminant classifiers (QDC) [16]. For each of 6 databases and for each of 3 classification algorithms, 10 classifiers were generated to constitute the pool of classifiers. We used different dynamic selection schemes to select ensembles from the pools of 10 classifiers, and then combined these ensembles with the simple majority voting rule (MAJ).

3.2. Random Subspace

The Random Subspace method creates diverse classifiers by using different subsets of features to train classifiers. Due to the fact that problems are represented in different subspaces, different classifiers develop different borders for the classification.

For Random Subspace, we observe that KNORA-UNION and LCA have more stable performances than other methods. We also observe that the A Priori and A Posteriori methods are not necessarily better than OLA or LCA (see Tables 2–4).

Table 1
UCI data for ensembles of classifiers

Database	Classes	Tr	Ts	Features	RS-Card.	Bagging (%)	Boosting (%)
Liver-disorders (LD)	2	172	172	6	4	66	66
Pima-diabetes (PD)	2	384	384	8	4	66	66
Wisconsin breast-cancer (WC)	2	284	284	30	5	66	66
Wine (W)	3	88	88	13	6	66	66
Image segmentation (IS)	7	210	2100	19	4	66	66
Letter recognition (LR)	26	10 000	10 000	16	12	66	66

Tr = training samples; Ts = test samples; RS-Card = Random Subspace Cardinality; Bagging = proportion of samples used for Bagging; Boost = proportion of samples used for Boost.

Table 2
Dynamic selection results for Random Subspace using KNN classifiers

Database	KN-E (%)	KN-E-W (%)	KN-U (%)	KN-U-W (%)	A Priori (%)	A Posteriori (%)	OLA (%)	LCA (%)	Oracle (%)	All (%)	Single best (%)
LD	78.47	78.47	80.56	**84.03**	77.78	70.14	79.17	70.83	100.00	76.39	74.31
PD	**97.54**	**97.54**	96.83	96.48	94.37	93.66	96.83	93.66	98.25	96.13	96.83
WC	93.66	93.66	**94.37**	93.66	90.85	80.99	93.31	88.38	99.65	92.61	95.07
W	**97.73**	**97.73**	**97.73**	**97.73**	**97.73**	37.50	**97.73**	**97.73**	97.73	76.14	90.91
IS	78.29	78.29	**78.67**	78.62	75.81	60.90	75.43	59.62	97.29	78.19	84.14
LR	83.33	83.33	83.85	84.20	84.84	87.02	84.84	**87.24**	94.78	83.08	85.32

KN-E = KNORA-ELIMINATE; KN-E-W = KNORA-ELIMINATE-W; KN-U = KNORA-UNION; KN-U-W = KNORA-UNION-W. All = the combined performances of all classifiers. Single best = the performance of the best classifier from the pool. The best classification rates of each method within the neighborhood sizes $1 \leqslant k \leqslant 30$ are shown.
Bold values are the best performances in each row of table.

Table 3
Dynamic selection results for Random Subspace using Parzen classifiers

Database	KN-E (%)	KN-E-W (%)	KN-U (%)	KN-U-W (%)	A Priori (%)	A Posteriori (%)	OLA (%)	LCA (%)	Oracle (%)	All (%)	Single best (%)
LD	71.53	71.53	72.22	**75.00**	**75.00**	65.28	71.53	67.36	89.58	70.83	75.00
PD	**82.82**	**82.82**	82.03	82.29	81.51	65.63	80.99	77.08	92.19	78.12	79.69
WC	92.96	92.96	92.96	92.96	91.20	83.10	**93.31**	87.68	98.94	91.55	92.96
W	88.64	88.64	81.82	89.77	87.50	84.09	89.77	**90.91**	100.00	76.14	88.71
IS	79.90	79.90	80.05	**80.19**	78.10	64.90	77.76	64.76	98.48	79.62	85.38
LR	89.07	89.07	89.68	89.81	**90.51**	88.43	**90.51**	88.49	96.70	89.52	90.61

KN-E = KNORA-ELIMINATE; KN-E-W = KNORA-ELIMINATE-W; KN-U = KNORA-UNION; KN-U-W = KNORA-UNION-W. All = the combined performances of all classifiers. Single best = the performance of the best classifier from the pool. The best classification rates of each method within the neighborhood sizes $1 \leqslant k \leqslant 30$ are shown.
Bold values are the best performances in each row of table.

Table 4
Dynamic selection results for Random Subspace using QDC classifiers

Database	KN-E (%)	KN-E-W (%)	KN-U (%)	KN-U-W (%)	A Priori (%)	A Posteriori (%)	OLA (%)	LCA (%)	Oracle (%)	All (%)	Single best (%)
LD	63.89	63.89	61.11	**70.19**	61.81	70.14	65.28	68.06	88.19	57.64	64.58
PD	**80.21**	**80.21**	**80.21**	**80.21**	79.69	63.28	**80.21**	75.26	93.23	77.86	79.43
WC	95.42	95.42	**95.07**	**95.07**	92.25	88.03	95.42	90.85	99.65	93.66	96.48
W	**98.86**	**98.86**	97.73	**98.86**	97.73	96.59	97.73	95.45	100.00	96.59	96.77
IS	83.29	83.29	81.76	82.19	83.14	39.52	**84.19**	37.76	95.29	78.24	83.24
LR	83.97	83.97	84.62	85.00	81.96	85.99	81.96	**86.73**	93.40	84.36	82.44

KN-E = KNORA-ELIMINATE; KN-E-W = KNORA-ELIMINATE-W; KN-U = KNORA-UNION; KN-U-W = KNORA-UNION-W. All = the combined performances of all classifiers. Single best = the performance of the best classifier from the pool. The best classification rates of each method within the neighborhood sizes $1 \leqslant k \leqslant 30$ are shown.
Bold values are the best performances in each row of table.

