Pattern Recognition ■ (■■■■) ■■■–■■■

Contents lists available at ScienceDirect

Pattern Recognition

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

Dynamic selection of classifiers—A comprehensive review

Alceu S. Britto Jr.[a,b,*], Robert Sabourin[c], Luiz E.S. Oliveira[d]

[a] Pontifícia Universidade Católica do Paraná (PUCPR), Curitiba, PR, Brazil
[b] Universidade Estadual de Ponta Grossa (UEPG), Ponta Grossa, PR, Brazil
[c] École de technologie supérieure (ÉTS), Université du Québec, Montreal, QC, Canada
[d] Universidade Federal do Paraná (UFPR), Curitiba, PR, Brazil

ARTICLE INFO

Article history:
Received 28 August 2013
Received in revised form
3 May 2014
Accepted 7 May 2014

Keywords:
Ensemble of classifiers
Dynamic selection of classifiers
Data complexity

ABSTRACT

This work presents a literature review of multiple classifier systems based on the dynamic selection of classifiers. First, it briefly reviews some basic concepts and definitions related to such a classification approach and then it presents the state of the art organized according to a proposed taxonomy. In addition, a two-step analysis is applied to the results of the main methods reported in the literature, considering different classification problems. The first step is based on statistical analyses of the significance of these results. The idea is to figure out the problems for which a significant contribution can be observed in terms of classification performance by using a dynamic selection approach. The second step, based on data complexity measures, is used to investigate whether or not a relation exists between the possible performance contribution and the complexity of the classification problem. From this comprehensive study, we observed that, for some classification problems, the performance contribution of the dynamic selection approach is statistically significant when compared to that of a single-based classifier. In addition, we found evidence of a relation between the observed performance contribution and the complexity of the classification problem. These observations allow us to suggest, from the classification problem complexity, that further work should be done to predict whether or not to use a dynamic selection approach.

1. Introduction

Classification is a fundamental task in Pattern Recognition, which is the main reason why the past few decades have seen a vast number of research projects devoted to classification methods applied to different fields of the human activity. Although the methods available in the literature may differ in many respects, the latest research results lead to a common conclusion; creating a monolithic classifier to cover all the variability inherent to most pattern recognition problems is somewhat unfeasible.

With this in mind, many researchers have focused on Multiple Classifier Systems (MCSs), and consequently, many new solutions have been dedicated to each of the three possible MCS phases: (a) generation, (b) selection, and (c) integration, which are represented in Fig. 1. In the first phase, a pool of classifiers is generated; in the second phase, one or a subset of these classifiers is selected, while in the last phase, a final decision is made based on the prediction(s) of the selected classifier(s). It is worth noting that such a representation is not unique, since the selection and integration phases may be facultative. For instance, one may find MCS where the whole pool of classifiers is used without any selection or systems where just one classifier is selected from the pool, making the integration phase unnecessary.

In a nutshell, recent contributions with respect to the first phase indicate that the most promising direction is to generate a pool of accurate and diverse classifiers. The authors in [1] state that a necessary and sufficient condition for an ensemble of classifiers to be more accurate than any of its individual members is for the classifiers to be accurate and diverse. Dietterich [2] explains that an accurate classifier has an error rate lower than the random guessing on new samples, while two classifiers are diverse if they make different errors on new samples. The rationale behind this is that the individual accurate classifiers in the pool may compete each other by making different and perhaps complementary errors. As for the selection phase, interesting results have been obtained by selecting specific classifiers for each test pattern, which characterizes a dynamic selection of classifiers, instead of using the same classifier for all of them (static selection). Moreover, additional contributions have been observed when ensembles are selected instead of just one single classifier. In such a case,

