Pattern Recognition 41 (2008) 2993 – 3009

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

A dynamic overproduce-and-choose strategy for the selection of classifier ensembles

Eulanda M. Dos Santos[a,*], Robert Sabourin[a], Patrick Maupin[b]

[a]*École de technologie supérieure, ETS, 1100, Rue Notre-Dame Quest, Montreal, Que, Canada H3C1K3*
[b]*Defence Research and Development Canada (DRDC Valcartier), Canada*

ARTICLE INFO

Article history:
Received 13 December 2007
Accepted 13 March 2008

Keywords:
Overproduce-and-choose strategy
Dynamic classifier selection
Optimization
Measures of confidence

ABSTRACT

The overproduce-and-choose strategy, which is divided into the overproduction and selection phases, has traditionally focused on finding the most accurate subset of classifiers at the selection phase, and using it to predict the class of all the samples in the test data set. It is therefore, a static classifier ensemble selection strategy. In this paper, we propose a dynamic overproduce-and-choose strategy which combines optimization and dynamic selection in a two-level selection phase to allow the selection of the most confident subset of classifiers to label each test sample individually. The optimization level is intended to generate a population of highly accurate candidate classifier ensembles, while the dynamic selection level applies measures of confidence to reveal the candidate ensemble with the highest degree of confidence in the current decision. Experimental results conducted to compare the proposed method to a static overproduce-and-choose strategy and a classical dynamic classifier selection approach demonstrate that our method outperforms both these selection-based methods, and is also more efficient in terms of performance than combining the decisions of all classifiers in the initial pool.

1. Introduction

Classifier ensembles techniques attempt to overcome the complex task of designing a robust, well-suited individual classifier by combining the decisions of relatively simpler classifiers. Two main approaches for the design of classifier ensembles are defined in the literature: (1) classifier fusion; and (2) classifier selection [1–3]. Classifier fusion relies on the assumption that all ensemble members make independent errors. Thus, combining the decisions of the ensemble members may lead to increasing the overall performance of the system. Bagging [4], boosting [5] and the random subspace method [6] are frequently used for the generation of ensemble members, while majority voting, sum, product, maximum and minimum [7] are examples of functions used to combine their decisions. Nonetheless, there is no guarantee that a particular ensemble generation method will achieve error independence. When the condition of independence is not verified, it cannot be assured that the combination of classifier members' decision will improve the final classification performance.

Classifier selection is traditionally defined as a strategy which relies on assuming that each ensemble member is an expert in some local regions of the feature space [1], thereby avoiding the assumption of independence among classifier members. The most locally accurate classifier is selected to estimate the class of each particular test pattern. Two categories of classifier selection techniques exist: static and dynamic. In the first case, regions of competence are defined during the training phase, while in the second case, they are defined during the classification phase taking into account the characteristics of the sample to be classified [2,3]. However, there is a drawback to both selection strategies: when the local expert does not classify the test pattern correctly, there is no way to avoid the misclassification [8]. Moreover, these approaches, for instance *dynamic classifier selection with local accuracy* (DCS-LA) [3], often involve high computing complexity as a result of estimating regions of competence. In addition, they may be critically affected by values of parameters such as the number of neighbors considered for regions defined by k nearest neighbors (kNNs) and distance functions.

Another definition of static classifier selection can be found in the neural network literature. It is called either the *overproduce-and-choose strategy* (OCS) [9] or the *test-and-select methodology* [10]. From this different perspective, the overproduction phase involves the generation of an initial large pool of candidate classifiers, while the selection phase is intended to select the best performing subset of classifiers, which is then used to classify the whole test set. The assumption behind OCS is that candidate classifiers are redundant as an analogy with the feature subset selection problem [11]. In [12], Zhou et al. formally showed that finding the most relevant subset of

* Corresponding author. Tel.: +1 4504627600.
E-mail addresses: eulanda@livia.etsmtl.ca (E.M. Dos Santos), Robert.Sabourin@etsmtl.ca (R. Sabourin), Patrick.Maupin@drdc-rddc.gc.ca (P. Maupin).

0031-3203/$30.00 © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2008.03.027

classifiers is more efficient in terms of performance than combining all the available classifiers.

The overproduction phase may be undertaken using any ensemble generation method and base classifier model. The selection phase, however, is the fundamental issue in OCS, since it focuses on finding the subset of classifiers with optimal accuracy. Although the search for the optimal subset of classifiers can be exhaustive [10], search algorithms might be used when a large initial pool of candidate classifiers $\mathscr{C}$ is involved due to the high computing complexity of an exhaustive search. Given the initial pool $\mathscr{C} = \{c_1, c_2, \ldots, c_n\}$ generated at the overproduction phase, the size of $\mathscr{P}(\mathscr{C})$ is 2^n, where $\mathscr{P}(\mathscr{C})$ is the powerset of $\mathscr{C}$ defining the population of all possible candidate ensembles C_j. Hence, several algorithms have been applied in the literature for the selection phase, ranging from ranking the n best classifiers [13] to *genetic algorithms* (GAs) [11]. Diversity measures, performance [11,13] and ensemble size [11] are frequently employed as search criteria.

OCS is subject to two main problems. First, a fixed subset of classifiers defined using a training/optimization data set may not be well adapted for the classification of the whole test set. This problem is similar to searching for a universal best individual classifier, i.e. due to differences among samples, there is no individual classifier that is perfectly adapted for every test sample. Moreover, as stated by the "No Free Lunch" theorem [14], no algorithm may be assumed to be better than any other algorithm when averaged over all possible classes of problems. The second problem occurs when Pareto-based algorithms, for instance the *fast elitist non-dominated sorting GA* (NSGA-II) [15], are used in the selection phase. These algorithms are efficient tools for OCS due to their capacity to solve *multi-objective optimization problems* (MOOPs) such as the simultaneous use of diversity and classification performance as the objective functions [11]. The Pareto-based algorithms use the Pareto dominance criterion to solve MOOPs. Since a Pareto front is a set of non-dominated solutions representing different tradeoffs with respect to the multiple objective functions, the task of selecting the best candidate ensemble is difficult because all solutions over the Pareto front are equally important. This is a persistent problem in MOOPs applications. Often, only one objective function is taken into account to perform the choice of the best solution. In Ref. [11], for example, the solution with the lowest error rate was picked up to classify the test samples, even though the solutions were optimized regarding both diversity and classification performance measures.

In this paper, we propose a dynamic OCS (DOCS), which relies on combining optimization process and dynamic selection to compose a two-level selection phase (see Fig. 2). In the overproduction phase, an ensemble creation method, such as bagging or the random subspace method, is applied to obtain the initial pool of candidate classifiers $\mathscr{C}$. Thus, at the first level of the selection phase, a population-based search algorithm is employed to generate a population $\mathbf{C}^{*'} = \{C_1, C_2, \ldots, C_w\}$ of highly accurate candidate ensembles C_j. This population is denoted $\mathbf{C}^{*'}$ to indicate that it was obtained by using a validation process, as described in Ref. [11]. Assuming $\mathbf{C}^*$ as the population found using an optimization data set, $\mathbf{C}^{*'}$ is an alternative population found using a validation data set to avoid overfitting. At the second level, the ensembles in $\mathbf{C}^{*'}$ are considered for dynamic selection in order to identify, for each test sample $\mathbf{x}_{i,g}$, the solution $C_j^{*'}$ most likely to be correct for classifying it.

Our objective is to overcome the three drawbacks mentioned above: Rather than selecting only one candidate ensemble found during the optimization level, as is done in *static* OCS (SOCS), the selection of $C_j^{*'}$ is based directly on the test patterns. Our assumption is that the generalization performance will increase, since all potential high accuracy candidate ensembles from the population $\mathbf{C}^{*'}$ are considered to select the most competent solution for each test sample. (1) This is particularly important in problems involving Pareto-based algorithms, because our method allows all equally competent solutions over the Pareto front to be tested; (2) Instead of using only one local expert to classify each test sample, as is done in traditional classifier selection strategies (both static and dynamic), the selection of a subset of classifiers may decrease misclassification; and, finally, (3) Our dynamic selection avoids estimating regions of competence and distance measures in selecting $C_j^{*'}$, since it relies on calculating confidence measures rather than on performance.

In Ref. [16], we first introduced the idea that choosing the candidate ensemble with the largest consensus to predict the test pattern class leads to selecting the solution with greatest certainty in the current decision. We proved both theoretically and experimentally that the selection of the solution with least ambiguity among its members permits an increase in the "degree of certainty" of the classification [17], increasing the generalization performance as a consequence. These interesting results motivated us to investigate two other confidence measures in this paper which also measure the extent of consensus of candidate ensembles: (1) *Margin*, inspired by the definition of margin, measures the difference between the number of votes assigned to the two classes with the highest number of votes, indicating the candidate ensemble's level of certainty about the majority voting class; and (2) *strength relative to the closest class* [18] also measures the difference between the number of votes received by the majority voting class and the class with the second highest number of votes; however, this difference is divided by the performance achieved by each candidate ensemble when assigning the majority voting class for samples contained in a validation data set. This additional information indicates how often each ensemble made the right decision in assigning the selected class. Different ensembles may have different levels of confidence for the same class [18].

Besides these two new confidence measures, ambiguity is also investigated, as is DCS-LA, which was tailored for the selection of classifier ensembles to be compared to the three confidence-based dynamic selection methods. In Ref. [16], we showed that our ambiguity-guided dynamic selection outperformed DCS-LA. However, only the random subspace method was applied to generate the initial pool of *decision tree* (DT) classifiers. In this paper, bagging and the random subspace method are used to generate ensemble members, while DT and kNN classifiers are used for the creation of homogeneous ensembles in the overproduction phase, showing that the validity of our approach does not depend on the particulars of the ensemble generation method. Single- and multi-objective GAs are used to perform the optimization of our DOCS, employing five objective functions for assessing the effectiveness of candidate ensembles. These objective functions comprise four diversity measures and the ensemble's classification error rate.

This paper is organized as follows. Section 2 presents research work related to dynamic classifier selection (DCS). Our proposed DOCS is introduced in Section 3. Then, Sections 4 and 5 describe the optimization and dynamic selection performed in the two-level selection phase by population-based GAs and confidence-based measures, respectively. Finally, the parameters employed in the experiments and the results obtained are presented in Section 6. Conclusions and suggestions for future work are discussed in Section 7.

2. Related Work

Classical DCS methods are divided into three levels, as illustrated in Fig. 1. (1) The *classifier generation* level uses a training data set ($\mathscr{T}$) to obtain classifiers to compose $\mathscr{C}$; (2) *region of competence generation* uses $\mathscr{T}$ or an independent validation data set ($\mathscr{V}$) to produce regions of competence (R_j); and (3) *dynamic selection* chooses a winning partition (R_j^*) and the winning classifier (c_i^*), over samples contained in

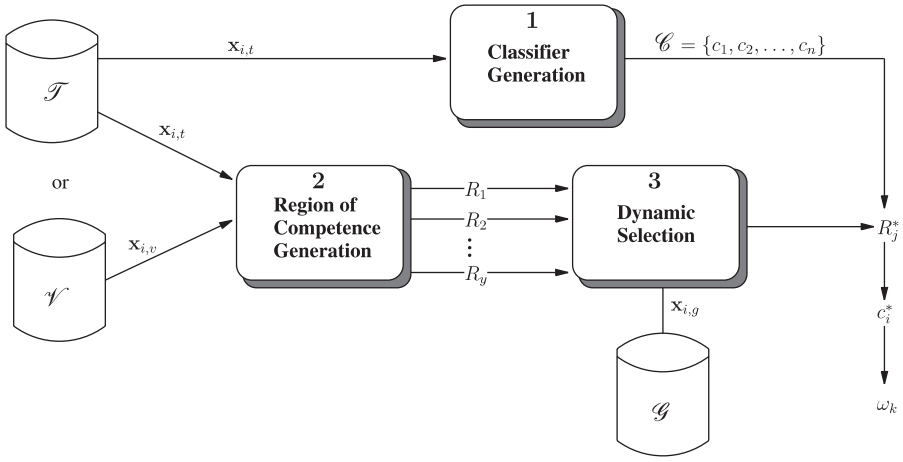

Fig. 1. The classical DCS process: DCS is divided into three levels focusing on training individual classifiers, generating regions of competence and selecting the most competent classifier for each region.

Table 1
Compilation of some of the results reported in the DCS literature highlighting the type of base classifiers, the strategy employed for generating regions of competence, and the phase in which they are generated, the criteria used to perform the selection and whether or not fusion is also used (Het: heterogeneous classifiers)

Reference number	Classifier members	Regions of competence	Partition phase	Selection criteria	Selection/fusion
[1]	DT	Blocks of samples	Training	Accuracy	Selection
[2]	Het	kNN rule	Test	Accuracy	Selection
[3]	Het	kNN rule	Test	Accuracy	Selection
[19]	kNN	Different features	Training	Distance measure	Selection
[21]	MLP	Clustering	Training	Distance and accuracy	Selection
[22]	Het	Clustering	Training	Distance and accuracy	Selection
[23]	MLP/Het	Clustering	Training	Distance and accuracy	Selection or fusion
[24]	Bayesian	Clustering	Training	Distance measure	Selection or fusion
[25]	kNN	kNN rule	Test	Oracle	Selection and fusion
[26]	Het	Clustering	Training	Accuracy and diversity	Selection and fusion

R_j^*, to assign the label ω_k to the sample $\mathbf{x}_{i,g}$ from the test data set ($\mathscr{G}$). The selection is called dynamic because levels 2 and 3 are performed during the test phase, i.e. based on samples $\mathbf{x}_{i,g}$. However, several methods reported in the literature as DCS methods preestimate regions of competence during the training phase [1,19,20] and perform only the third level during the test phase. Thus, the term DCS will hereafter be used to refer to approaches, which assign label ω_k taking into account the test samples, whatever the phase in which the regions of competence are generated.