Table 5
Dynamic selection results for Bagging using KNN classifiers

Database	KN-E (%)	KN-E-W (%)	KN-U (%)	KN-U-W (%)	A Priori (%)	A Posteriori (%)	OLA (%)	LCA (%)	Oracle (%)	All (%)	Single best (%)
LD	59.03	59.03	**60.42**	**60.42**	58.33	**60.42**	59.03	59.72	79.17	60.42	63.19
PD	74.22	74.22	**74.74**	**74.74**	73.70	72.92	74.22	72.92	90.10	75.00	75.26
WC	**94.72**	**94.72**	93.66	94.01	93.31	92.96	**94.72**	93.31	96.83	93.66	94.72
W	73.86	73.86	73.86	73.86	**75.00**	73.86	73.86	73.86	81.82	72.73	73.86
IS	**87.67**	**87.67**	**87.67**	**87.67**	86.67	85.24	86.52	**87.67**	93.19	86.24	84.57
LR	93.89	93.89	93.94	93.94	93.07	93.97	93.07	**94.05**	97.64	93.76	92.33

KN-E = KNORA-ELIMINATE; KN-E-W = KNORA-ELIMINATE-W; KN-U = KNORA-UNION; KN-U-W = KNORA-UNION-W. All = the combined performances of all classifiers. Single best = the performance of the best classifier from the pool. The best classification rates of each method within the neighborhood sizes $1 \leqslant k \leqslant 30$ are shown.
Bold values are the best performances in each row of table.

Table 6
Dynamic selection results for Bagging using Parzen classifiers

Database	KN-E (%)	KN-E-W (%)	KN-U (%)	KN-U-W (%)	A Priori (%)	A Posteriori (%)	OLA (%)	LCA (%)	Oracle (%)	All (%)	Single best (%)
LD	67.36	67.36	66.67	**68.75**	68.06	61.81	67.36	62.50	94.44	65.28	68.06
PD	**74.74**	**74.74**	72.40	71.88	73.70	74.22	74.22	74.48	84.64	71.88	72.40
WC	94.72	94.72	93.31	93.31	93.31	92.61	**95.07**	92.61	97.18	91.90	94.01
W	73.86	73.86	73.86	73.86	**76.14**	73.86	**76.14**	73.86	85.23	71.59	73.86
IS	**84.62**	**84.62**	82.90	82.95	84.43	82.14	83.76	84.43	89.90	80.00	81.76
LR	94.51	94.51	94.56	**94.58**	93.72	94.17	93.72	94.22	97.63	94.33	92.99

KN-E = KNORA-ELIMINATE; KN-E-W = KNORA-ELIMINATE-W; KN-U = KNORA-UNION; KN-U-W = KNORA-UNION-W. All = the combined performances of all classifiers. Single best = the performance of the best classifier from the pool. The best classification rates of each method within the neighborhood sizes $1 \leqslant k \leqslant 30$ are shown.
Bold values are the best performances in each row of table.

This means that the probabilities weighted by the Euclidean distances between the test pattern and validation patterns are not always useful for dynamic classifier selection.

Similarly, we note that KNORA-UNION-W is not always better than KNORA-UNION. More interestingly, KNORA-ELIMINATE-W and KNORA-ELIMINATE have the same performances on Random Subspaces. This indicates that the probabilities weighted by the Euclidean distances between the test pattern and validation patterns do not affect the decisions of KNORA-ELIMINATE on Random Subspaces.

3.3. Bagging

Bagging generates diverse classifiers by randomly selecting subsets of samples to train classifiers. Intuitively, we can see that classifiers will have different behaviors based on different sample subsets.

For Bagging, we note that KNORA-ELIMINATE, KNORA-UNION and LCA have good performances. As with Random Subspaces, A Priori and A Posteriori are not necessarily better than OLA or LCA on Bagging. Again, KNORA-UNION-W is not always better than KNORA-UNION (see Tables 5–7). This indicates that the probabilities weighted by the Euclidean distances between the test pattern and validation patterns do not always contribute to higher classification rates for either dynamic classifier selection or dynamic ensemble selection.

Still, KNORA-ELIMINATE-W and KNORA-ELIMINATE have the same performances on Bagging.

3.4. Boosting

Boosting uses a part of the samples to train classifiers, but not randomly. As stated above, difficult samples have higher probability of being selected, and easier samples have less chance of being used for training. With this mechanism, most of the classifiers created will focus on hard samples and can be more effective.

For Boosting, KNORA-ELIMINATE, KNORA-UNION and LCA seem to be quite stable. We observe the same situations as for Random Subspaces and Bagging: the A Priori and A Posteriori methods are not necessarily better than OLA or LCA; KNORA-UNION-W is not always better than KNORA-UNION, and KNORA-ELIMINATE-W and KNORA-ELIMINATE have the same performances (see Tables 8–10).

However, these results cannot discount the usefulness of the probabilities weighted by the Euclidean distances between the test pattern and validation patterns, because, in many problems, the number of samples is quite small. Moreover, since there

Table 7
Dynamic selection results for Bagging using QDC classifiers

Database	KN-E (%)	KN-E-W (%)	KN-U (%)	KN-U-W (%)	A Priori (%)	A Posteriori (%)	OLA (%)	LCA (%)	Oracle (%)	All (%)	Single best (%)
LD	**70.83**	**70.83**	63.89	66.67	68.75	61.11	70.14	62.50	91.67	56.94	68.75
PD	74.22	74.22	**74.48**	73.96	73.70	72.66	**74.48**	72.92	83.85	73.96	74.22
WC	97.89	97.89	96.83	96.83	97.54	98.94	97.54	**99.30**	100.00	96.83	98.24
W	**100.00**	**100.00**	98.86	98.86	94.32	94.32	94.32	95.45	100.00	97.73	96.59
IS	**100.00**	**100.00**	99.14	97.33	**100.00**	91.29	**100.00**	**100.00**	100.00	100.00	100.00
LR	89.70	89.70	89.01	88.99	89.64	91.04	89.61	**91.29**	92.81	88.47	88.21

KN-E = KNORA-ELIMINATE; KN-E-W = KNORA-ELIMINATE-W; KN-U = KNORA-UNION; KN-U-W = KNORA-UNION-W. All = the combined performances of all classifiers. Single best = the performance of the best classifier from the pool. The best classification rates of each method within the neighborhood sizes $1 \leqslant k \leqslant 30$ are shown.
Bold values are the best performances in each row of table.

Table 8
Dynamic selection results for Boosting using KNN classifiers

Database	KN-E (%)	KN-E-W (%)	KN-U (%)	KN-U-W (%)	A Priori (%)	A Posteriori (%)	OLA (%)	LCA (%)	Oracle (%)	All (%)	Single best (%)
LD	**66.67**	**66.67**	64.58	65.28	65.97	64.58	65.28	65.28	90.28	62.50	62.50
PD	72.14	72.14	71.88	71.09	73.44	73.44	**75.00**	73.44	91.67	71.09	72.14
WC	95.77	95.77	95.42	**96.13**	95.42	94.72	94.37	95.42	96.83	95.42	95.42
W	73.86	73.86	73.86	73.86	73.86	73.86	73.86	**76.14**	78.41	71.59	73.86
IS	86.57	86.57	86.57	86.57	86.86	86.71	86.86	**87.67**	90.00	86.43	87.67
LR	93.57	93.57	93.79	93.80	92.76	93.95	92.75	**94.00**	97.20	93.62	92.57

KN-E = KNORA-ELIMINATE; KN-E-W = KNORA-ELIMINATE-W; KN-U = KNORA-UNION; KN-U-W = KNORA-UNION-W. All = the combined performances of all classifiers. Single best = the performance of the best classifier from the pool. The best classification rates of each method within the neighborhood sizes $1 \leqslant k \leqslant 30$ are shown.
Bold values are the best performances in each row of table.