* Corresponding author at: Post-graduate Program in Informatics (PPGIa), Pontifical Catholic University of Parana Rua Imaculada Conceição, 1155, Curitiba (PR), 80215-901, Brazil. Tel.: +55 41 3271 1669; fax: +55 41 3271 2121.
E-mail addresses: alceu@ppgia.pucpr.br (A.S. Britto Jr.), robert.sabourin@etsmtl.ca (R. Sabourin), lesoliveira@inf.ufpr.br (L.E.S. Oliveira).

http://dx.doi.org/10.1016/j.patcog.2014.05.003
0031-3203/

Fig. 1. The possible phases of a Multiple Classifier System.

the outputs of the selected classifiers must be combined and the third phase of the MCS is necessary. The main contributions for this phase have been comprised of different strategies combining the classifiers and the assumption that the best integration choice is usually problem depended.

The focus of this paper is on the second phase of an MCS, particularly, the approaches based on dynamic selection (DS) of classifiers or ensembles of such classifiers. Despite the large number of DS methods available in the literature, there is no comprehensive study available to those wishing to explore the advantages of using such an approach. In addition, due to the high computational cost usually observed in the DS solutions, its application is often criticized. In fact, the decision as to whether or not to use DS is still an open question.

In this scenario, we have three research questions, namely:

1. Are the performance results of the DS methods reported in the literature significantly better than those obtained by a single-based classifier approach?
2. Is there any relation between the classification complexity and the observed DS performance for a given problem?
3. Can we predict whether or not DS should be used for a given classification problem?

To answer these questions, we have reviewed several works on dynamic selection and performed a thorough statistical analysis of the results reported in the literature for different classification problems.

The motivation for investigating the possible existence of a relation between the DS contribution and the complexity of a classification problem is inspired by previous works in which the data complexity is used to better define the classifier models. An interesting work in this vein is presented in [3], in which the authors use geometrical characteristics of data to determine the classifier models. Two other interesting studies are presented in [4,5], where the authors characterize the behavior of a specific classifier approach considering problems with different complexities.

With this in mind, our contribution is two-fold that (a) presents a comprehensive review of the main DS methods available in the literature, providing a taxonomy for them and (b) performs a further analysis of the DS results reported in the literature to determine when to apply DS.

This paper is organized as follows. After this brief introduction, Section 2 presents the main basic concepts and definitions related to the dynamic selection of classifiers. Section 3 presents the state of the art of DS methods and describes the suggested taxonomy. The algorithms of some key examples of each category are presented based on the same notation to facilitate comprehension. Section 4 presents further analysis of the DS results available in the literature, in a bid to answer our research questions. Finally, Section 5 presents the conclusions and further works.

2. Basic concepts and definitions

This section presents the main concepts related to MCS and DS approaches, which represent the necessary background for the comprehension of the different works available in the literature.

The first concepts are related to the generation phase of the MCS. As described earlier, this first phase is responsible for the generation of a pool of base classifiers, considering a given strategy, to create diverse and accurate experts. A pool may be composed of homogeneous classifiers (same base classifiers) or heterogeneous classifiers (different base classifiers). In both cases, some diversity is expected. The idea is to generate classifiers that make different mistakes, and consequently, show some degree of complementarity. A comprehensive study of different diversity measures may be found in the work of Kuncheva and Whitaker [6]. The schemes to provide diversity are categorized in [7] as implicit, when there is no use of diversity measures during the generation process, or as explicit, in opposite cases.

In homogeneous pools, diversity is achieved by varying the information used to construct their elements, such as changing the initial parameters, using different subsets of training data (Bagging [8], Boosting [9]), or using different feature subspaces (Random Subspace Selection [10]). On the other hand, the basic idea behind heterogeneous pools is to obtain experts that differ in terms of the properties and concepts on which they are based.

Regarding the selection phase of an MCS, the main concepts are related to the type of selection and the notion of classifier competence. The type of selection may be static or dynamic, as explained earlier. The rationale behind the preference for dynamic over static selection is to select the most locally accurate classifiers for each unknown pattern. Both static and dynamic schemes may be devoted to classifier selection, providing a single classifier, or to ensemble selection, selecting a subset of classifiers from the pool.