The main difference between the various DCS methods is the strategy employed to generate regions of competence. kNNs [3], clustering [21] and various training data sets [19] are examples of techniques used. In DCS-LA, proposed by Woods [3], the first level generates a population of five heterogeneous classifiers through feature subset selection. The algorithm defines the local region R_j^* as the set of kNNs from $\mathscr{T}$ surrounding $\mathbf{x}_{i,g}$. Thus, at the third level, the local accuracy of each c_i is estimated, and the most locally accurate classifier c_i^* is then selected to classify $\mathbf{x}_{i,g}$. Giacinto and Roli [2] proposed an approach very similar to Woods' method. The difference is that the local region used to estimate the individual performances of each c_i is defined as the NN from $\mathscr{V}$ that have a similarity with $\mathbf{x}_{i,g}$ that is higher than a threshold. Such a similarity is measured by comparing the vector of class labels assigned by each c_i to $\mathbf{x}_{i,g}$ and to its neighbors.

In the clustering and selection method proposed by Kuncheva [21], *multilayer perceptrons* (MLPs) with different nodes in the hidden layer compose $\mathscr{C}$. Thus, at the second level, the feature space is partitioned into clusters using the K means algorithm, and cluster centroids are computed. At the third level, the region with a cluster center nearest to $\mathbf{x}_{i,g}$ is picked up as R_j^* and the c_i with the highest classification accuracy is nominated to label $\mathbf{x}_{i,g}$. Liu et al. [22] presented a clustering and selection-based method that first generates three heterogeneous classifiers to compose $\mathscr{C}$. At the second level, $\mathscr{T}$ is divided into two groups for each c_i: (1) correctly classified training samples; and (2) misclassified training samples. These two groups are further partitioned using a clustering algorithm to compose regions of competence, i.e. each c_i has its own. At the dynamic selection level, the cluster closest to $\mathbf{x}_{i,g}$ from each of c_i's regions of competence is pointed out and the most accurate classifier is chosen to assign $\mathbf{x}_{i,g}$'s label.

Singh and Singh [19] described a DCS method for image region labeling in which the first level generates $\mathscr{C}$ by training each c_i (kNN classifiers) with n different training data sets which are obtained through applying n different texture analysis methods. The regions of competence are defined at the second level as the class centroids of each training data set. The selection level measures the distance between $\mathbf{x}_{i,g}$ and all the class centroids. Then, the c_i responsible by the closest region is selected to classify $\mathbf{x}_{i,g}$. In Ref. [1], a DCS method for data stream mining applications is proposed. For the first level, $\mathscr{T}$ is divided into n chunks which are further used to train the n DT classifiers that compose $\mathscr{C}$. Local regions are generated with statistical information on the attribute values of samples from $\mathscr{V}$. Finally, at the third level, the most accurate classifier in the region sharing the same statistical information on attribute values with $\mathbf{x}_{i,g}$ is selected to label it.

It is important to mention that all these methods pick up only one candidate classifier to make the decision. This may lead to a classifier with a low level of confidence in its decision, or even one

with a wrong decision, being chosen. A combination of selection and fusion has been investigated in the literature as a strategy for avoiding this drawback. Kuncheva [23] proposed to use statistical tests to switch between selection and fusion. The classifier c_i^*, selected using clustering and selection [21], is employed to label $\mathbf{x}_{i,g}$ only when it is significantly better than the remaining classifiers. Otherwise, all classifiers in $\mathscr{C}$ are combined through decision templates. Gunes et al. [24] applied a fuzzy clustering algorithm in combination with ambiguity rejection in order make it possible to deal with overlapping regions of competence. They switch between classifying $\mathbf{x}_{i,g}$ using either c_i^* or the combination of the best adapted classifiers whether $\mathbf{x}_{i,g}$ falls into a single cluster or into an ambiguous cluster, respectively. The k-nearest-oracles (KNORA) method proposed by Ko et al. [25] explores the properties of the oracle concept [37] to select the most suitable classifier ensemble for each test sample. KNORA first finds the set of kNNs from $\mathscr{V}$ surrounding $\mathbf{x}_{i,g}$. Then the algorithm selects each classifier c_i, which correctly classifies this set of neighbors, to compose a classifier ensemble. This selected classifier ensemble is then used for classifying $\mathbf{x}_{i,g}$. Finally, Soares et al. [26] have tailored Kuncheva's method [23] to select candidate classifiers based on accuracy and diversity.

Table 1 summarizes the DCS methods reported in the literature and mentioned in this section. It is interesting to note that heterogeneous classifiers at the first level, clustering at the second level and accuracy as a selection criterion at the third level are most often applied. The method proposed in this paper does not follow the classical definition of DCS, however, and so a partition generation procedure is not needed and the selection level is not based on accuracy. Instead, our approach is based on the definition of OCS. We propose a *dynamic* OCS that combines, in the selection phase, an optimization process to generate candidate classifier ensembles and a dynamic selection performed by calculating a measure of confidence. Our method is described in the following section.

3. The proposed dynamic OCS

Traditionally, OCS is divided into two phases: (1) overproduction; and (2) selection. The former is devoted to constructing $\mathscr{C}$. The latter tests different combinations of these classifiers in order to identify the optimal solution, $C_j^{*'}$. Fig. 2 shows that, in SOCS, $C_j^{*'}$ is picked up from the population of candidate ensembles, $\mathbf{C}^{*'}$, found and analyzed during the selection phase, and is used to classify all samples contained in $\mathscr{G}$. However, as mentioned in the introduction, there is no guarantee that the $C_j^{*'}$ chosen is indeed the solution most likely to be the correct one for classifying each $\mathbf{x}_{i,g}$ individually.

We propose a DOCS in this paper, as summarized in Fig. 2 and Algorithm 1. In the overproduction phase, $\mathscr{T}$ is used by an ensemble creation method to generate $\mathscr{C}$. Thus, the selection phase is divided into two levels: (1) optimization; and (2) dynamic selection. In this paper, the optimization is performed by GAs guided by both single- and multi-objective functions. In order to avoid overfitting at this level, we employ the same global validation strategy as described in [11]. According to this strategy, an optimization data set ($\mathscr{O}$) is used by the search algorithm to calculate fitness, and $\mathscr{V}$ is used to keep stored, in an auxiliary archive, the population of n best solutions for the GA or the Pareto front for NSGA-II found before overfitting starts to occur. Such a population of solutions, $\mathbf{C}^{*'}$, is further used at the dynamic selection level, which allows the dynamic choice of $C_j^{*'}$ to classify $\mathbf{x}_{i,g}$, based on the certainty of the candidate ensemble's decision. Finally, $C_j^{*'}$ is combined by majority voting.

Algorithm 1. Dynamic Overproduce and Choose Strategy (DOCS).

1: Design a pool of classifiers $\mathscr{C}$.
2: Perform the optimization level using a search algorithm to generate a population of candidate ensembles $\mathbf{C}^{*'}$.
3: **for** each test sample $\mathbf{x}_{i,g}$ **do**
4: **if** all candidate ensembles agree on the label **then**
5: classify $\mathbf{x}_{i,g}$ assigning it the consensus label.
6: **else**
7: perform the dynamic selection level calculating the confidence of solutions in $\mathbf{C}^{*'}$;
8: **if** a winner candidate ensemble is identified
9: select the most competent candidate ensemble $C_j^{*'}$ to classify $\mathbf{x}_{i,g}$
10: **else**
11: **if** a majority voting class among all candidate ensembles with equal competence is identified **then**

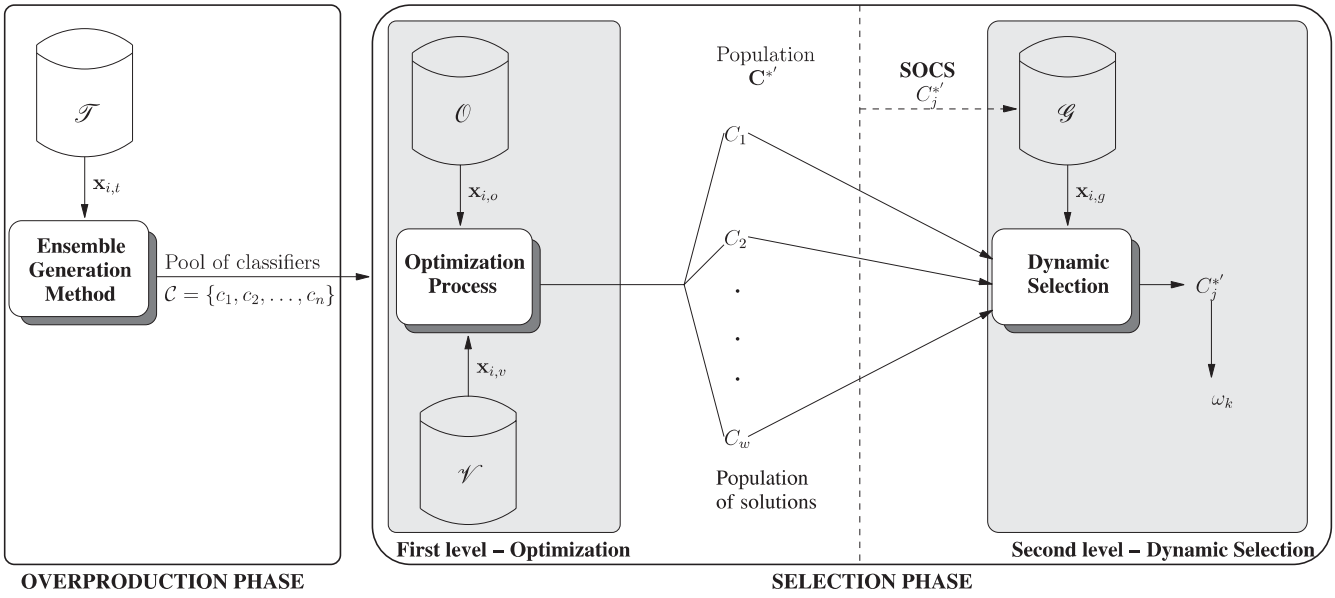

Fig. 2. Overview of the proposed DOCS. The method divides the selection phase into an optimization level, which yields a population of ensembles, and a dynamic selection level, which chooses the most competent ensemble for classifying each test sample. In SOCS, only one ensemble is selected to classify the whole test data set.

```
12:            assign the majority voting class to x_{i,g}
13:          else
14:            select the second highest competent candidate
               ensemble
15:            if a majority voting class among all candidate
               ensembles with the first and the second highest
               competence is identified then
16:              assign the majority voting class to x_{i,g}
17:            else
18:              randomly select a candidate ensemble to
                 classify x_{i,g}
19:            end if
20:          end if
21:        end if
22:      end if
23:  end if
```

3.1. Overproduction Phase

In the overproduction phase, any ensemble generation technique may be used, such as varying the classifier type [13], the classifier architecture [10], the learning parameter initialization [1], boosting [27], the random subspace method [11], etc. In this paper, we employ *bagging* (BAG) and the *random subspace* (RSS) method to generate $\mathscr{C}$. RSS [6] works by randomly choosing n different subspaces from the original feature space. Each random subspace is used to train one individual classifier c_i. BAG is a bootstrap technique [4] which builds n replicate training data sets by randomly sampling, with replacement, from $\mathscr{T}$. Thus, each replicated data set is used to train one c_i.

3.2. Selection phase

The selection phase may be conducted by exhaustive enumeration [10] or heuristic rules [9], or as an optimization problem guided by search algorithms [13,11]. As shown in Fig. 2, the selection phase in our method is divided into two levels, optimization and dynamic selection, as described in the following sections.

4. Optimization level

In this section, we discuss the search criterion and the search algorithm, which are the two main factors analyzed when dealing with the optimization level.

4.1. Search Criteria

Ensemble combination performance, ensemble size and diversity measures are the most frequent search criteria employed in the literature [11,13]. Performance is the most obvious of these, since it allows the main objective of pattern recognition, i.e. finding predictors with a high recognition rate, to be achieved. In terms of ensemble size, we have shown in Ref. [11] that the reduction in the number of classifiers is a consequence of the optimization task, whatever the objective function used to guide the search. Hence, there is no need to explicitly include ensemble size in the optimization level. Finally, the important role played by diversity is clearly defined in the literature, because ensembles of classifiers are more accurate than individual classifiers only when classifier members present diversity among themselves [28]. Although the relationship between diversity and performance is unclear, OCS offers a way of explicitly using diversity to improve performance, leading to the selection of accurate and diverse candidate ensembles. From these standpoints, the optimization level of our DOCS is performed by using only minimization of the error rate (1-performance) and maximization of diversity as search criteria.