Table 9
Dynamic selection results for Boosting using Parzen classifiers

Database	KN-E (%)	KN-E-W (%)	KN-U (%)	KN-U-W (%)	A Priori (%)	A Posteriori (%)	OLA (%)	LCA (%)	Oracle (%)	All (%)	Single best (%)
LD	66.67	66.67	67.36	**72.92**	63.89	63.89	66.67	68.06	100.00	65.97	63.89
PD	74.74	74.74	73.96	73.18	73.70	71.61	**75.00**	73.18	99.74	72.40	73.18
WC	**93.31**	**93.31**	92.96	92.96	92.96	92.96	**93.31**	92.96	94.72	92.96	92.96
W	80.68	80.68	77.27	**81.82**	78.41	73.86	79.55	73.86	95.45	75.00	79.55
IS	84.19	84.19	83.33	83.38	**84.90**	83.76	**84.90**	84.71	88.43	80.48	82.81
LR	94.03	94.03	94.07	94.10	93.02	94.17	92.95	**94.19**	97.29	94.13	93.18

KN-E = KNORA-ELIMINATE; KN-E-W = KNORA-ELIMINATE-W; KN-U = KNORA-UNION; KN-U-W = KNORA-UNION-W. All = the combined performances of all classifiers. Single best = the performance of the best classifier from the pool. The best classification rates of each method within the neighborhood sizes $1 \leqslant k \leqslant 30$ are shown.
Bold values are the best performances in each row of table.

Table 10
Dynamic selection results for Boosting using QDC classifiers

Database	KN-E (%)	KN-E-W (%)	KN-U (%)	KN-U-W (%)	A Priori (%)	A Posteriori (%)	OLA (%)	LCA (%)	Oracle (%)	All (%)	Single best (%)
LD	73.61	73.61	**77.08**	**77.08**	70.14	61.81	73.61	64.58	96.53	70.83	75.00
PD	**75.26**	**75.26**	73.96	74.48	73.70	73.18	74.22	73.96	86.98	74.22	74.74
WC	97.18	97.18	96.83	97.18	95.77	**97.89**	95.77	**97.89**	98.59	96.83	97.89
W	96.59	96.59	96.59	96.59	96.59	**97.73**	96.59	96.59	97.73	96.59	97.73
IS	86.38	86.38	86.52	86.48	86.24	86.43	86.05	**86.57**	90.00	86.43	87.67
LR	93.54	93.54	93.69	93.73	92.63	93.95	92.61	**94.00**	97.20	93.62	92.57

KN-E = KNORA-ELIMINATE; KN-E-W = KNORA-ELIMINATE-W; KN-U = KNORA-UNION; All = the combined performances of all classifiers. Single best = the performance of the best classifier from the pool. The best classification rates of each method within the neighborhood sizes $1 \leqslant k \leqslant 30$ are shown.
Bold values are the best performances in each row of table.

are only 10 classifiers in a pool, there are not many choices for either dynamic classifier selection or dynamic ensemble selection. This might also be a reason why KNORA-ELIMINATE and KNORA-ELIMINATE-W have the same performances.

To conclude, the results demonstrate that using weak classifiers, the dynamic ensemble selection can marginally improve the accuracy, but not always performs better than dynamic classifier selection. According to the 'no free lunch' theorem [17,18], it is understandable that one search algorithm might not always be better than another search algorithm in all problems. However, the unstable performance may be also caused by the simplicity of ensemble selection strategy. Since the oracles in a neighborhood might differ from one another, this will result in the fact that non-optimal ensembles are selected.

Although the experiments suggest that the four KNORA schemes proposed for dynamic ensemble selection might be applicable in various ensemble creation methods—such as Random Subspace, Bagging and Boosting—the problems extracted from the UCI machine learning repository usually consist of a small number of samples with few features. Furthermore, given these constraints, the classifier pool is composed of only 10 classifier in our experiment, which makes the results less convincing. As a result, we were able to justify the need to carry out a larger scale experiment on a problem with more features and larger classifier pools. This is why we conducted our next experiment on a 10-class handwritten-numeral problem with 132 features and 100 classifiers.

4. Experiments for dynamic selection on handwritten numerals

4.1. Experimental protocol for KNN

Our experiments were carried out on a 10-class handwritten-numeral problem. The data were extracted from NIST SD19, essentially as in Ref. [19], based on the ensembles of KNNs generated by the Random Subspaces method. We used nearest-neighbor classifiers ($K = 1$) for KNN, each KNN classifier having a different feature subset of 32 features extracted from the total of 132 features.

To evaluate the static ensemble selection and the dynamic ensemble selection schemes, four databases were used: the training set with 5000 samples (*hsf*_{0 − 3}) to create 100 KNN in Random Subspaces. The optimization set containing 10,000 samples (hsf_{0 − 3}) was used for GA searching for static ensemble selection. To avoid overfitting during GA searching, the selection set containing 10,000 samples (*hsf*_{0 − 3}) was used to select the best solution from the current population according to the objective function defined, and then to store it in a separate archive after each generation. Using the best solution from this archive [20], the test set containing 60,089 samples (*hsf*_{7}) was used to evaluate the EoC accuracies.

We need to address the fact that the classifiers used were generated with feature subsets having only 32 features out of a total of 132. The weak classifiers can help us better observe the effects of EoCs. If a classifier uses all the available features and all the training samples, a much better performance can be observed [10,11]. But, since this is not the objective of this paper, we are focusing on the improvement of EoCs through the optimization of performances by combining classifiers. The benchmark KNN classifier uses all 132 features, and so, with $K = 1$, we can have 93.34% recognition rates. The combination of all 100 KNN by simple MAJ gives 96.28% classification accuracy. The possible upper limit of classification accuracy (the oracle) is defined as the ratio of samples classified correctly by at least one classifier in a pool to all samples. The oracle is 99.95% accurate for KNN.