Usually, the selection is done by estimating the competence of the classifiers available in the pool on local regions of the feature space. To that end, a partitioning process is commonly used during the training or testing phases of the MCS. In this process, the feature space is divided into different partitions, and the most capable classifiers for each of them are determined. In static selection methods, the partitioning is usually based on clustering or evolutionary algorithms, and it is executed during the training phase. This means that the classifier competence is always determined during the training phase of the system. Although it is possible to apply similar strategies for dynamic selection methods, what is mostly commonly seen with this approach is the use of a partitioning scheme based on the NN-rule to define the neighborhood of the unknown pattern in the feature space during the testing phase. In this case, the competence of each classifier is defined on a local region on the entire feature space represented by the training or validation dataset.

Regarding the competence measures, the literature reports several of them, which consider the classifiers either individually or in groups. This is the basis of the DS taxonomy proposed in the next section. It is worth noting that, basically, the individual-based measures most often take into account the classifier accuracy. However, the measures are conducted in different ways. For instance, one may find measures based on pure accuracy (overall local accuracy or local class accuracy) [11], ranking of classifiers [12], probabilistic information [13,14], classifier behavior calculated on output profiles [15–17], and oracle information [18,19]. Moreover, we may find measures that consider interactions among classifiers, such as diversity [20–22], ambiguity [23,24,17] or other grouping approaches [25].

The third phase of an MCS consists in applying the selected classifiers to recognize a given testing pattern. In cases where all classifiers are used (without selection) or when an ensemble is selected, a fusion strategy is necessary. For the integration of the classifier outputs, there are different schemes available in the literature. Complete details regarding the combination methods and their taxonomy are available in Jain et al. [26] and in Kittler et al. [27].

Please cite this article as: A.S. Britto Jr. et al., Dynamic selection of classifiers—A comprehensive review, Pattern Recognition (2014), http://dx.doi.org/10.1016/j.patcog.2014.05.003

With respect to the experimental protocols used to evaluate a DS approach, one may usually find a comparison of the proposed method against the single best (*SB*) classifier, a combination of all classifiers in the pool (*CC*), the static selection (*SS*) using the same pool, and other DS approaches. In fact, it suggests that the minimum requirement for a DS method is to surpass the SB, CC, and any SS in the same pool. Moreover, the concept of oracle performance is usually present in the evaluation of the proposed methods. This means that the proposed method is compared against the upper limit in terms of performance of the pool of classifiers. The oracle performance is estimated by considering that if at least one classifier can correctly classify a particular test sample, then the pool can also do so as well.

Finally, since in the next section, we present the algorithms of some key works available in the literature, for the sake of comprehension, we have adopted the same notation. To that end, let $\Omega = \{\omega_1, \omega_2, \ldots, \omega_L\}$ denote the set of classes of a hypothetical pattern recognition problem, while *Tr*, *Va*, and *Te* represent training, validation, and testing datasets, respectively. Moreover, let $C = \{c_1, c_2, \ldots, c_M\}$ be a pool composed of M diverse classifiers, and $EoC = \{EoC_1, EoC_2, \ldots, EoC_N\}$ be a pool of N diverse ensembles of classifiers. The unknown sample, or the testing sample, is referred to as t. In addition, let Ψ be the region of the feature space used to compute the competence of the base classifiers.

The next section presents the proposed taxonomy, the key works that represent each category, and some general statistics related to their performances.