Various approaches to defining diversity have been proposed. Kuncheva and Whitaker [28] three groups of diversity measures among the 10 investigated: (1) double fault; (2) coincident failure diversity; and (3) the remaining eight. We chose to use the double fault (δ) and coincident failure diversity (σ), as well as the difficulty measure (θ) from the third group. In addition, we selected ambiguity (γ) (as defined in Ref. [29]), which was not investigated in Ref. [28]. It is worth noting that *dissimilarity* measures, such as γ and σ, must be maximized, while *similarity* measures, such as θ and δ, must be minimized when used as objective functions during the optimization process. In addition, σ, θ and γ are calculated on the whole candidate ensemble, while δ is calculated for each pair of classifiers, since it is a pairwise measure. These measures are described below:

(a) *Ambiguity*: Given the candidate ensemble $C_j = \{c_1, c_2, \ldots, c_l\}$ and its output ω_k, the ambiguity of the i-th classifier for sample x is calculated as follows:

$$a_i(x) = \begin{cases} 0 & \text{if } y_i = \omega_k, \\ 1 & \text{otherwise,} \end{cases} \tag{1}$$

where y_i is the output of the i-th classifier. If we let X be the data set, $|X|$ its cardinality and $|C_j|$ the ensemble's size, then the ambiguity for the whole C_j is as follows:

$$\gamma = \frac{1}{|X|.|C_j|} \sum_{i \in C_j} \sum_{x \in X} a_i(x). \tag{2}$$

(b) *Coincident failure diversity*: This measure is based on the same distribution F as proposed for θ. Here, however, $Y = i/|C_j|$ denotes the proportion of classifiers that do not correctly classify x. If $p(q)$ is the probability that q classifiers will fail when classifying x, this measure is defined as follows:

$$\sigma = \begin{cases} 0, & p_0 = 1.0, \\ \dfrac{1}{(1-p_0)} \sum_{q=1}^{|C_j|} \dfrac{q}{|C_j|} \dfrac{(q-1)}{(|C_j|-1)} p_q, & p_0 < 0. \end{cases} \tag{3}$$

(c) *Difficulty measure*: Letting F be calculated from $\{0/|C_j|, 1/|C_j|, \ldots, 1\}$, which represents the number of classifiers in C_j that correctly classify x, this measure may be calculated as follows:

$$\theta = Var(F) \tag{4}$$

(d) *Double-fault*: Let N^{ab} be the number of examples classified in X, where a, b determine whether the classifier is correct (1) or not (0). This pairwise measure is defined for a pair of classifiers c_i and c_k as follows:

$$\delta_{i,k} = \frac{N^{00}}{N^{11} + N^{10} + N^{01} + N^{00}} \tag{5}$$

4.2. Search algorithms: single- and multi-objective GAs

Evolutionary algorithms are attractive search algorithms, since they allow classifier ensemble selection tasks to be implemented quite easily as optimization processes. Besides, they generate populations of candidate ensembles which may be analyzed at the dynamic selection level of our DOCS. (1) Single- and (2) *multi-objective* GA (MOGA) are the two strategies available when dealing with GAs. These two strategies are discussed in this section. In order to clarify the proposed DOCS, we will use a case study obtained in one replication using the NIST-letters database (Section 6) throughout this paper. The initial pool of DT classifiers was generated using RSS. The maximum number of generations is fixed at $max(g) = 1000$ and the size of the population of individuals for both GAs is 128.

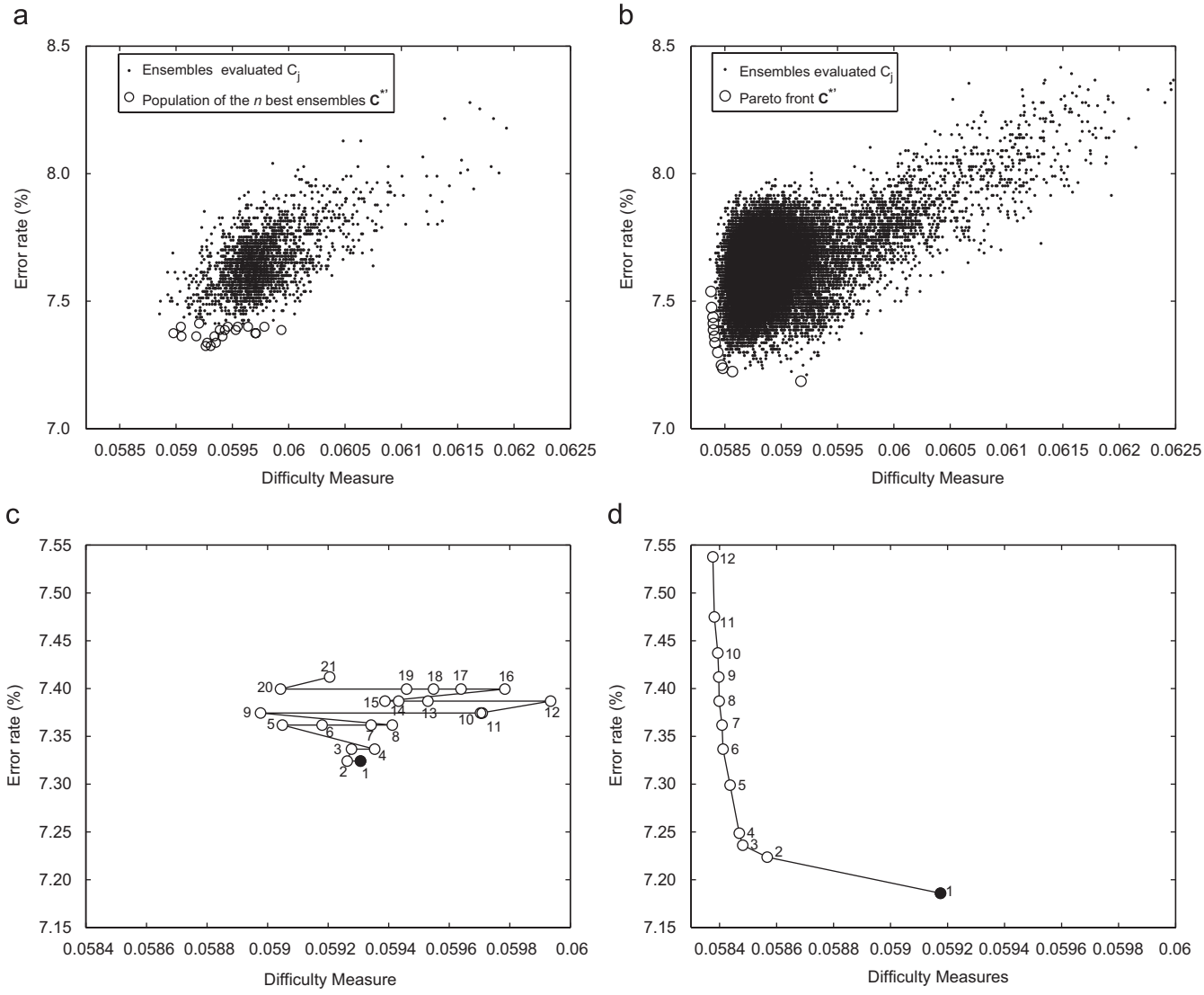

Fig. 3. Ensembles generated using single-objective GA guided by ε in Fig. 3(a) and NSGA-II guided by θ and ε in Fig. 3(b). The output of the optimization level obtained as the n best solutions by GA in Fig. 3(c) and the Pareto front by NSGA-II in Fig. 3(d). These results were calculated for samples contained in $\mathcal{Y}$. The black circles indicate the solutions with lowest ε, which are selected when performing SOCS. (a) GA—Evaluated solutions, (b) NSGA-II—Evaluated solutions, (c) GA—n best solutions and (d) NSGA-II—Pareto front.

When the optimization level is conducted as a single-objective problem, GA is guided by an objective function for assessing the effectiveness of candidate ensembles. Fig. 3a depicts this process. Although GA was guided by the minimization of the error rate (ε) as the objective function, we show plots of ε versus θ to better illustrate the process. Each point on the plot corresponds to a candidate ensemble, i.e. they represent all solutions evaluated for $max(g)$. The population $\mathbf{C}^{*'}$ (circles) is composed of the n best solutions. We fixed $n = 21$ (see Section 6). In Fig. 3(c), only the 21 best solutions are shown.

In SOCS, $C_j^{*'}$ is assumed to be the solution with the lowest ε, black circle in Fig. 3(c), without knowing whether the $C_j^{*'}$ chosen is indeed the best solution for correctly classifying each $\mathbf{x}_{i,g}$. Hence, the additional dynamic selection level proposed in this paper is a post-processing strategy which takes advantage of the possibility of dealing with a set of high-performance solutions rather than only one. In this way, the whole population $\mathbf{C}^{*'}$ is picked up at the dynamic selection level of our method. The parameter n should be defined experimentally.

MOGAs are often solutions to optimization processes guided by multi-objective functions. Since the combination of ε and diversity measures as search criteria has been investigated in the literature as a strategy for selecting accurate and diverse candidate ensembles [30,29], MOGAs allow the simultaneous use of both measures to guide the optimization level in our method. These algorithms use Pareto dominance to reproduce the individuals. The selection of $C_j^{*'}$ is more difficult in this context, since a Pareto front is a set of non-dominated solutions representing different tradeoffs between the multiple objective functions. In our optimization process, a candidate ensemble solution C_i is said to dominate solution C_j, denoted $C_i \preccurlyeq C_j$, if C_i is no worse than C_j on all the objective functions and C_i is better than C_j in at least one objective function. Based on this non-domination criterion, solutions over the Pareto front are considered to be equally important.

Among several Pareto-based evolutionary algorithms proposed in the literature, NSGA-II (elitist non-dominated sorting GA) [15] appears to be interesting because it has two important characteristics: a full elite-preservation strategy and a diversity-preserving

Table 2
Case study: the results obtained by GA and NSGA-II when performing SOCS are compared with the result achieved by combining all classifiers in the initial pool $\mathscr{C}$

Optimization level	Error rate	$C_j^{*'}$ size	Average size of $C_j \in \mathbf{C}^{*'}$
Combination	6.06	100	–
NSGA-II (ε & θ)	5.86	45	50 (2.50)
GA (ε)	5.90	55	55 (3.89)

mechanism using the crowding distance as the distance measure. The crowding distance does not require any parameter to be set [15]. Elitism is used to provide a means to keep good solutions among generations, and the diversity preservation mechanism is used to allow a better spread among the solutions over the Pareto front.

NSGA-II [15] works as follows: At each generation step g, a parent population $\mathbf{C}(g)$ of size w evolves and an offspring population $\mathbf{C}^q(g)$, also of size w, is created. These two populations are combined to create a third population $\mathbf{C}^r(g)$ of size $2w$. The population $\mathbf{C}^r(g)$ is sorted according to the non-dominance criteria and different non-dominated fronts are obtained. Then, the new population $\mathbf{C}(g+1)$ is filled by the fronts according to the Pareto ranking. In this way, the worst fronts are discarded, since the size of $\mathbf{C}(g+1)$ is w. When the last front allowed to be included into $\mathbf{C}(g+1)$ has more solutions than the $\mathbf{C}(g+1)$'s available free space, crowding distance is measured in order to select the most isolated solutions in the objective space to increase diversity. Algorithm 2 summarizes NSGA-II.

Algorithm 2. Fast elitist non-dominated sorting GA (NSGA-II)

```
Create initial population C(1) of w chromosomes
while g < max(g) do
    create C^q(g)
    set C^r(g) = C(g) ∪ C^q(g)
    perform a non-dominated sorting to C^r(g) and
        identify different fronts C_k, k = 1, 2, ..., etc
    while |C(g+1)| + |C_k| ⩽ w do
        set C(g+1): = C(g+1) ∪ C_k
        set k: = k + 1
    end while
    perform crowding distance sort to C_k
    set C(g+1): = C(g+1) ∪ C_k⌊1 : (w − |C(g+1)|)⌋
    create C^q(g+1) from C(g+1)
    set g: = g + 1
endwhile
```

Fig. 3(b) shows all the classifier ensembles evaluated using NSGA-II guided by the following pair of objective functions: jointly minimize θ and ε. Here, the Pareto front is assumed to be $\mathbf{C}^{*'}$, circles in Figs. 3(b) and (d). Although the solutions over the Pareto front are by definition equally important, the candidate ensemble with the lowest ε, the black circle in Fig. 3(d), is usually chosen to be $C_j^{*'}$ in classical SOCS, as was done in Refs. [11,30].

Considering this case study, Table 2 shows the results calculated using samples from $\mathscr{G}$ comparing SOCS and the combination of the initial pool of classifiers $\mathscr{C}$. The selection of a subset of classifiers using SOCS outperformed the combination of $\mathscr{C}$. It is interesting to observe that NSGA-II was slightly superior to GA, and that the sizes of the candidate ensembles in $\mathbf{C}^{*'}$ found using both GAs were smaller than the initial pool size. In addition, NSGA-II found a $C_j^{*'}$ even smaller than the solution found by GA. The assumption of proposing an additional dynamic selection level is that performance may still be increased when selecting $C_j^{*'}$ dynamically for each $\mathbf{x}_{i,g}$.

5. Dynamic selection level

The selection process in classical DCS approaches is based on the certainty of the classifiers decision for each particular $\mathbf{x}_{i,g}$. Consequently, these methods explore the domain of expertise of each classifier to measure the degree of certainty of its decision, as described in Section 2. The dynamic selection level proposed in this paper is also based on decision certainty. However, instead calculating the confidence of each individual classifier, our method calculates the confidence of each candidate ensemble that composes $\mathbf{C}^{*'}$, when assigning a label for $\mathbf{x}_{i,g}$. In Ref. [16], we have shown that it is possible to calculate the certainty of a candidate ensemble decision by measuring the extent of the consensus associated with it. The standpoint is that the higher the consensus among classifier members, the higher the level of confidence in the decision.