Table 11
The recognition rates on test data of ensembles searched by GA with the mean classifier error, majority voting error

OF	Min (%)	Q_L (%)	Median (%)	Q_U (%)	Max (%)
ME	94.18	94.18	94.18	94.18	94.18
MVE	96.32	96.41	96.45	96.49	96.57

ME = mean classifier error; MVE = majority voting error; OF = objective functions. Min = minimum accuracy of ensembles; Max = maximum accuracy of ensembles; Q_U = upper quarter accuracy of ensembles; Q_L = lower quarter accuracy of ensembles; Median = median accuracy of ensembles.

4.2. Static ensemble selection with classifier performance

The MVE was tested because of its reputation as one of the best objective functions in selecting classifiers for ensembles [8]. It directly evaluates the global EoC performance by the MAJ. For this reason, we tested the MAJ as the objective function for static and dynamic ensemble selection, as well as using it as the fusion function. We also tested the mean classifier error (ME).

In Table 11, we observe that the MVE performs better than the ME as an objective function for static ensemble selection. The ensemble selected by the MVE also outperforms that of all 100 KNNs.

4.3. Dynamic ensemble selection

Even though the MVE has thus far been able to find the best ensemble for all the samples, this does not mean that a single ensemble is the best solution for combining classifiers. In other words, each sample may have a most suitable ensemble that is different from that of the others. We intend to determine whether or not the use of different ensembles on different samples can further increase the accuracy of the system.

Note that dynamic ensemble selection does not use any search algorithm for selecting the ensemble, because each sample has its own ensemble for the classifier combination. As a result, it was not necessary to repeat the search.

For dynamic ensemble selection, only three databases were used: the training set with 5000 samples (*hsf*_{0 − 3}) to create 100 KNN in Random Subspaces, the optimization set containing 10,000 samples (*hsf*_{0 − 3}) and the test set containing 60,089 samples (*hsf*_{7}) to evaluate the EoC accuracies. We

Table 12
The best recognition rates of each dynamic ensemble selection methods within the neighborhood sizes $1 \leqslant k \leqslant 30$

	KN-E	KN-E-W	KN-U	KN-U-W	OLA	LCA	A Priori	A Posteriori
RR	97.52%	97.52%	97.25%	97.25%	94.11%	97.40%	94.12%	97.40%
K-value	7,8	7,8	1	1	30	1	30	1

RR = recognition rates.

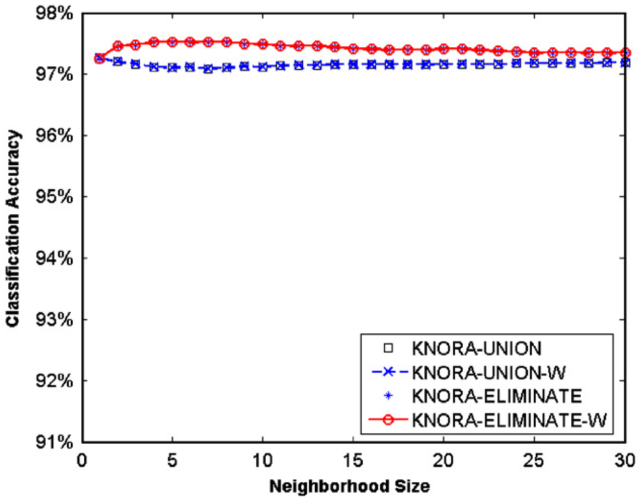

Fig. 4. The performances of proposed dynamic ensemble selection schemes based on different neighborhood sizes $1 \leqslant k \leqslant 30$ on NIST SD19 database. In the figure KNORA-ELIMINATE overlaps with KNORA-ELIMINATE-W, and KNORA-UNION overlaps with KNORA-UNION-W.

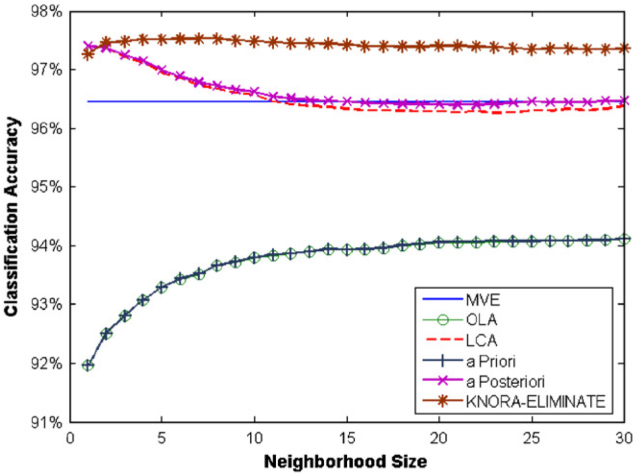

Fig. 5. The performances of various ensemble selection schemes based on different neighborhood sizes $1 \leqslant k \leqslant 30$ on NIST SD19 database. In the figure OLA overlaps with A Priori selection.

tested the four KNORA algorithms and compared them with the other proposed schemes: OLA, LCA, and the A Priori and A Posteriori local class accuracy methods.

We note that most of the dynamic schemes have so far proved better than all the tested objective functions for static ensemble selection. The exceptions are OLA and the A Priori method. Both LCA and the A Posteriori method achieved very good performances, with 97.40% recognition rates. But KNORA-ELIMINATE and KNORA-ELIMINATE-W have good performances as well, and, with recognition rates of 97.52%, KNORA-ELIMINATE and KNORA-ELIMINATE-W turned out to constitute the best dynamic selection scheme for our handwritten-numeral problems (Table 12).

However, KNORA-UNION and KNORA-UNION-W do not perform as well as KNORA-ELIMINATE. They are still better than OLA and the A Priori method, but not as good as LCA and the A Posteriori method (Fig. 4).

If we compare their performances in different neighborhood sizes, we note that, while the LCA and A Posteriori dynamic selection schemes outperform static GA selection with the MVE as the objective function in a small neighborhood, their performances declined with an increase in the value of k (Fig. 5). In this case, static GA selection with the MVE may still be better than the LCA or A Posteriori dynamic selection schemes. By contrast, KNORA-ELIMINATE has a more stable performance, even when the value of k increases. It gives a better recognition rates than all the other schemes in our experimental study, except when $k = 1$. But still, the stable performance of KNORA-ELIMINATE suggests that the dynamic selection schemes are worthy of more attention.