3. State of the art

It is worth noting that a categorization of the existing methods is not a trivial task since they present a large overlapped region. So, in order to better present the state of the art, we first outline a taxonomy in the context of MCS, which is inspired by the taxonomy of ensembles [7], and then we review the literature following the diagram depicted in Fig. 2. However, here, the focus is DS, and the main criterion is the source of information used to evaluate the competence of the base classifiers in the pool. The measures of competence are organized into two groups: individual-based and group-based. The former presents the measures wherein somehow, the individual performance of each classifier is the main source of information. This category was subdivided into other five subcategories, as follows: ranking, accuracy, probabilistic, behavior, and oracle-based. The latter is composed of the measures that consider the interaction among the elements in the pool. This category was subdivided into three subcategories, as follows: diversity, ambiguity, and data handling-based.

The next subsections describe each category by reviewing the most important methods available in the literature. However, before proceeding it is necessary to clarify some points. First, the proposed categorization contemplates only the DS methods where the competence of each base classifier, or its contribution inside an ensemble, is used to decide whether or not it will be selected. Second, even knowing the importance of selection mechanisms based on dynamic weighting of scores or mixture of experts [28–30], they were not described here since they are dedicated to the use of a specific base classifier (multilayer perceptron neural networks). Third, it is known that the best strategy for calculating competence of the base classifiers in a DS method is to use a validation set. However, since we follow the original description of each algorithm can be observed in some cases that the training set is used instead. In addition, most of algorithms originally defined to select one classifier may be modified to select an ensemble by just applying an additional threshold on the proposed competence measure.