Considering a classification problem with the following set of class labels $\Omega = \{\omega_1, \omega_2 \ldots, \omega_c\}$, the confidence level is related to the *posterior probability* $P(\omega_k|\mathbf{x}_{i,g})$ that $\mathbf{x}_{i,g}$ comes from class ω_k. Hansen et al. [17] have observed that $P(\omega_k|\mathbf{x}_{i,g})$ may be calculated in the context of an ensemble of classifiers by measuring the extent of the consensus of ensembles, as given below:

Given the candidate ensemble $C_j = \{c_1, c_2, \ldots, c_l\}$ and the output of the i-th classifier $y_i(\mathbf{x}_{i,g})$, without loss of generality, we can assume that each classifier produces a class label as output. The number of votes $v(\omega_k|\mathbf{x}_{i,g})$ for class ω_k given $\mathbf{x}_{i,g}$ is obtained as follows:

$$v(\omega_k|\mathbf{x}_{i,g}) = |\{c_i : y_i(\mathbf{x}_{i,g}) = \omega_k\}| \tag{6}$$

Assuming majority voting as the combination function, the consensus decision is

$$mv(\mathbf{x}_{i,g}) = \underset{k=1}{\overset{c}{\arg\max}}\, v(\omega_k|\mathbf{x}_{i,g}) \tag{7}$$

Thus, the extent of consensus on sample $\mathbf{x}_{i,g}$ is

$$P(\omega_k|\mathbf{x}_{i,g}) = \frac{v(mv(\mathbf{x}_{i,g})|\mathbf{x}_{i,g})}{|C_j|} \tag{8}$$

The extent of consensus measures the number of classifiers in agreement with the majority voting. Consequently, by maximizing the extent of consensus of an ensemble, the degree of certainty that it will make a correct classification is increased. Another important point to mention is that no information on the correctness of the output is needed. These observations allow us to present three confidence measures that calculate the extent of consensus of each candidate ensemble from the population $\mathbf{C}^{*'}$, to be used at the dynamic selection level of our DOCS: (1) ambiguity, (2) margin, and (3) strength relative to the closest class. The first two measures are directly related to the evaluation of the extent of consensus. The third measure also considers the candidate ensembles' performance measured for each class involved in the classification problem. This additional information is calculated over samples contained in $\mathscr{V}$. In addition, DCS-LA is adapted to the context of DOCS to be compared to these three confidence-based strategies.

5.1. Ambiguity-guided dynamic selection (ADS)

The classification ambiguity proposed by Zenobi and Cunningham [29] attempts to estimate the diversity of opinions among classifier members. This diversity measure, which is defined in Eq. (2), Section 4.1, appears to be well suited for the dynamic selection level we are proposing, since it does not need knowledge on the correctness of the decision.

It is important to note in Eq. (2) that, if we calculate the ambiguity γ for a particular test sample $\mathbf{x}_{i,g}$ instead of calculating it for the whole data set, γ becomes the complement of the extent of consensus

in Eq. (8). Denoting γ calculated for the given $\mathbf{x}_{i,g}$ as $\bar{\gamma}$, we may assume $\bar{\gamma} + P(\omega_k|\mathbf{x}_{i,g}) = 1$. Thus, Eq. (2) simplifies to

$$\bar{\gamma} = \frac{1}{|C_j|} \sum_{i \in C_j} a_i(\mathbf{x}_{i,g}) \tag{9}$$

Since such a local ambiguity measures the number of classifiers in disagreement with the majority voting, the minimization of $\bar{\gamma}$ leads to the maximization of the extent of consensus. Consequently, the certainty of correct classification is increased. In addition, although $\bar{\gamma}$ does not take into account the label of the given sample, the minimization of $\bar{\gamma}$ also leads to the maximization of the margin in the case of a correct classification. The so-called margin is a measure of confidence of classification. There are two general definitions of margin reported in the literature [31]. The first definition is presented below and the second is presented in Section 5.2.

The classification margin for sample $\mathbf{x}_{i,g}$ is the following:

$$\mu(\mathbf{x}_{i,g}) = v(\omega_t|\mathbf{x}_{i,g}) - \sum_{k \neq t} v(\omega_k|\mathbf{x}_{i,g}) \tag{10}$$

where ω_t is the true class label of $\mathbf{x}_{i,g}$. Hence, the margin measures the difference between the number of votes assigned for the true class label and the number of votes given for any other class. Consequently, the certainty of the classification is increased by trying to maximize the margin. Based on the standpoints presented in this section, our dynamic level guided by $\bar{\gamma}$, denoted ADS, will pick up $C_j^{*'}$ as the candidate ensemble with lowest $\bar{\gamma}$. The assumption is that the candidate ensemble with the lowest $\bar{\gamma}$ presents the lowest possibility of making a mistake when classifying $\mathbf{x}_{i,g}$.

However, it is important to mention the difference between γ (Eq. (2)) and $\bar{\gamma}$ (Eq. (9)). The former, called global ambiguity in this paper, is used to guide the optimization level, and it is calculated for the whole data set ($\mathcal{O}$ or $\mathcal{V}$). The latter, called local ambiguity, is used in the dynamic selection level calculated for each $\mathbf{x}_{i,g}$. Since the global ambiguity is a dissimilarity measure (Section 4.1), it must be maximized at the optimization level.

5.2. Margin-based dynamic selection (MDS)

The second measure proposed for use in guiding dynamic selection in our approach has been inspired by the second definition of the margin. Following this definition, the margin of sample $\mathbf{x}_{i,g}$ is computed as follows:

$$\mu(\mathbf{x}_{i,g}) = v(\omega_t|\mathbf{x}_{i,g}) - \max_{k \neq t} v(\omega_k|\mathbf{x}_{i,g}) \tag{11}$$

This equation calculates the difference between the number of votes given to the correct class $v(\omega_t|\mathbf{x}_{i,g})$ and the number of votes given to the incorrect class label with the highest number of votes. In our approach, however, $v(\omega_t|\mathbf{x}_{i,g})$ is unknown, since the dynamic selection is performed for test samples. In order to employ this measure to guide the dynamic selection of our DOCS, we have tailored the margin measure defined in Eq. (11) to our problem. Considering $v(mv|\mathbf{x}_{i,g})$ as the number of votes assigned to the majority voting class, we propose to replace $v(\omega_t|\mathbf{x}_{i,g})$ by $v(mv|\mathbf{x}_{i,g})$. In this way, the margin of sample $\mathbf{x}_{i,g}$ for each C_j from the population $\mathbf{C}^{*'}$ may be calculated as follows:

$$\mu(\mathbf{x}_{i,g}) = \frac{v(mv|\mathbf{x}_{i,g}) - \max_{k \neq mv} v(\omega_k|\mathbf{x}_{i,g})}{|C_j|} \tag{12}$$

Hence, our definition of margin measures the difference between the number of votes assigned to the majority voting class and the number of votes assigned to the class with second highest result. Then, the margin value represents the confidence of the classifications, since the higher the margin from Eq. (12), the higher the confidence of the ensemble consensus decision. Thus, the dynamic selection level guided by the margin, denoted MDS, will choose as $C_j^{*'}$ the candidate ensemble with the highest margin. For instance, when $\mu(\mathbf{x}_{i,g}) = 1$ the majority voting matches well to just one class, indicating the highest level of certainty of correct classification.

5.3. Class strength-based dynamic selection (CSDS)

The definition of margin in Eq. (11) also inspired this third confidence measure. Here, however, besides calculating Eq. (12), we also consider the candidate ensemble's confidence with respect to the identity of the majority voting class measured in $\mathcal{V}$. This is additional knowledge related to the performance achieved by each candidate ensemble when assigning the chosen class. Our objective is to investigate whether or not performance may help a confidence measure, which does not take into account the correctness of the output, to increase the candidate ensemble's level of certainty of classification.

Strength relative to the closest class is presented in Ref. [18] as a method for defining weights in DCS-LA. It is calculated for each classifier c_i to verify whether or not the input pattern is closely similar to more than one class. We have adapted this measure to enable us to calculate it for candidate ensembles C_j in the dynamic selection of our DOCS. Assuming $p_j(mv)$ as the performance of C_j measured over samples contained in $\mathcal{V}$ for the majority voting class, strength relative to the closest class may be calculated for $\mathbf{x}_{i,g}$ as follows:

$$\Theta(\mathbf{x}_{i,g}) = \frac{(v(mv|\mathbf{x}_{i,g}) - \max_{k \neq mv} v(\omega_k|\mathbf{x}_{i,g}))/|C_j|}{p_j(mv)} \tag{13}$$

A low value of $\Theta(\mathbf{x}_{i,g})$ means a low level of certainty of correct classification. In contrast, higher $\Theta(\mathbf{x}_{i,g})$ leads to an increase in the level of confidence of classification. Thus, the dynamic selection level guided by Θ, called CSDS, will choose as $C_j^{*'}$ the ensemble with the highest $\Theta(\mathbf{x}_{i,g})$ to classify $\mathbf{x}_{i,g}$.

5.4. Dynamic ensemble selection with local accuracy (DCS-LA)

As explained in Section 2, DCS-LA dynamically selects the most accurate individual classifier from the population $\mathscr{C}$ to predict the label of the test sample $\mathbf{x}_{i,g}$. Local accuracy is measured in the region of competence composed as the set of kNNs from $\mathcal{T}$ surrounding $\mathbf{x}_{i,g}$. Woods et al. [3] compared two strategies to measure local accuracy: (1) overall local accuracy; and (2) local class accuracy. In this paper, we use the second strategy, since they concluded that this is the strategy that achieves the better results.

Given the pool $\mathscr{C}$ and the class assigned by the i-th classifier, ω_y, to the test sample $\mathbf{x}_{i,g}$, we denote N^y as the number of neighbors of $\mathbf{x}_{i,g}$ for which classifier c_i has correctly assigned class ω_y, and $\sum_{i=1}^{k} N^{iy}$ is the total number of neighbors labeled for c_i as class ω_y. According to the definition provided in Ref. [3], the local class accuracy estimation is computed as follows:

$$\alpha_{c_i}(\mathbf{x}_{i,g}) = \frac{N^y}{\sum_{i=1}^{k} N^{iy}} \tag{14}$$

Taking into account that DCS-LA was originally proposed to deal with populations of classifiers, as summarized in Eq. (14), it cannot be directly employed in problems involving populations of classifier ensembles. Thus, we propose to change the DCS-LA with local class accuracy estimation slightly in order to allow it to be used to guide dynamic selection in our proposed approach. Given a population $\mathbf{C}^{*'}$ of candidate ensembles, we assume ω_y as the class assigned by

Table 3
Summary of the four strategies employed at the dynamic selection level

Name	Label	↑ / ↓
Ambiguity-guided dynamic selection	ADS	(↓)
Margin-based dynamic selection	MDS	(↑)
Class strength-based dynamic selection	CSDS	(↑)
Dynamic ensemble selection with local accuracy	DCS-LA	(↑)

The arrows specify whether or not the certainty of the decision is greater if the strategy is lower (↓) or greater (↑).

Table 4
Case study: comparison among the results achieved by combining all classifiers in the initial pool $\mathscr{C}$ and by performing classifier ensemble selection employing both SOCS and DOCS

Optimization	Combination	SOCS	ADS	MDS	CSDS	DCS-LA
NSGA-II (ε & θ)	6.06	5.86	5.74	**5.71**	6.01	5.74
GA (ε)	6.06	5.90	5.90	5.88	6.13	5.98

The lowest error rate is shown in bold.

the candidate ensemble C_j (composed of l classifiers) to the test pattern $\mathbf{x}_{i,g}$. We define as region of competence the set of kNNs from $\mathscr{V}$ surrounding $\mathbf{x}_{i,g}$. Clearly, DCS-LA can be critically affected by the choice of the k parameter. The local candidate ensemble's class accuracy is then estimated as follows:

$$\alpha_{C_j}(\mathbf{x}_{i,g}) = \frac{\sum_{i=1}^{l} \alpha_{c_i}(\mathbf{x}_{i,g})}{|C_j|} \tag{15}$$

To summarize, in this paper, α_{C_j} for pattern $\mathbf{x}_{i,g}$ is calculated as the sum of the local class accuracy (α_{c_i}) of each classifier composing C_j, divided by the size of C_j. The higher α_{C_j}, the greater the certainty of the decision. Table 3 shows a summary of the four different strategies proposed for use at the dynamic selection level of our DOCS.