4.4. Effect of validation sample size

Since all the traditional dynamic selection schemes and KNORA take into account the neighborhood of the test pattern for classifier and ensemble selection, the size of the validation samples will have somewhat of an effect on these methods.

We thus varied the size of the validation samples from 1000 to 10,000 samples, and measured the impact of the variation on these dynamic selection schemes. As the number of validation samples increases, a test pattern is more likely to have better nearest neighbors. These nearest neighbors might also better distinguish truly useful classifiers from the pool.

Our results seem to confirm this supposition. When the validation sample size increases, all four proposed KNORA methods show slight improvement (Fig. 6). However, for the traditional dynamic selection schemes, the benefit to be derived from the increase in validation samples seems to be less stable. We observe some fluctuations in classification accuracy on the four traditional dynamic selection schemes when the validation sample size increases (Fig. 7).

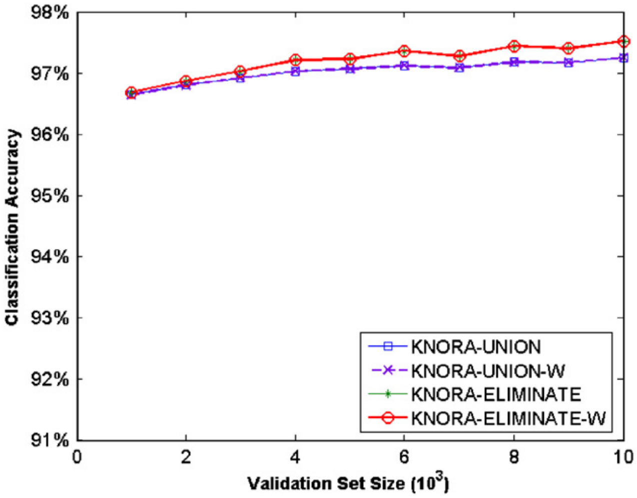

Fig. 6. The performances of proposed dynamic ensemble selection schemes based on different validation sample sizes from 1000 to 10,000 on NIST SD19 database. The best performances from neighborhood sizes $1 \leqslant k \leqslant 30$ are shown. The classifier pool size is 100. In the figure KNORA-ELIMINATE overlaps with KNORA-ELIMINATE-W, and KNORA-UNION overlaps with KNORA-UNION-W.

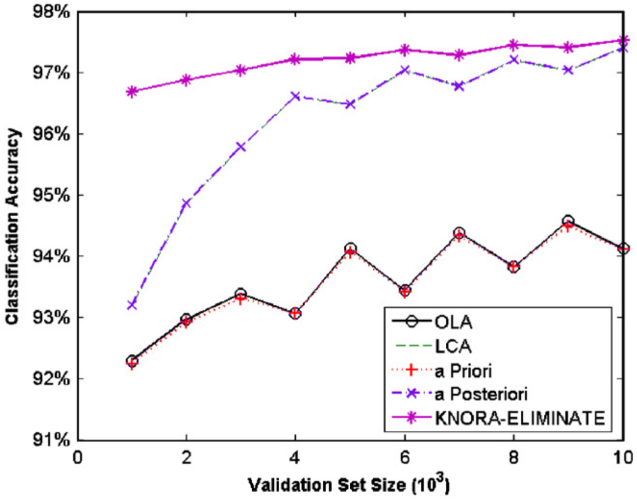

Fig. 7. The performances of various ensemble selection schemes based on different validation sample sizes from 1000 to 10,000 on NIST SD19 database. The best performances from neighborhood sizes $1 \leqslant k \leqslant 30$ are shown. The classifier pool size is 100. In the figure OLA overlaps with A Priori selection, and LCA overlaps with A Posteriori selection.

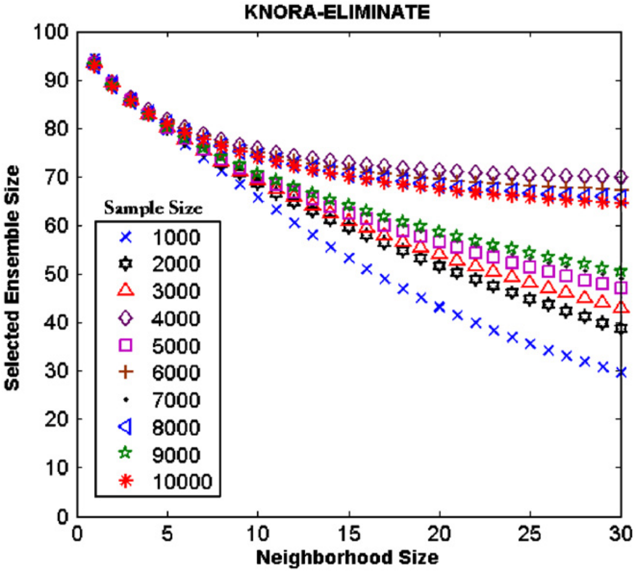

Fig. 8. The relationship between selected ensemble size and neighborhood size on different validation sample sizes from 1000 to 10,000 on NIST SD19 database for KNORA-ELIMINATE. The classifier pool size is 100.

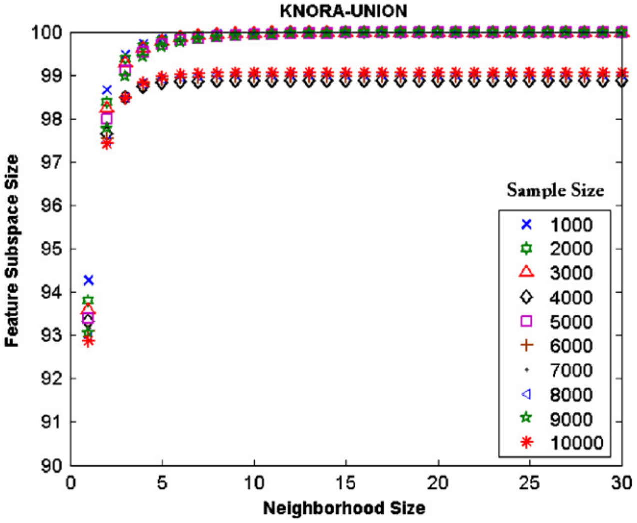

Fig. 9. The relationship between selected ensemble size and neighborhood size on different validation sample sizes from 1000 to 10,000 on NIST SD19 database for KNORA-UNION. The classifier pool size is 100.

The interesting point is that all four KNORA methods demonstrate better performances than other traditional dynamic selection schemes when the validation sample size is small. Also note that the increase in sample size does not necessarily increase the selected ensemble sizes (Figs. 8 and 9).