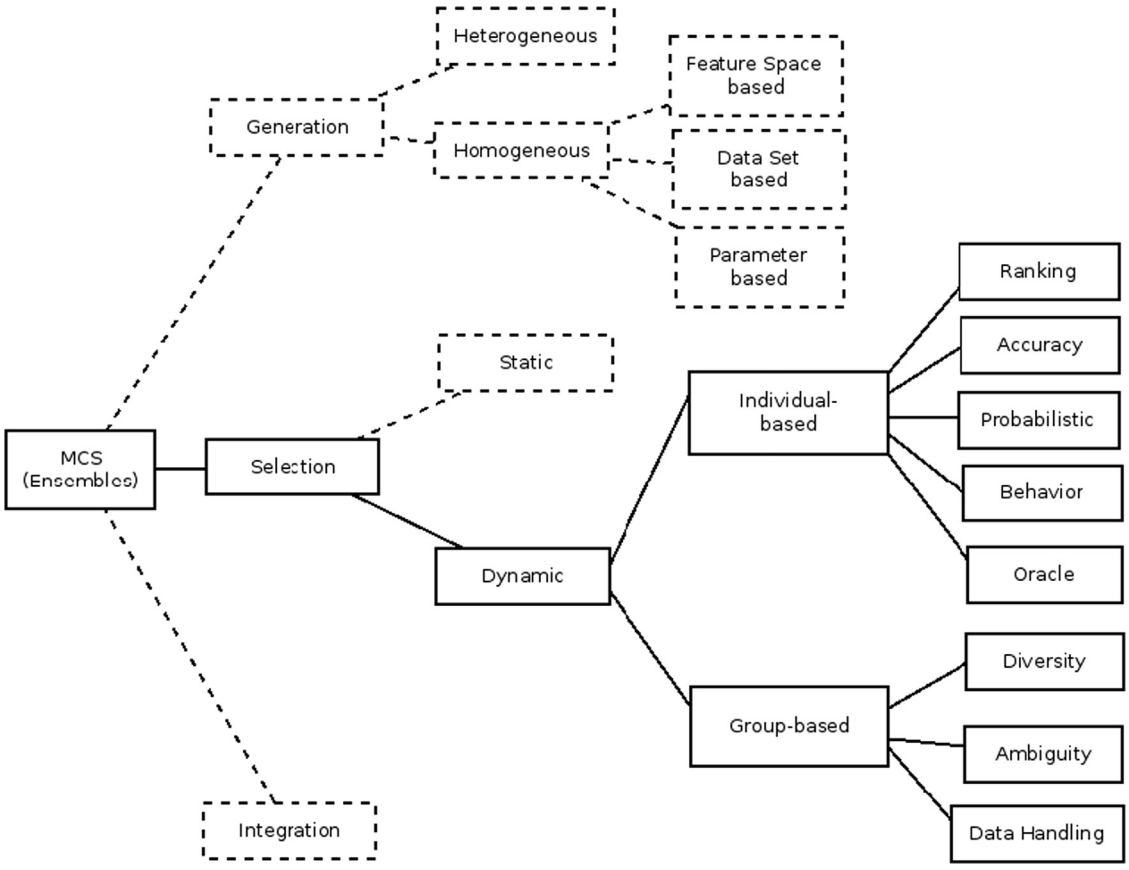

Fig. 2. Proposed DS Taxonomy in the context of MCS.

Please cite this article as: A.S. Britto Jr. et al., Dynamic selection of classifiers—A comprehensive review, Pattern Recognition (2014), http://dx.doi.org/10.1016/j.patcog.2014.05.003

3.1. Individual-based measures

In this category, the classifiers are selected based on their individual competence on the whole feature space represented by the training or validation set, or on part of it referred to as a local region. As described earlier, the local region may be defined in advance, during the training phase, by using partitioning techniques, or during the testing phase, by using the *NN-rule* to define the k-nearest-neighbors of the unknown pattern in the feature space also represented by the training or validation datasets. Basically, while the main source of information in this category is related to classifier accuracy, its subcategories however differ in their representation.

3.1.1. Ranking-based measures

The methods in this subcategory exploit a rank of the classifiers available in the pool. An interesting approach was proposed in 1993 by Sabourin et al. in [12], referred in this paper as the *DSC-Rank* (see Algorithm 1). It may be considered as one of the pioneers in DS. In their work, the ranking was done by estimating three parameters related to the correctness of the classifiers in the pool. The mutual information of these three parameters with correctness was estimated using part of their training dataset. Let X be a set of classifier parameters, like those suggested by the authors for their k-nearest neighbors base classifiers, the distance to the winner, the distance to the first non-winner, and the distance ratio of the first non-winner to the winner. In addition, let S be the classifier "success" variable, defined as $S=\delta(t,o)$, where t is the true label of a given training sample, and o is the classifier output. The mutual information between S and X can be calculated as

$$I(S,X)=H(S,X)-H(X) \tag{1}$$

where $H(X)$ is estimated based on $p(x)$, the probability mass function of outcome x, as

$$H(X)=-\sum_{x\epsilon X} p(x)\log(p(x)) \tag{2}$$

and the joint entropy between S and X is given by Eq. (3), where $p(l,x)$ is the probability of the classifier parameter x being related to correctness.

$$H(S,X)=-\sum_{l\epsilon S}\sum_{x\epsilon X} p(l,x)\log(p(l,x)) \tag{3}$$

The rationale behind the calculation of $I(S,X)$ is to estimate the uncertainty in the decision that is resolved by observing each classifier parameter. After determining the most informative classifier parameters, the authors defined what they called a meta-pattern space (*MP*), represented by a subset of training samples, where for each element it is kept the values of the classifier parameters. During the classification step, the parameter values of the classifiers associated with the nearest neighbor of the test pattern in the meta-pattern space are ranked. The authors have considered a single parameter decision based on the largest parameter value. The classifier with the best ranking position is selected. Thus, it selects just one classifier and the partitioning process is done during the training phase, when *MP* is created.

Algorithm 1. *DSC-Rank* method.