Using the case study mentioned in Section 4, we compare, in Table 4, the results obtained in $\mathscr{G}$ using the combination of the initial pool $\mathscr{C}$, SOCS and our DOCS employing the four dynamic strategies presented in this section. These preliminary results show that, except for CSDS, our dynamic method guided by confidence

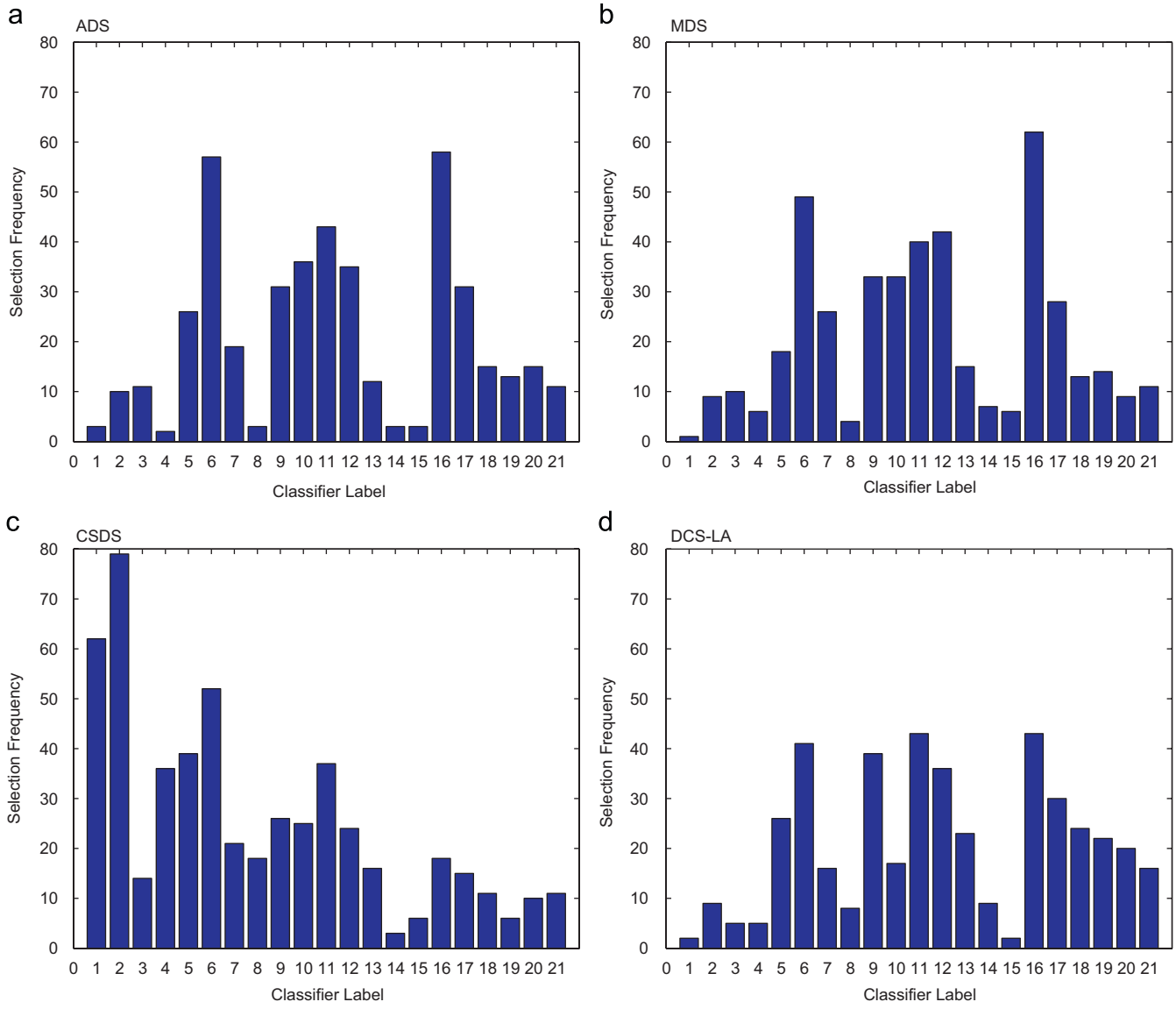

Fig. 4. Case study: histogram of the frequency of selection of candidate ensembles performed by each dynamic selection strategy. The population $\mathbf{C}^{*'}$ is the n best solution generated by GA (see Fig. 3(c)).

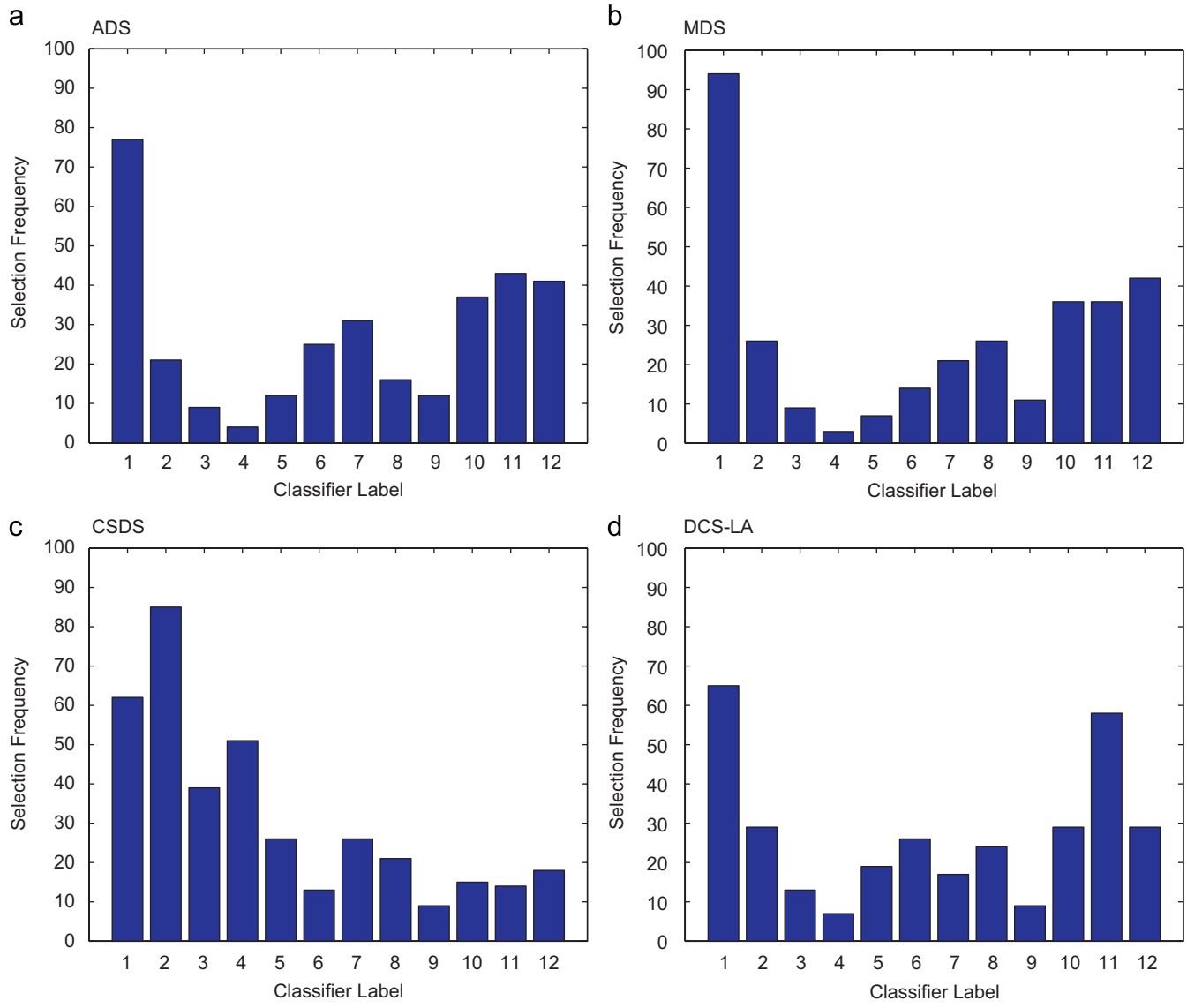

Fig. 5. Case study: histogram of the frequency of selection of candidate ensembles performed by each dynamic selection method. The population $\mathbf{C}^{*'}$ is the Pareto front generated by NSGA-II (see Fig. 3(d)).

Table 5
Specifications of the large data sets used in the experiments

Data set	# of classes	Train set $\mathscr{T}$	Optimization set $\mathscr{O}$	Validation set $\mathscr{V}$	Test set $\mathscr{G}$	Features RSS	Pool $\mathscr{C}$ size
NIST-digits	10	5000	10,000	10,000	Test 1 60,089 Test 2 58,646	32	100
NIST-letters	26	43,160	3980	7960	12,092	32	100

measures outperformed SOCS performed by NSGA-II. In terms of the single-objective GA, static and dynamic selection methods presented similar results, except for CSDS and DCS-LA.

Figs. 4 and 5 show histograms of the frequency of selection of each candidate ensemble performed using each dynamic selection strategy presented in this section. The histograms in Fig. 4 were obtained for the population $\mathbf{C}^{*'}$ composed of n best candidate ensembles generated by GA shown in Fig. 3(c), while the histograms in Fig. 5 were obtained considering the Pareto front determined by II shown in Fig. 3(d). Especially noteworthy is the fact that the test set $\mathscr{G}$ used for the case study is composed of 12,092 samples (Table 5), even though the dynamic selection level was conducted over only 325 and 436 of test samples for MOGA and GA respectively. All candidate ensembles in $\mathbf{C}^{*'}$ agreed on the label for the remaining test samples.

It is also important to observe in Fig. 5 that ADS, MDS and DCS-LA more often selected as $C_j^{*'}$ the same candidate ensemble selected statically (C_1 in Fig. 3(d)). In contrast, the opposite behavior is shown in Fig. 4. These results indicate why DOCS did not outperform SOCS for the GA's population of candidate ensembles (see Table 4). In addition, two of the confidence-based dynamic strategies, namely ADS and MDS, more frequently selected the same candidate ensembles as

selected by DCS-LA, which is an accuracy-oriented strategy. These results support our assumption that selecting the candidate ensemble with the largest consensus improves the performance of the system. Moreover, considering the results obtained by CSDS, we can conclude that measures of confidence that do not take into account the correctness of the output provide enough information about a candidate ensemble's level of certainty of classification. The additional information calculated by CSDS through measuring the performance of each candidate ensemble over samples in $\mathscr{V}$ did not help in finding better performing ensembles. In next section, we present experimental results to verify whether or not these preliminary results are general, considering other databases and ensemble generation methods.

6. Experiments

A series of experiments has been carried out to determine the best strategy for the dynamic selection level proposed in our approach and to show whether or not DOCS is better than SOCS. As a consequence, we also point out the best method for the overproduction phase on comparing RSS and BAG, and the best search algorithm and search criteria for the optimization level. We used seven data sets divided into two groups: (1) two large; and (2) five small, Tables 5 and 6, respectively. Taking into account that the data sets from group 1 are large enough to be partitioned into the four independent data sets, illustrated in Fig. 2: $\mathscr{T}$, $\mathscr{O}$, $\mathscr{V}$ and $\mathscr{G}$, the classical holdout validation strategy is employed for the evaluation of performance using large data sets. By contrast, 10-fold cross-validation is applied for the evaluation of performance using small data sets due to the small number of samples available for evaluation.

The NIST digits Special Database 19 (NIST SD19), called NIST-digits here, and the NIST SD19 handwritten uppercase letters [32], called NIST-letters here, are the two large data sets used. We employ the representation proposed by Oliveira et al. [33], which is a combination of concavity, contour and surface of characters. The final feature vector is composed of 132 components: 78 for concavity, 48 for contour and 6 for surface. Table 5 lists important information about the databases and the partitions used to compose the four separate sets. There are two test sets from NIST-digits, called Test 1 (60,089 samples) and Test 2 (58,646 samples). Test 2 is well known to be more difficult to use for classification than Test 1 [32].

Table 6 describes the five small data sets: the *Dna* and *satimage* data sets are provided by Project Statlog on www.niaad.liacc.up.pt/old/statlog; *feltwell* is a multisensor remote-sensing data set [34]; *ship* is a data set composed of forward-looking infra-red (FLIR) ship images [35]; and *texture* is available within the UCI machine Learning Repository. The original data sets were divided into 10 folds. Each time, one of the 10 folds was used as $\mathscr{G}$, another fold as $\mathscr{V}$, a third as $\mathscr{O}$ and the other 7 were put together to form $\mathscr{T}$. This process was repeated 10 times, i.e. our whole method summarized in Algorithm 1 was repeated for 10 trials. It is also important to mention that the selection phase was replicated 30 times for each trial owing to the use of stochastic algorithms. Thus, the mean of the error rates over 30 replications for each trial were computed and the error rates reported in all tables of results were obtained as the mean of the error rates across all 10 trials. The selection phase performed for large data sets was also replicated 30 times.

Table 7
Genetic algorithms parameters

Population size	128
Number of generations	1000
Probability of crossover	0.8
Probability of mutation	0.01

One-point crossover and bit-flip mutation.

We chose DT and kNN as the base classifiers in our experiments. The C4.5 algorithm [36] (Release 8) was used to construct the trees with pruning. We used $k=1$ for kNN classifiers without fine-tuning this parameter in order to avoid additional experiments. Because BAG is mostly effective with unstable classifiers, it was employed to generate only a pool of 100 DT, while RSS was used to generate two pools of 100 homogeneous classifiers: 100 DT and 100 kNN. The size of the subsets of features used by RSS is shown in Table 5 for large data sets and in Table 6 for small data sets. The same subspaces were used for both kNN and DT classifiers.

The optimization level was applied in the context of GAs based on binary vectors. Since we used initial pools of classifiers composed of 100 members, each individual is represented by a binary vector with a size of 100. Each bit determines whether a classifier is active (1) or not (0). We defined the same genetic parameters employed in Ref. [11] and summarized in Table 7. The same parameters were used for the single-objective GA and NSGA-II. As mentioned in Section 4.2, we pick up all solutions over the Pareto front to constitute the input of the dynamic selection level when NSGA-II is used at the optimization level. When the single-objective GA is employed, the n best solutions should be picked up as the optimization level output. We set $n=21$ in our experiments in an attempt to have a number of candidate ensembles close to the number of candidate ensembles over the Pareto front found by NSGA-II in order to better compare the results obtained by both GAs.