4.5. *Effect of classifier pool size*

The classifier pool size has a clear effect on the performances of the proposed KNORA methods. While all four of these methods show improvement as the classifier pool size increases, KNORA-ELIMINATE and KNORA-ELIMINATE-W show a better improvement than KNORA-UNION and KNORA-UNION-W (Fig. 10).

Compared with the traditional dynamic selection schemes, we note that KNORA-ELIMINATE is apparently superior to OLA and the A Priori method, but is not necessarily better than LCA or the A Posteriori method (Fig. 11). It is clear that the increase in classifier pool size benefits all kinds of dynamic selection methods, because more classifiers are available. Nevertheless, KNORA-ELIMINATE has shown more improvement than other dynamic selection schemes.

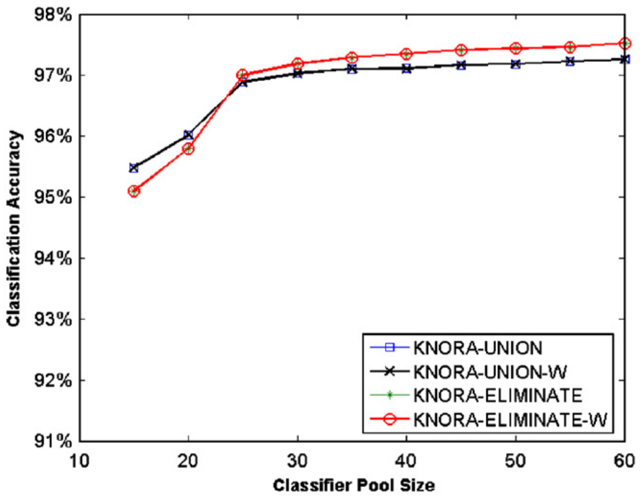

Fig. 10. The performances of proposed dynamic ensemble selection schemes based on different classifier pool sizes from 10 to 100 on NIST SD19 database. The best performances from neighborhood sizes $1 \leqslant k \leqslant 30$ are shown. The validation sample size is 10,000. In the figure KNORA-ELIMINATE overlaps with KNORA-ELIMINATE-W, and KNORA-UNION overlaps with KNORA-UNION-W.

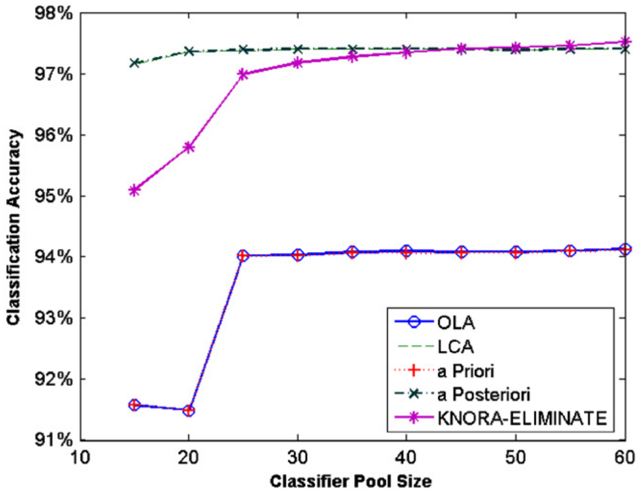

Fig. 11. The performances of various ensemble selection schemes based on different classifier pool sizes from 10 to 100 on NIST SD19 database. The best performances from neighborhood sizes $1 \leqslant k \leqslant 30$ are shown. The validation sample size is 10,000. In the figure OLA overlaps with A Priori selection, and LCA overlaps with A Posteriori selection.

We note that, when there are fewer than 70 classifiers in the pool, LCA and the A Posteriori method outperform KNORA-ELIMINATE. By contrast, when there are more than 70 classifiers in the pool, KNORA-ELIMINATE has a slightly better classification accuracy than LCA and the A Posteriori method.

This is an interesting finding, since it indicates that the KNORA methods are better suited to large classifier pools. Since problems extracted from the UCI machine learning repository use only relatively small classifier pools, this might be why KNORA is not always better than the traditional dynamic selection schemes. Moreover, we also note that the increase in sample size does lead to the increase in the selected ensemble sizes (Figs. 12 and 13).

5. Discussion

In this paper, we propose a new dynamic ensemble selection scheme which directly applies the concept of the oracle on the validation set. Unlike other dynamic selection methods which use the estimated best classifier for a certain data point, the K-nearest oracle uses the EoCs that are estimated to be the best for dynamic ensemble selection.

In our study of handwritten numerals, the proposed method apparently outperforms the static ensemble selection schemes such as the use of the MVE or the ME as the objective function in a GA search. Using the GA search, the MVE can achieve 96.45% recognition rates, and ME 94.18%. Nevertheless, with 97.52% recognition rates, KNORA-ELIMINATE is significantly better than the static ensemble selection methods evaluated.

We note that the OLA and A Priori dynamic selection schemes were not as good as the static GA selection scheme with the MVE. The OLA takes into the account neither class dependence nor the weighting of each classifier, while the A Priori method ignores class dependence. Since our experiment has a high class dimension (10) and the ensemble pool size is quite large (100), it is not surprising that they do not perform well.

We also observe that KNORA-UNION and KNORA-UNION-W perform less well than KNORA-ELIMINATE or KNORA-ELIMINATE-W. This might be due to the extreme elitism in the behavior of the oracle. Since only very few classifiers can correctly classify some difficult patterns, the increase in ensemble size does not lead to a better recognition rate. So, when the value of K increases, the performances of KNORA-UNION and KNORA-UNION-W decline.

KNORA-ELIMINATE also performs slightly better than the other dynamic selection schemes. The LCA and A Posteriori schemes can achieve recognition rates of 97.40%, which is better than the other static methods, but not as good as KNORA-ELIMINATE. However, the performance of KNORA-ELIMINATE is still far from the oracle, which can achieve rates of 99.95%.

This might indicate that addressing the behavior of the oracle is much more complex than applying a simple neighborhood approach, and that the task of figuring out its behavior merely based on the pattern feature space is not an easy one.