Input the pool of classifiers C; the set of classifier parameters X; the datasets Tr and Te;
Output c_t^*, the most promising classifier for each testing sample t in Te;
Compute $S=(t,o)$ as the classifiers "success" variable using the training samples in Tr;
Compute $I(S,X)$ as the mutual information between X and S using the training samples in Tr;
Determine X' as the most informative classifier parameters based on $I(S,X)$;
Create the meta-pattern space MP, as a subset of Tr with the corresponding values of the parameters in X';
for each testing sample t in Te **do**
  Apply NN-rule to find ψ as the nearest neighbor of the unknown sample t in MP;
  Rank the classifiers based on the parameter values associated to ψ;
  Select c_t^* as the classifier in the best ranking position;
  Use c_t^* to classify t;
end for

The second example of this category is the *DS-MR* proposed by Woods et al. in [11]. In fact, it is a simplification of the original *DSC-Rank method*. Different from the original, the modified method ranks the classifiers based on local class accuracy calculated as the number of correct classified samples on the k-nearest neighbors of the unknown pattern in the training set. During classification, a local region of the feature space near the test pattern is defined, the rank is constructed, and the best classifier is selected. Another difference from the original ranking approach is that the partitioning process used to define the local region in estimating the classifier competence is done during the testing phase.

3.1.2. Accuracy-based measures

Here, the main characteristic is the estimation of the classifier accuracy, overall or local, as a simple percentage of corrected classified samples. The two variations of the *DS-LA* method proposed in [11] are examples of this subcategory.

Algorithm 2. DS-LA OLA-based method.

Input the pool of classifiers C; the datasets Tr and Te; and the neighborhood size K;
Output c_t^*, the most promising classifier for each testing sample t in Te;
for each testing sample t in Te **do**
  Submit t to all classifiers in C;
  if (all classifiers agree with the label of the sample t) **then**
    return the label of t;
  else
    Find Ψ as the K nearest neighbors of the sample t in Tr;
    for each classifier c_i in C **do**
      Calculate OLA_i as the percentage of correct classification of c_i on Ψ;
    end for
    Select the best classifier for t as $c_t^*=\arg\max_i\{OLA_i\}$;
    Use c_t^* to classify t;
  end if
end for

The *DS-LA* was proposed by Woods et al. with two different versions. The first calculates the overall local accuracy (OLA) of the base classifiers in the local region of the feature space close to the unknown pattern in the training dataset (see Algorithm 2). The OLA of each classifier is computed as the percentage of the correct recognition of the samples in the local region.

Algorithm 3. DS-LA LCA-based method.

Input the set of classes Ω; the pool of classifiers C; the datasets Tr and Te; and the neighborhood size K;

Output c_t^*, the most promising classifier for each testing sample t in Te;
for each testing sample t in Te **do**
  Submit t to all classifiers in C;
  if (all classifiers agree with the label of the sample t) **then**
    return the label of t;
  else
    for each classifier c_i in C **do**
      $\omega_j = c_i(t)$, the predicted output of c_i for the sample t;
      Find Ψ as the K nearest neighbors of the sample t in Tr that belongs to the class ω_j;
      Calculate $LCA(i,j)$ as the percentage of correct labeled samples of class ω_j by the classifier c_i on Ψ;
    end for
    Select the best classifier for t as $c_t^* = \arg\max_i\{LCA(i,j)\}$;
    Use c_t^* to classify t;
  end if
end for

In the second one, they calculate the local class accuracy (LCA), as shown in Algorithm 3. The LCA is estimated for each base classifier as the percentage of correct classifications within the local region, but considering only those examples where the classifier has given the same class as the one it gives for the unknown pattern. In both versions of the *DS-LA* method, OLA and LCA-based, the partitioning of the feature space is defined based on the k-nearest neighbors of the unknown pattern in the training dataset during the testing phase. Moreover, only one classifier is selected from the pool.

3.1.3. Probabilistic-based measures

More than just estimating the classifier accuracy based on a simple percentage of corrected classified samples, the methods in this subcategory use some probabilistic representation. Two interesting schemes, named *A Priori* and *A Posteriori* selection, were proposed in [13]. Both schemes select a single classifier from the pool based on a local region defined by the k-nearest neighbors of the test pattern in the training set during the testing phase.