To summarize, for the overproduction phase, three different initial pools of 100 homogeneous classifiers were created: 100 kNN and 100 DT generated by RSS, and 100 DT generated by BAG. The selection phase was divided into two levels: for the optimization level, each diversity measure presented in Section 3.2 was employed in combination with ε to guide NSGA-II, while only ε was used to guide GA. For the dynamic selection level, all four dynamic strategies defined in Section 5 were tested so that they could be compared. We set $k=10$ for experiments with DCS-LA in this paper, as employed in Refs. [26,20]. The results obtained are given in subsequent sections.

6.1. Comparison of dynamic selection strategies

A summary of the experimental results comparing the four dynamic selection strategies defined in Section 5 is given in Tables A1–A3 in the appendix. The best result for each data set is

Table 6
Specifications of the small data sets used in the experiments

Data set	Number of samples	Number of classes	Number of features	Features RSS	Pool $\mathscr{C}$ size
Dna	3186	3	180	45	100
Feltwell	10,944	5	15	8	100
Satimage	6435	6	36	18	100
Ship	2545	8	11	6	100
Texture	5500	11	40	20	100

Table 8
Error rates attained by several methods

Method	Dna	Feltwell	NIST-digits		NIST-letters	Satimage	Ship	Texture
			Test 1	Test 2				
Bagging DT	X-I θ	Y-I	Y-II	Y-II	X-II θ	X-III γ	Y-I	X-II θ
Initial pool	5.02	12.80	5.65	10.99	7.63	9.59	8.09	3.60
Best classifier	5.87	**10.59**	9.70	16.62	14.31	12.77	9.08	6.64
Oracle of $\mathscr{C}$	0.38	2.57	0.24	0.63	0.29	0.11	0.35	0.02
Best DOCS	**4.75**	11.69	**5.14**	**10.06**	**7.50**	**9.21**	**7.72**	**3.41**
RSS DT	X-II θ	Y-II	Y-II θ	X-II θ	X-II θ	X-III θ	Y-III	X-II θ
Initial pool	5.05	11.86	2.92	6.67	6.06	8.64	**6.80**	2.56
Best classifier	11.33	11.86	11.07	19.18	17.13	11.83	10.45	6.07
DT all features	6.85	16.81	10.30	18.20	13.5	14.17	10.92	7.56
Oracle of $\mathscr{C}$	0.03	0.60	0.01	0.04	0.04	0.22	0.24	0.02
Best DOCS	**4.59**	**11.50**	**2.77**	**6.45**	**5.84**	**8.63**	6.95	**2.35**
RSS kNN	Y-II	X-I θ	Y-II	Y-II	X-III θ	Y-II	X-II δ	X-II θ
Initial pool	**6.87**	10.44	3.72	8.10	6.60	**8.59**	9.94	1.11
Best classifier	23.10	**9.46**	7.52	13.99	14.47	8.95	10.26	**0.62**
kNN all features	26.30	12.35	6.66	9.76	7.82	9.84	11.24	1.13
Oracle of $\mathscr{C}$	0.03	0.67	0.05	0.17	0.18	0.36	0.28	0.04
Best DOCS	7.24	9.70	**3.53**	**7.78**	**6.27**	8.61	**9.13**	0.94

NSGA-II (X), GA (Y), I (DCS-LA), II (ADS), III (MDS), IV (CSDS). Values in bold and underlined indicate the best result in each data set for each overproduction method and the best overall result for each data set, respectively.

Table 9
The error rates obtained, the data partition and the selection method employed in works which used the databases investigated in this paper (FSS: feature subset selection)

Database	Reference number	Ensemble creation	Classifiers members	Partition strategy	Selection type	Error (%)
Dna	[6]	RSS	DT	Holdout	Fusion	9.19
Feltwell	[37]	Het	Het	Cross-validation	DCS-LA	13.62
NIST-digits Test 1	[30]	RSS	kNN	Holdout	SOCS	3.65
NIST-letters	[38]	FSS	MLP	Holdout	SOCS	4.02
Satimage	[37]	Het	Het	Cross-validation	DCS-LA	10.82
Ship	[39]	Het	Het	Holdout	Fusion	5.68
Texture	[3]	DT	Het	Holdout	DCS-LA	0.75

Table 10
Mean and standard deviation values of the error rates obtained on 30 replications comparing DOCS and SOCS with no rejection

Data set	SOCS		DOCS	
	NSGA-II (θ & ε)	GA ε	NSGA-II (θ & ε)	GA ε
Dna	4.77 (0.17)	4.97 (0.16)	**4.59** (0.17)	4.95 (0.19)
Feltwell	12.02 (0.38)	11.86 (0.06)	11.65 (0.33)	**11.50** (0.17)
NIST-digits Test 1	2.82 (0.06)	2.84 (0.06)	**2.77** (0.03)	**2.77** (0.03)
NIST-digits Test 2	6.59 (0.09)	6.53 (0.09)	**6.45** (0.05)	**6.45** (0.05)
NIST-letters	5.89 (0.07)	6.02 (0.09)	**5.84** (0.06)	5.96 (0.06)
Satimage	8.76 (0.13)	8.82 (0.12)	**8.64** (0.10)	8.78 (0.12)
Ship	7.35 (0.17)	7.14 (0.23)	7.17 (0.15)	**6.98** (0.14)
Texture	2.41 (0.09)	2.51 (0.09)	**2.35** (0.06)	2.44 (0.05)

Values in bold indicate the lowest error rate and underlined when a method is significantly better than the others.

shown in bold. These results indicate that ADS and MDS were the best strategies for performing the dynamic selection level of our approach, considering all three ensemble creation methods investigated, i.e. (1) ensembles of RSS; (2) kNN generated through RSS (Tables A1 and A2); and (3) ensembles of DT generated by BAG (Table A3). These two dynamic strategies presented equivalent performances, which confirms the preliminary results in our case-study problem (see Table 4). CSDS was the worst dynamic selection strategy for 1 and 3, while DCS-LA was most likely to be the worst dynamic selection strategy for ensembles of DT generated by RSS. In terms of the optimization level, single-objective GA and NSGA-II presented equivalent performances for ensembles of kNN generated by RSS and ensembles of DT generated by BAG, while NSGA-II found the best results for populations of ensembles of DT generated by RSS. θ was clearly the best diversity measure for composing, with ε, a pair of objective functions to guide NSGA-II, while γ and σ were the worst diversity measures.

It is important to note that DCS-LA was better than the other dynamic selection strategies in three of the eight data sets in problems involving DT generated by BAG. The reason for this behavior is that BAG provides the complete representation of the problem to each classifier member, whereas RSS provides only a partial representation. Thus, DCS-LA calculates the local accuracy of each classifier more accurately when BAG is used as the ensemble creation method. Also important is the fact that CSDS was less effective in

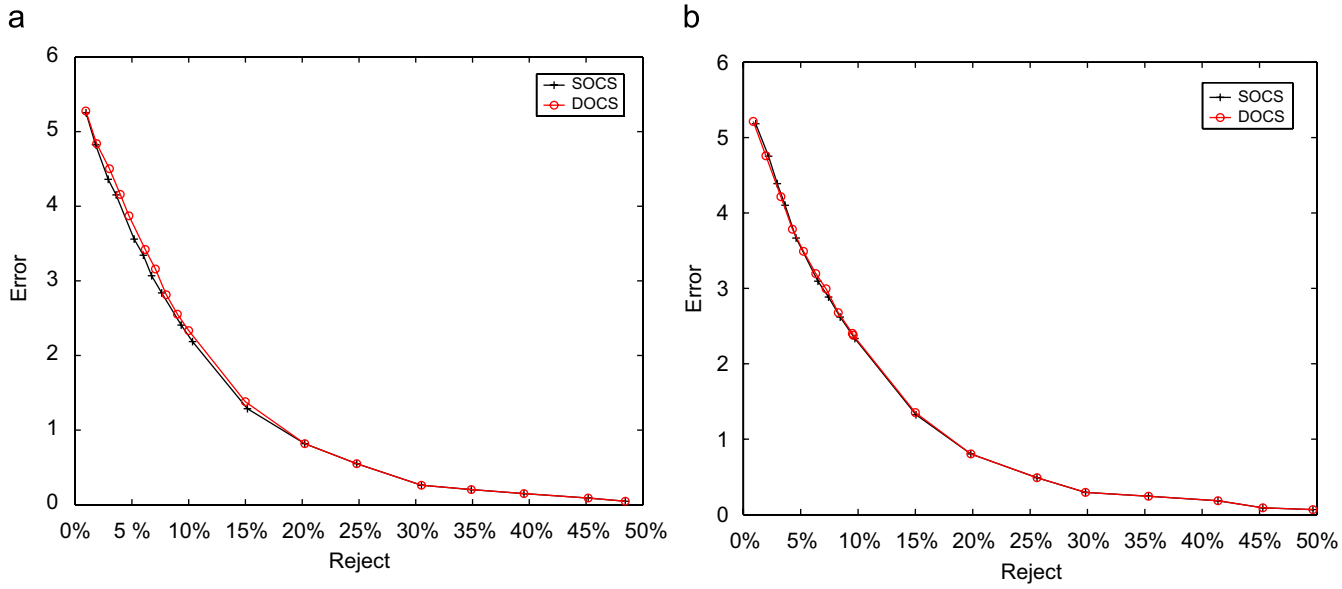

Fig. 6. Case study: error-reject curves for GA in (a) and NSGA-II in (b).

problems involving a large number of classes, such as *NIST-letters*, because it takes into account the performance of the candidate ensembles for each class involved in the classification problem, in addition to the extent of consensus of the ensembles. Moreover, for the same reason, CSDS is much more critically affected by the quality of the population of candidate ensembles found at the optimization level. For instance, since γ and σ were the worst objective functions when guiding the optimization level, CSDS was much worse than the other three strategies when used to perform the dynamic selection on populations of candidate ensembles found by these two diversity measures.

6.2. Comparison between DOCS and several methods

In this section, we summarize the best results obtained by our DOCS for each data set, in order to show that the proposed approach outperforms other related methods. Table 8 reports the results attained by the following methods:

- Fusion of the initial pool of classifiers $\mathscr{C}$ by majority voting.
- Selection of the best individual classifier from the initial pool $\mathscr{C}$.
- Individual kNNs and DTs trained using all available features.
- DOCS using the best dynamic selection strategy.
- Oracle for each initial pool of candidate classifiers.

Oracle is an upper bound of selection strategies [37], because it correctly classifies the test sample if any of the classifier members predicts the correct label for the sample. Values are shown in bold for the best result obtained for each data set considering each ensemble generation method, and are shown underlined for the best overall result obtained for each data set, whatever the ensemble creation method used. From this table, we observe the following:

- Analyzing the results obtained for ensembles of DT generated using BAG, our DOCS outperformed the other methods, except for *feltwell*. The same scenario was observed for ensembles of DT generated using RSS, but the exception was the *ship* data set. For ensembles of kNN generated using RSS, however, the proposed method outperformed the other methods in only four of the eight cases. For the remaining four data sets, our DOCS was the second best method. The combination of the initial pool $\mathscr{C}$ was the best method for *dna* and *satimage* while the individual best classifier from $\mathscr{C}$ was the best method for *feltwell* and *texture*.
- Confidence-based dynamic strategies were better than DCS-LA for performing the dynamic selection of the selection phase. Even though ADS and MDS presented similar behavior, we show in Table 8 values obtained using ADS to perform the comparison in next section.
- The search algorithms presented equivalent performances. The best results found using DOCS were obtained in populations of candidate ensembles optimized by NSGA-II in 13 cases and by single-objective GA in 11 cases out of a total of 24 cases investigated.
- θ was the best diversity measure for combining with ε to guide NSGA-II.
- RSS was better than BAG to for use during the overproduction phase. Results obtained using ensembles generated by RSS were both better than those obtained using ensembles generated by BAG.
- DT was the best classifier model to be used as a base classifier during the overproduction phase. Five of the best overall results were obtained with ensembles of DT, whereas in the remaining three data sets they were obtained with ensembles of kNN. Both ensembles were generated by RSS.

Table 11
Case study: comparing oracle results

Fusion of initial pool $\mathscr{C}$	Oracle of initial pool $\mathscr{C}$	SOCS	DOCS	Oracle of population of ensembles $\mathbf{C}^{*'}$
6.06	0.04	5.86	5.71	4.81

Table 9 presents some of the results reported in the literature dealing with the selection of classifiers, except for Refs. [6,39], on the databases used in this paper. In this way, it is possible to gain an overview of the results obtained in the literature, even though the method of partition of the data used in some of these works was not the same as that used in this paper.