Considering the effect of validation sample size, we note that all four KNORA methods demonstrate much better performances than other traditional dynamic selection schemes when the validation sample size is small. On the contrary, classifier pool size has an even more dramatic effect on KNORA performances. In general, when there are few classifiers in the pool, LCA and the A Posteriori method outperform the KNORA methods. However, when the classifier pool size increases, KNORA seems to improve more than LCA and the A Posteriori method. When a number of classifiers is given, KNORA

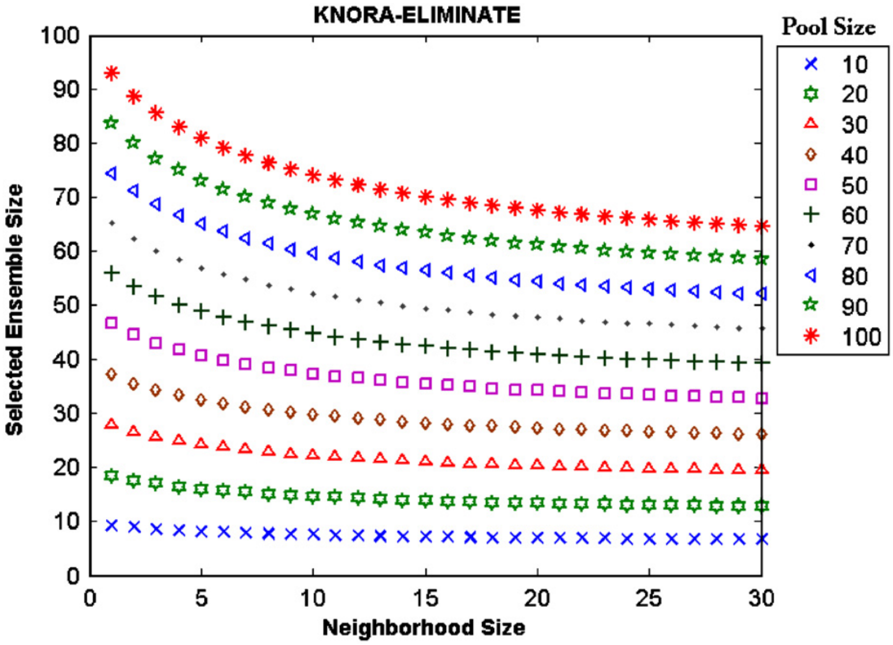

Fig. 12. The relationship between selected ensemble size and neighborhood size on different classifier pool sizes from 10 to 100 on NIST SD19 database for KNORA-ELIMINATE. The validation sample size is 10,000.

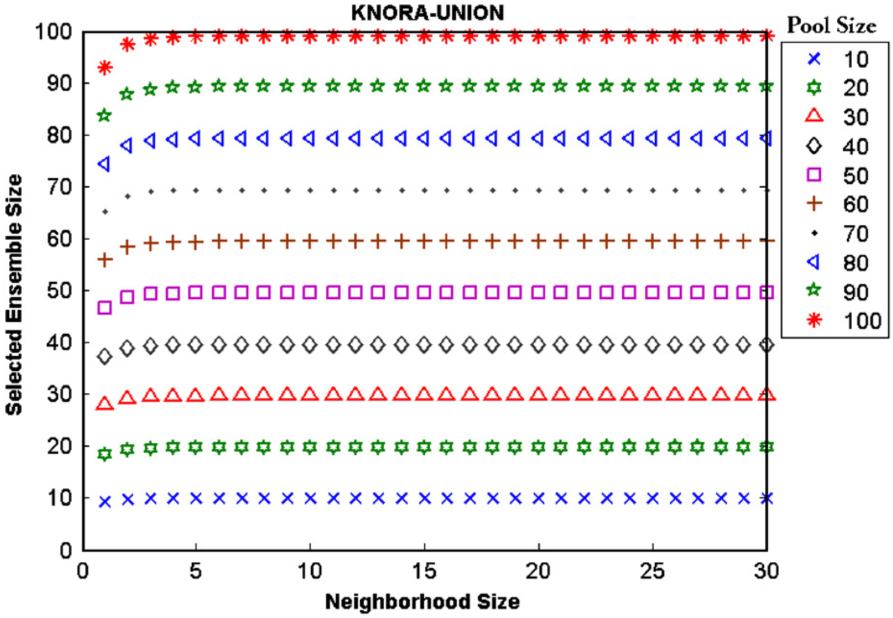

Fig. 13. The relationship between selected ensemble size and neighborhood size on different classifier pool sizes from 10 to 100 on NIST SD19 database for KNORA-UNION. The validation sample size is 10,000.

seems to perform better than either LCA or the A Posteriori method (Fig. 11).

Note that, for an ensemble of M KNN classifiers with N training samples and with total features d and a cardinality of features c (size of fixed feature subspaces), we can first pre-calculate the Euclidean distance on each feature. This pre-calculation has the complexity $O(d \cdot N)$. After the pre-calculation, we only need to carry out the summation and the sorting calculation, which have the complexity $O(M \cdot (c \cdot N + N \log N))$ of the ensemble, rather than the complexity $O(d \cdot N + N \log N)$ of a single KNN classifier. In our study, the best dynamic selection scheme is KNORA-ELIMINATE with the neighborhood size 7, which used 76 classifiers on average, which means that its ensemble is 11.78 times more complex than a single KNN classifier, including the pre-calculation cost. However, the best performance of

KNORA-ELIMINATE is 4.18% better than that of a single KNN classifier.

Finally, we must emphasize that the purpose of this work is not to achieve the best handwritten pattern recognition rate using dynamic selection, but to explore the potential advantages of dynamic selection that might suit the nature of the dynamic environment in machine learning, such as incremental learning. In order to gain a better understanding of the impact of dynamic selection, we use 100 KNN classifiers trained with only 5000 samples in our experimental study. The combination of these 100 KNN by simple MAJ gives only a 96.28% recognition rate. Considering other classification methods applied in the same data set, KNN trained with 150000 samples can achieve 98.57% accuracy, MLP can achieve 99.16% accuracy [21], and the use of SVM can achieve a 99.30% recognition rate with a pairwise coupling strategy and a 99.37% rate with the one-against-all strategy [22]. However, the use of weak classifiers can demonstrate more differences between various ensemble selection schemes, which makes this a better option for comparing different ensemble selection schemes.

6. Conclusion

We describe a methodology to dynamically select an ensemble for every test data point. We find that by the direct use of the concept of the oracle, the proposed scheme apparently gives better performances than static ensemble selection schemes such as GA with the MVE as the objective function. Moreover, the proposed scheme also perform slightly better than other dynamic selection methods in our study.