To verify whether or not our DOCS is better that the classical SOCS, in the next section we concentrate our analysis on the methods that

Table A1
Mean and standard deviation values obtained on 30 replications of the selection phase of our method

Data set	Method	NSGA-II				GA
		Ambiguity (γ)	Coincident (σ)	Difficulty (θ)	Double-fault (δ)	Error (ε)
DNA	DCS-LA	9.33 (0.27)	9.84 (0.28)	7.48 (0.24)	9.36 (0.30)	7.47 (0.15)
	ADS	10.31 (0.28)	9.72 (0.47)	7.51 (0.22)	9.30 (0.23)	7.24 (0.21)
	MDS	10.34 (0.30)	9.57 (0.41)	7.47 (0.21)	9.16 (0.28)	**7.24** (0.15)
	CSDS	10.56 (0.30)	9.89 (0.19)	7.48 (0.19)	8.36 (0.30)	7.25 (0.17)
Feltwell	DCS-LA	11.17 (0.21)	10.46 (0.25)	**9.70** (0.14)	10.59 (0.31)	9.87 (0.13)
	ADS	10.51 (0.11)	10.34 (0.25)	9.82 (0.16)	10.62 (0.28)	9.95 (0.17)
	MDS	10.51 (0.11)	10.34 (0.26)	9.82 (0.14)	10.59 (0.29)	9.95 (0.16)
	CSDS	10.57 (0.10)	10.40 (0.25)	9.84 (0.14)	10.66 (0.32)	9.97 (0.17)
Nist-digits Test 1	DCS-LA	4.12 (0.07)	4.59 (0.17)	3.72 (0.05)	4.43 (0.26)	3.60 (0.06)
	ADS	3.80 (0.04)	3.95 (0.05)	3.58 (0.03)	7.49 (2.29)	3.53 (0.05)
	MDS	3.80 (0.04)	3.94 (0.05)	3.58 (0.02)	7.46 (2.32)	**3.53** (0.05)
	CSDS	5.21 (0.09)	7.25 (0.26)	3.80 (0.04)	6.67 (0.22)	3.80 (0.06)
Nist-digits Test 2	DCS-LA	8.57 (0.14)	9.34 (0.28)	8.18 (0.11)	9.48 (0.43)	7.91 (0.10)
	ADS	8.11 (0.06)	8.55 (0.08)	7.97 (0.05)	8.40 (0.48)	**7.78** (0.10)
	MDS	8.12 (0.07)	8.53 (0.09)	7.97 (0.05)	8.36 (0.55)	7.78 (0.10)
	CSDS	9.95 (0.11)	9.89 (0.33)	8.21 (0.08)	9.42 (0.45)	8.09 (0.10)
Nist-letters	DCS-LA	7.63 (0.43)	8.22 (0.69)	6.48 (0.15)	7.20 (0.33)	6.55 (0.16)
	ADS	7.23 (0.17)	8.22 (0.12)	6.48 (0.07)	6.95 (0.12)	6.43 (0.06)
	MDS	7.16 (0.16)	7.07 (0.11)	**6.27** (0.07)	6.88 (0.12)	6.41 (0.08)
	CSDS	10.15 (0.45)	14.53 (0.63)	6.47 (0.14)	9.35 (0.41)	6.70 (0.09)
Satimage	DCS-LA	8.75 (0.08)	9.60 (0.15)	8.73 (0.13)	9.40 (0.13)	8.63 (0.07)
	ADS	8.67 (0.08)	9.05 (0.17)	8.64 (0.13)	9.16 (0.16)	**8.61** (0.09)
	MDS	8.68 (0.09)	9.04 (0.18)	8.65 (0.14)	9.16 (0.18)	8.61 (0.09)
	CSDS	8.96 (0.13)	10.13 (0.24)	8.80 (0.16)	9.78 (0.19)	8.67 (0.11)
Ship	DCS-LA	14.24 (0.32)	10.31 (0.29)	9.34 (0.24)	9.24 (0.23)	10.40 (0.20)
	ADS	13.25 (0.24)	9.60 (0.33)	9.25 (0.20)	**9.13** (0.21)	9.81 (0.16)
	MDS	13.37 (0.26)	9.66 (0.31)	9.24 (0.20)	9.15 (0.22)	9.83 (0.13)
	CSDS	14.06 (0.25)	10.17 (0.34)	9.39 (0.21)	9.31 (0.29)	10.23 (0.18)
Texture	DCS-LA	1.51 (0.06)	1.64 (0.08)	0.97 (0.05)	1.02 (0.07)	1.18 (0.03)
	ADS	1.37 (0.05)	1.23 (0.09)	**0.94** (0.05)	0.98 (0.07)	1.11 (0.01)
	MDS	1.37 (0.05)	1.22 (0.09)	0.94 (0.05)	0.98 (0.07)	1.11 (0.02)
	CSDS	1.37 (0.07)	1.28 (0.10)	0.94 (0.06)	0.98 (0.09)	1.11 (0.03)

The overproduction phase was performed using an initial pool of kNN classifiers generated by RSS. The best result for each data set is shown in bold.

attained the best results in this section, i.e. DT ensembles generated by RSS, ADS as dynamic strategy; MOGA guided by θ and ε; and GA guided by ε.

6.3. Comparison of DOCS and SOCS results

The static selection results were obtained by picking up the candidate ensemble presenting the lowest ε value of all the solutions composing the n best solution (for single-objective GA) or the Pareto front (for NSGA-II). These ensembles are represented in Figs. 3(c) and (d) by black circles. It is important to mention that the results were tested on multiple comparisons using the Kruskal–Wallis non-parametric statistical test by testing the equality between mean values. The confidence level was 95% ($\alpha = 0.05$), and the Dunn–Sidak correction was applied to the critical values.

Table 10 summarizes the mean and the standard deviation of the error rates obtained on all eight databases comparing the static and dynamic OCS. These results indicate, without exception, that our DOCS was better than the traditional SOCS. In two specific situations (*feltwell* and *NIST-letters*), the differences between the two methods were not significant. It is important to note that the results achieved using NSGA-II guided by both θ and ε outperformed the results found using GA guided by ε the single-objective function. This is a different result from those presented in Refs. [11,30]. It appears that performing dynamic selection in the selection phase resulted in exploiting all possible potentials of the population of candidate ensembles over the Pareto front, leading to more benefits for NSGA-II with the new level. However, only with the *texture* database was GA better than NSGA-II in SOCS, whereas in DOCS the opposite is true.

All these results were obtained with a zero reject rate. However, as advocated by Hansen et al. [17], the reject mechanism for an ensemble of classifiers is based on the extent of consensus among its members. This means that the decision to reject a pattern is related to the confidence level of the ensemble. They assume that it is better to reject the pattern if the ensemble presents a low confidence level to take a decision. Such a confidence level is clearly related to $\bar{\gamma}$ (Eq. (9)), which is used to guide ADS. Thus, since $\bar{\gamma}$ is the criterion used to perform the reject mechanism, we might assume that the difference between SOCS and DOCS will increase as the rejection rate increases. We analyze such an assumption in Fig. 6 taking into account the case study investigated throughout this paper. This preliminary result does not confirm this assumption. Both the static and the dynamic OCS presented similar error-reject curves for ensembles generated by GA 6(a) and NSGA-II 6(b). The results obtained by analyzing the error-reject curves calculated for all data sets investigated in this paper confirm that we should not assume that DOCS is more effective than SOCS when increasing the rejection rate.

7. Conclusion and discussion

We propose a dynamic overproduce-and-choose strategy which is composed of the traditional overproduction and selection phases. The novelty is to divide the selection phase into two levels: optimization and dynamic selection, conducting the second level using

Table A2
Mean and standard deviation values obtained on 30 replications of the selection phase of our method

Data set	Method	NSGA-II				GA
		Ambiguity (γ)	Coincident (σ)	Difficulty (θ)	Double-fault (δ)	Error (ε)
DNA	DCS-LA	7.05 (0.21)	7.54 (0.28)	4.63 (0.15)	7.23 (0.23)	5.14 (0.14)
	ADS	7.94 (0.23)	6.50 (0.29)	**4.59** (0.17)	6.27 (0.23)	4.95 (0.18)
	MDS	7.92 (0.23)	6.54 (0.30)	4.63 (0.17)	6.27 (0.19)	4.92 (0.19)
	CSDS	8.01 (0.24)	6.59 (0.28)	4.62 (0.17)	6.34 (0.22)	4.93 (0.19)
Feltwell	DCS-LA	12.68 (0.19)	13.27 (0.47)	11.65 (0.32)	12.79 (0.51)	11.59 (0.14)
	ADS	11.60 (0.15)	12.71 (0.40)	11.65 (0.33)	12.51 (0.51)	**11.50** (0.17)
	MDS	11.59 (0.14)	12.74 (0.42)	11.65 (0.32)	12.49 (0.48)	11.53 (0.16)
	CSDS	11.66 (0.15)	12.80 (0.43)	11.65 (0.32)	12.51 (0.50)	11.51 (0.16)
Nist-digits Test 1	DCS-LA	3.82 (0.12)	5.37 (0.29)	2.91 (0.04)	3.68 (0.20)	2.89 (0.04)
	ADS	3.35 (0.04)	4.11 (0.20)	**2.77** (0.03)	3.59 (0.23)	**2.77** (0.09)
	MDS	3.35 (0.04)	3.83 (0.13)	2.77 (0.02)	3.40 (0.12)	2.77 (0.04)
	CSDS	5.63 (0.09)	5.53 (0.12)	2.98 (0.06)	4.47 (0.78)	3.12 (0.06)
Nist-digits Test 2	DCS-LA	8.18 (0.18)	10.06 (0.50)	6.79 (0.09)	8.07 (0.37)	6.66 (0.08)
	ADS	7.38 (0.07)	9.16 (0.32)	**6.45** (0.05)	8.11 (0.40)	**6.45** (0.05)
	MDS	7.34 (0.06)	8.61 (0.20)	6.50 (0.05)	7.80 (0.20)	6.45 (0.05)
	CSDS	9.05 (0.14)	8.77 (0.54)	6.50 (0.09)	8.69 (0.50)	6.98 (0.10)
Nist-letters	DCS-LA	7.57 (0.30)	6.12 (0.50)	6.03 (0.14)	7.20 (0.39)	6.17 (0.14)
	ADS	7.13 (0.09)	9.31 (0.29)	**5.84** (0.06)	7.12 (0.14)	5.96 (0.06)
	MDS	7.12 (0.09)	7.69 (0.19)	5.84 (0.06)	6.93 (0.10)	5.95 (0.06)
	CSDS	10.65 (0.11)	15.92 (0.14)	5.98 (0.08)	10.95 (0.46)	6.29 (0.09)
Satimage	DCS-LA	9.30 (0.09)	10.66 (0.19)	8.97 (0.09)	10.12 (0.15)	8.96 (0.09)
	ADS	8.73 (0.10)	9.22 (0.19)	8.64 (0.10)	9.17 (0.15)	8.78 (0.12)
	MDS	8.73 (0.10)	9.22 (0.19)	**8.63** (0.09)	9.16 (0.15)	8.77 (0.13)
	CSDS	9.13 (0.13)	11.23 (0.24)	8.94 (0.12)	10.64 (0.26)	8.86 (0.12)
Ship	DCS-LA	10.65 (0.30)	9.16 (0.30)	7.53 (0.18)	7.93 (0.29)	7.24 (0.14)
	ADS	8.86 (0.25)	8.26 (0.34)	7.17 (0.15)	7.72 (0.28)	6.98 (0.14)
	MDS	8.94 (0.26)	8.26 (0.32)	7.18 (0.15)	7.75 (0.29)	**6.95** (0.13)
	CSDS	9.97 (0.24)	9.15 (0.41)	7.18 (0.17)	8.03 (0.27)	7.35 (0.12)
Texture	DCS-LA	3.42 (0.17)	3.85 (0.14)	2.47 (0.07)	3.51 (0.16)	2.84 (0.09)
	ADS	2.98 (0.08)	3.02 (0.14)	**2.35** (0.06)	2.97 (0.14)	2.44 (0.05)
	MDS	3.00 (0.08)	3.04 (0.14)	2.35 (0.05)	2.65 (0.16)	2.44 (0.05)
	CSDS	3.01 (0.08)	3.06 (0.15)	2.35 (0.05)	2.96 (0.16)	2.44 (0.06)

The overproduction phase was performed using an initial pool of DT classifiers generated by RSS. The best result for each data set is shown in bold.

confidence measures based on the extent of consensus of ensembles. Two classical ensemble creation methods, the random subspace method and bagging, were employed at the overproduction phase, while single- and multi-objective GAs were used at the optimization level. The ensemble's error rates and diversity measures guided the optimization. Finally, three confidence measures were applied at the dynamic selection level: (1) ambiguity; (2) margin; and (3) strength relative to the closest class. In addition, DCS-LA was compared to the confidence-based strategies.

Experiments conducted using eight data sets demonstrated that the proposed approach outperforms both static selection and the fusion of the initial pool of classifiers. Ambiguity and margin were the best measures to use at the dynamic selection level, presenting equivalent performances. It was shown that NSGA-II guided by the difficulty measure combined with the error rate found better ensembles than single-objective GA guided only by the error rate. In addition, although not as clearly as one might have hoped, our results indicate that our method is especially valuable for tasks using NSGA-II, since differences between the results found using NSGA-II and GA are greater in the dynamic than in the static overproduce-and-choose strategy. Ideally, we should exploit all the potential of the population of candidate ensembles over a Pareto front using the proposed dynamic selection level.

However, the quality of the population of candidate ensembles found at the optimization level critically affects the performance of our approach. We can confirm this observation taking into account the concept of oracle. As mentioned before, the so-called oracle is an upper bound of selection strategies. Table 11 shows results attained by the fusion of the initial pool of classifiers $\mathscr{C}$ and its oracle for the same case study investigated throughout this paper. Only NSGA-II guided by θ combined with ε is considered in this example.

It is important to mention that, instead of $\mathscr{C}$, the Pareto front (population of ensembles $\mathbf{C}^{*'}$) is the input to our dynamic selection level. Hence, we should consider the oracle of the population $\mathbf{C}^{*'}$ as the upper bound for our DOCS. Thus, it is assumed that oracle correctly classifies the test sample if any of candidate ensembles in $\mathbf{C}^{*'}$ predicts the correct label for the sample. In Table 11, it is shown that our DOCS may not achieve an error rate lower than 4.81. The large difference between the results of the $\mathscr{C}$ and $\mathbf{C}^{*'}$ oracles leads us to conclude that there is a major loss of oracle power at the optimization level. Populations of candidate ensembles with oracle error rates closer to those obtained using $\mathscr{C}$ will bring about an effective improvement in the performances attained by our method. The next stage of this research will involve determining strategies to improve the quality of the population of ensembles.

Acknowledgments

This research was supported by CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior), Brazil, and Defence Research and Development Canada, DRDC-Valcartier under the contract W7701-2-4425.

Appendix A. Tables of results on comparing dynamic selection strategies

Tables A1–A3 show the summary of the experimental results comparing the four dynamic selection strategies defined in Section 5.

Table A3
Mean and standard deviation values obtained on 30 replications of the selection phase of our method

Data set	Method	NSGA-II				GA
		Ambiguity (γ)	Coincident (σ)	Difficulty (θ)	Double-fault (δ)	Error (ε)
DNA	DCS-LA	5.15 (0.15)	5.20 (0.18)	**4.75** (0.15)	5.30 (0.15)	4.88 (0.09)
	ADS	5.20 (0.16)	5.20 (0.16)	4.86 (0.11)	5.32 (0.21)	4.95 (0.06)
	MDS	5.17 (0.17)	5.20 (0.18)	4.86 (0.10)	5.31 (0.23)	4.95 (0.06)
	CSDS	5.19 (0.15)	5.22 (0.17)	4.86 (0.11)	5.36 (0.25)	4.97 (0.07)
Feltwell	DCS-LA	12.50 (0.36)	12.47 (0.28)	12.00 (0.22)	12.40 (0.32)	**11.69** (0.20)
	ADS	12.70 (0.21)	12.28 (0.34)	12.04 (0.18)	12.24 (0.31)	11.95 (0.14)
	MDS	12.70 (0.17)	12.28 (0.35)	12.03 (0.18)	12.24 (0.29)	11.94 (0.14)
	CSDS	12.75 (0.19)	12.38 (0.33)	12.04 (0.19)	12.34 (0.32)	11.98 (0.15)
Nist-digits Test 1	DCS-LA	5.74 (0.09)	6.72 (0.36)	5.67 (0.09)	6.04 (0.21)	5.23 (0.06)
	ADS	5.53 (0.05)	5.79 (0.11)	5.51 (0.05)	5.47 (0.10)	**5.14** (0.06)
	MDS	5.52 (0.05)	5.69 (0.10)	5.51 (0.05)	5.44 (0.09)	5.14 (0.05)
	CSDS	5.87 (0.12)	6.86 (0.56)	5.64 (0.09)	6.72 (0.41)	5.44 (0.06)
Nist-digits Test 2	DCS-LA	11.00 (0.16)	11.60 (0.55)	10.98 (0.15)	10.64 (0.37)	10.21 (0.12)
	ADS	10.71 (0.09)	10.81 (0.25)	10.71 (0.09)	10.59 (0.18)	**10.06** (0.11)
	MDS	10.70 (0.09)	11.02 (0.24)	10.70 (0.10)	10.54 (0.12)	10.06 (0.10)
	CSDS	11.18 (0.20)	12.76 (0.91)	10.88 (0.12)	11.68 (0.61)	10.41 (0.11)
Nist-letters	DCS-LA	7.81 (0.17)	9.13 (0.62)	7.64 (0.12)	8.45 (0.44)	7.79 (0.12)
	ADS	7.61 (0.06)	8.36 (0.15)	**7.50** (0.06)	8.06 (0.17)	7.64 (0.07)
	MDS	7.63 (0.06)	8.30 (0.10)	7.50 (0.05)	8.02 (0.14)	7.64 (0.07)
	CSDS	7.85 (0.07)	10.16 (0.94)	7.60 (0.08)	9.84 (0.54)	7.78 (0.01)
Satimage	DCS-LA	9.70 (0.38)	11.35 (0.57)	9.81 (0.11)	10.99 (0.22)	9.74 (0.09)
	ADS	9.34 (0.43)	10.36 (0.60)	9.62 (0.09)	10.25 (0.13)	9.61 (0.11)
	MDS	**9.21** (0.37)	10.44 (0.58)	9.63 (0.07)	10.22 (0.11)	9.61 (0.12)
	CSDS	9.78 (0.17)	12.08 (0.22)	9.78 (0.11)	11.82 (0.28)	9.68 (0.12)
Ship	DCS-LA	7.79 (0.19)	8.32 (0.29)	7.81 (0.29)	8.58 (0.29)	**7.72** (0.13)
	ADS	8.14 (0.20)	8.70 (0.29)	8.36 (0.19)	8.83 (0.27)	8.07 (0.09)
	MDS	8.16 (0.20)	8.71 (0.29)	8.38 (0.17)	8.84 (0.28)	8.05 (0.09)
	CSDS	8.46 (0.23)	9.04 (0.32)	8.53 (0.21)	9.12 (0.27)	8.15 (0.16)
Texture	DCS-LA	3.69 (0.09)	4.40 (0.43)	3.44 (0.09)	4.17 (0.13)	3.66 (0.07)
	ADS	3.57 (0.09)	4.02 (0.56)	**3.41** (0.008)	4.03 (0.15)	3.60 (0.06)
	MDS	3.56 (0.09)	3.93 (0.63)	3.42 (0.08)	4.04 (0.15)	3.60 (0.06)
	CSDS	3.63 (0.10)	4.01 (0.18)	3.43 (0.11)	3.43 (0.11)	3.58 (0.08)

The overproduction phase was performed using an initial pool of DT classifiers generated by Bagging. The best result for each data set is shown in bold.

References

[1] X. Zhu, X. Wu, Y. Yang, Dynamic selection for effective mining from noisy data streams, in: Proceedings of Fourth IEEE International Conference on Data Mining, 2004, pp. 305–312.

[2] G. Giacinto, F. Roli, Dynamic classifier selection based on multiple classifier behaviour, Pattern Recognition 34 (9) (2001) 1879–1881.

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

[4] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.

[5] Y. Freund, R. Schapire, Experiments with a new boosting algorithm, in: Proceedings of XIII International Conference on Machine Learning, 1996, pp. 148–156.

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

[7] J. Kittler, M. Hatef, R.P.W. Duin, J. Matas, On combining classifiers, IEEE Trans. Pattern Anal. Mach. Intell. 20 (3) (1998) 226–238.

[8] H.W. Shin, S.Y. Sohn, Selected tree classifier combination based on both accuracy and error diversity, Pattern Recognition 38 (2) (2005) 191–197.

[9] D. Partridge, W.B. Yates, Engineering multiversion neural-net systems, Neural Comput. 8 (4) (1996) 869–893.

[10] A.J.C. Sharkey, N.E. Sharkey, The "test and select" approach to ensemble combination, in: Proceedings of the First International Workshop on MCS, 2000, pp. 30–44.

[11] E.M. Dos Santos, R. Sabourin, P. Maupin, Single and multi-objective genetic algorithms for the selection of ensemble of classifiers, in: Proceedings of International Joint Conference on Neural Networks, 2006, pp. 5377–5384.

[12] Z.-H. Zhou, J. Wu, W. Tang, Ensembling neural networks: many could be better than all, Artif. Intell. 137 (1–2) (2002) 239–263.

[13] D. Ruta, B. Gabrys, Classifier selection for majority voting, Inform. Fusion 6 (1) (2005) 163–168.

[14] D. Corne, J. Knowles, No free lunch and free leftovers theorems for multiobjective optimization problems, in: Proceedings of Evolutionary Multi-Criterion Optimization, Faro, Portugal, 2003, pp. 327–341.

[15] K. Deb, Multi-Objective Optimization using Evolutionary Algorithms, Wiley, UK, 2001.

[16] E.M. Dos Santos, R. Sabourin, P. Maupin, Ambiguity-guided dynamic selection of ensemble of classifiers, in: Proceedings of International Conference on Information Fusion, 2007.

[17] L.K. Hansen, C. Liisberg, P. Salomon, The error-reject tradeoff, Open Systems Inform. Dyn. 4 (2) (1997) 159–184.

[18] D. Fagundes, A. Canuto, Applying weights in the functioning of the dynamic classifier selection method, in: Proceedings of Ninth Brazilian Symposium on Neural Networks, 2006.

[19] S. Singh, M. Singh, A dynamic classifier selection and combination approach to image region labelling, Signal Process.: Image Commun. 20 (2005) 219–231.

[20] H.W. Shin, S.Y. Sohn, Combining both ensemble and dynamic classifier selection schemes for prediction of mobile internet subscribers, Expert Syst. Appl. 25 (2003) 63–68.

[21] L.I. Kuncheva, Cluster-and-selection model for classifier combination, in: Proceedings of International Conference on Knowledge Based Intelligent Engineering Systems and Allied Technologies, 2000, pp. 185–188.

[22] R. Liu, B. Yuan, Multiple classifier combination by clustering and selection, Inform. Fusion 2 (2001) 163–168.

[23] 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.

[24] V. Gunes, M. Menard, P. Loonis, A multiple classifier system using ambiguity rejection for clustering-classification cooperation, Uncertainty, Fuzziness and Knowledge-Based Systems 8 (6) (2000) 747–762.

[25] A. Ko, R. Sabourin, A. Britto Jr., From Dynamic Classifier Selection to Dynamic Ensemble Selection, Pattern Recognition 41 (5) (2008) 1718–1731.

[26] A.M.P. Canuto, R.G.F. Soares, A. Santana, M.C.P. de Souto, Using accuracy and diversity to select classifiers to build ensembles, in: Proceedings of International Joint Conference on Neural Networks, 2006, pp. 2289–2295.

[27] D.D. Margineantu, T.G. Dietterich, Pruning adaptive boosting, in: Proceedings of the International Conference on Machine Learning, 1997, pp. 358–366.

[28] 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.

[29] G. Zenobi, P. Cunningham, Using diversity in preparing ensembles of classifiers based on different feature subsets to minimize generalization error, in: Proceedings of European Conference on Machine Learning, 2001, pp. 576–587.

[30] G. Tremblay, R. Sabourin, P. Maupin, Optimizing nearest neighbour in random subspaces using a multi-objective genetic algorithm, in: Proceedings of ICPR, 2004, pp. 208–211.
[31] A.J. Grove, D. Schuurmans, Boosting in the limit: maximizing the margin of learned ensembles, in: Proceedings of 15th National Conference on Artificial Intelligence, 1998, pp. 692–699.
[32] P.J. Gother, Nist special database 19—handprited forms and characters database, National Institute of Standard and Technology—NIST: database CD documentation, 1995.
[33] 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.
[34] S.B. Serpico, L. Bruzzone, F. Roli, An experimental comparison of neural and statistical non-parametric algorithms for supervised classification of remote-sensing images, Pattern Recognition Lett. 17 (13) (1996) 1331–1341.
[35] Y. Park, J. Sklansky, Automated design of linear tree classiers, Pattern Recognition 23 (12) (1990) 1393–1412.
[36] J.R. Quinlan, Programs for Machine Learning, Morgan Kaufmann, Los Altos, CA, 1993.
[37] L. Didaci, G. Giacinto, F. Roli, G.L. Marcialis, A study on the performances of the dynamic classifier selection based on local accuracy estimation, Pattern Recognition 28 (11) (2005) 2188–2191.
[38] P.V.W. Radtke, R. Sabourin, T. Wong, Classification system optimization with multi-objective genetic algorithms, in: Proceedings of 10th International Workshop on Frontiers in Handwriting Recognition, 2006.
[39] F. Rheaume, A.-L. Jousselme, D. Grenier, E. Bosse, P. Valin, New initial basic probability assignments for multiple classifiers, in: Proceedings of XI Signal Processing, Sensor Fusion, and Target Recognition, 2002, pp. 319–328.

About the Author—EULANDA M. DOS SANTOS received a B.Sc. degree in Informatics from Federal University of Para (Brazil) and a M.Sc. degree in Informatics from Federal University of Paraiba (Brazil) in 1999 and 2002, respectively. Currently, she is a Ph.D. candidate in Engineering at École de Technologie Supérieure, University of Quebec (Canada). Her research interests include pattern recognition, ensembles of classifiers and evolutionary computation.

About the Author—ROBERT SABOURIN received B.Sc., M.Sc. and Ph.D. degrees in Electrical Engineering from the École Polytechnique de Montreal (Canada) in 1977, 1980 and 1991, respectively. In 1977, he joined the Physics Department of the Université de Montreal (Canada) where he was responsible for the design and development of scientific instrumentation for the Mégantic Astronomical Observatory. In 1983, he joined the staff of the École de Technologie Supérieure, Université du Québec (Canada) where he co-founded the Department of Automated Manufacturing Engineering where he is currently Full Professor. In 1992, he joined also the Computer Science Department of the Pontifícia Universidade Católica do Paraná (Curitiba, Brazil). Since 1996, he is a senior member of the Centre for Pattern Recognition and Machine Intelligence (CENPARMI, Concordia University, Canada).

About the Author—PATRICK MAUPIN received a B.Sc. degree from Université de Montréal, Canada, a Diplôme d'études supérieures and a M.Sc. degree from Université du Québec à Montréal, Canada in 1992, 1995 and 1997, respectively. Between 1998 and 2001 he worked as a consultant in GIS information processing and multivariate statistical analysis, in particular for Québec's Public Health Authority and as a research assistant at McGill University (Canada) on hyperspectral data processing. In 2001, he joined DRDC-Valcartier as a Defence Scientist in the Decision Support Systems for Command and Control Section and is currently the leader of the Situation Analysis and Monitoring group. His research interests include information fusion, reasoning under uncertainty, epistemic logic, situation analysis, human-centered pattern recognition and the design of pattern recognition systems for sensor networks (from data collection to hardware implementation).