We show that a dynamic ensemble selection scheme can, in some cases, perform better than some static ensemble selection methods. Furthermore, our study suggests that an ensemble of classifiers might be more stable than a single classifier in the case of dynamic selection. Yet our method is limited by the uncertainty of the behavior of the oracle, since the recognition rates achieved are still not close to those of the oracle.

Our study also arises more questions, such as how to guarantee the optimality and generalization of such selection, since the number of possible combination is very large. The four strategies presented in this paper are simple and straightforward, there is still enough space for further improvements. We believe that this methodology can be greatly enhanced with theoretical studies on the connection between the feature subspaces and the classifier accuracies, the influence of geometrical and topological constraints on the oracle, better statistical studies to quantify the uncertainty of the oracle's behavior and empirical studies in more real-world problems with various ensemble generation methods.

Acknowledgment

This work was supported in part by Grant OGP0106456 to Robert Sabourin from the NSERC of Canada.

References

[1] G. Brown, J. Wyatt, R. Harris, X. Yao, Diversity creation methods: a survey and categorisation, Int. J. Inf. Fusion 6 (1) (2005) 5–20.

[2] J. Kittler, M. Hatef, R. Duin, J. Matas, On combining classifiers, IEEE Trans. Pattern Anal. Mach. Intell. 20 (3) (1998) 226–239.

[3] L.I. Kuncheva, C.J. Whitaker, Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy, Mach. Learn. 51 (2) (2003) 181–207.

[4] T.K. Ho, The random space method for constructing decision forests, IEEE Trans. Pattern Anal. Mach. Intell. 20 (8) (1998) 832–844.

[5] A. Grove, D. Schuurmans, Boosting in the limit: maximizing the margin of learned ensembles, in: Proceedings of the 15th National Conference on Artificial Intelligence, 1998, pp. 692–699.

[6] L.I. Kuncheva, M. Skurichina, R.P.W. Duin, An experimental study on diversity for Bagging and Boosting with linear classifiers, Int. J. Inf. Fusion 3 (2) (2002) 245–258.

[7] R.E. Schapire, Y. Freund, P. Bartlett, W.S. Lee, Boosting the margin: a new explanation for the effectiveness of voting methods, Ann. Stat. 26 (5) (1998) 1651–1686.

[8] D. Ruta, B. Gabrys, Classifier selection for majority voting, Int. J. Inf. Fusion (2005) 63–81.

[9] J. Cao, M. Ahmadi, M. Shridhar, Recognition of handwritten numerals with multiple feature and multistage classifier, Pattern Recognition 28 (2) (1995) 153–160.

[10] L. Didaci, G. Giacinto, F. Roli, G.L. Marcialis, A study on the performances of dynamic classifier selection based on local accuracy estimation, Pattern Recognition 38 (11) (2005) 2188–2191.

[11] L. Didaci, G. Giacinto, Dynamic classifier selection by adaptive K-nearest-neighbourhood rule, International Workshop on Multiple Classifier Systems (MCS 2004), 2004, pp. 174–183.

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

[13] V. Gunes, M. Ménard, P. Loonis, S. Petit-Renaud, Combination, cooperation and selection of classifiers: a state of the art, Int. J. Pattern Recognition Artif. Intell. 17 (8) (2003) 1303–1324.

[14] L.I. Kuncheva, Switching between selection and fusion in combining classifiers: an experiment, IEEE Trans. Syst. Man Cybern. Part B 32 (2) (2002) 146–156.

[15] K. Woods, W.P. Kegelmeyer Jr., K. Bowyer, Combination of multiple classifiers using local accuracy estimates, IEEE Trans. Pattern Anal. Mach. Intell. 19 (4) (1997) 405–410.

[16] R.P.W. Duin, Pattern Recognition Toolbox for Matlab 5.0+, available free at: ⟨ftp://ftp.ph.tn.tudelft.nl/pub/bob/prtools⟩.

[17] D. Whitley, Functions as permutations: regarding no free lunch, Walsh analysis and summary statistics, Parallel Problem Solving from Nature (PPSN 2000), 2000, pp. 169–178.

[18] D.H. Wolpert, W.G. Macready, No free lunch theorems for search, IEEE Trans. Evol. Comput. 1 (1) (1997) 67–82.

[19] G. Tremblay, R. Sabourin, P. Maupin, Optimizing nearest neighbour in Random Subspaces using a multi-objective genetic algorithm, in: Proceedings of the 17th International Conference on Pattern Recognition (ICPR 2004), 2004, pp. 208–211.

[20] P. Radtke, T. Wong, R. Sabourin, An evaluation of over-fit control strategies for multi-objective evolutionary optimization, IEEE World Congress on Computational Intelligence (WCCI 2006)—International Joint Conference on Neural Networks (IJCNN 2006), 2006.

[21] L.S. Oliveira, R. Sabourin, F. Bortolozzi, C.Y. Suen, Automatic recognition of handwritten numerical strings: a recognition and verification strategy, IEEE Trans. Pattern Anal. Mach. Intell. 24 (11) (2002) 1438–1454.

[22] J. Milgram, M. Cheriet, R. Sabourin, Estimating accurate multi-class probabilities with support vector machines, International Joint Conference on Neural Networks (IJCNN 05), 2005, pp. 1906–1911.

About the Author—ALBERT HUNG-REN KO received M.Sc.A. degree in Artificial Intelligence and Pattern Recognition from the Universite Pierre et Marie Curie in 2002. Since 2003, he started his Ph.D. degree in Pattern Recognition in Ecole de Technologie Superieure, Universite du Quebec. His research interests are Ensemble Classification Methods, Small World Structure and Neural Networks.

About the Author—ROBERT SABOURIN received B.ing., M.Sc.A., Ph.D. degrees 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 where he was responsible for the design and development of scientific instrumentation for the Observatoire du Mont Megantic. In 1983, he joined the staff of the Ecole de Technologie Superieure, Universite du Quebec, Montreal, P.Q, Canada, where he is currently a professeur titulaire in the Departement de Genie de la Production Automatisee. In 1995, he joined also the Computer Science Department of the Pontificia Universidade Catolica do Parana (PUC-PR, Curitiba, Brazil) where he was coresponsible since 1998 for the implementation of a Ph.D. program in Applied Informatics. Since 1996, he is 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 for banking and postal applications.

About the Author—ALCEU DE SOUZA BRITTO, JR. received M.Sc. degree in Industrial Informatic from the Federal Center for Technological Education of Parana (Brazil) in 1996, and Ph.D. degree in Computer Science from Pontifical Catholic University of Parana (PUC-PR, Brazil). 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 document analysis and handwriting recognition.