In the *A Priori* method, a classifier is selected based on its accuracy within the local region, without considering the class assigned to the unknown pattern. This measure of classifier accuracy is calculated as the class posterior probability of the classifier c_j on the neighborhood Ψ of the unknown sample t. As we can see in Eq. (4), the class posterior probability is weighted by δ_i, which represents the Euclidian distance between the sample ψ_i and the unknown pattern t.

Similarly, in the *A Posteriori* method, local accuracies are estimated using the class posterior probabilities and the distances of the samples in the defined local region (neighborhood of size K). However, Eq. (5) shows that in this measure the class ω_l assigned by the classifier c_j to the unknown sample t is taken into account. Both methods are presented in Algorithm 4, where the *Threshold* value suggested by the authors was 0.1.

$$\hat{p}(correct_j) = \frac{\sum_{i=1}^{K} \hat{P}_j(\omega_l|\psi_i \in \omega_l)\cdot\delta_i}{\sum_{i=1}^{K}\delta_i} \tag{4}$$

$$\hat{p}(correct_j|c_j(t)=\omega_l) = \frac{\sum_{\psi_i\in\omega_l} \hat{P}_j(\omega_l|\psi_i)\cdot\delta_i}{\sum_{i=1}^{K}\hat{P}_j(\omega_l|\psi_i)\cdot\delta_i} \tag{5}$$

$$\delta_i = \frac{1}{Euclidian\ Distance(\psi_i, t)} \tag{6}$$

Algorithm 4. A Priori/A Posteriori method.

Input the pool of classifiers C; the datasets Tr and Te; the neighborhood size K;
Output c_t^*, the most promising classifier for each unknown sample t in Te;
for each testing sample t in Te **do**
  Find Ψ as the K nearest neighbors of the sample t in Tr;
  for each classifier c_j in C **do**
    Compute $\hat{p}(correct_j)$ on Ψ by using Eq. (4) or Eq. (5);
    if $(\hat{p}(correct_j) \geq 0.5)$ **then**
      $CS = CS \cup c_j$;
    end if
  end for
  $\hat{p}(correct_m) = max_j(\hat{p}(correct_j))$;
  $c_m = \arg\max_j(\hat{p}(correct_j))$;
  $selected = TRUE$;
  for each classifier c_j in CS **do**
    $d = \hat{p}(correct_m) - \hat{p}(correct_j)$;
    if $((j \neq m)$ **and** $(d < Threshold))$ **then**
      $selected = FALSE$;
    end if
  end for
  if $(selected == TRUE)$ **then**
    $c_t^* = c_m$;
  else
    $c_t^* =$ a classifier randomly selected from CS, with $d < Threshold$;
  end if
  Use the classifier c_t^* to classify t;
end for

Kurzynski et al. [14] proposed two interesting classifier competence measures. The first one (*DES-M1*) is an interesting accuracy-based approach, where the competence of each classifier for a given unknown pattern is computed based on a potential function model which is able to estimate the classifier capability of doing the correct classification. The competence of each classifier is computed considering the support it gives for the correct class of each validation sample. The second measure named *DES-M2* is a probabilistic-based example, where the authors use the probability of correct classification of a probabilistic reference classifier (PRC). Both measures inspired by the work described in [31] differ from the majority of the methods available in the literature, since the competence of each classifier is estimated using the whole validation set during the testing phase. The ideas presented in *DES-M2* were extended and improved in [32], where the main contribution is the modeling scheme based on a unified model representing the whole vector of class supports. Different variants of the proposed method were evaluated considering the selection of classifiers and ensembles. The best results were achieved by the system named *DES-CS*, which selects an ensemble of classifiers, considers continuous-valued outputs and weighted class supports.

3.1.4. Behavior-based measures

The methods in this subcategory are based on some kind of behavior analysis using classifier predictions as information sources. Inspired by the Behavior-Knowledge Space (BKS) proposed by Huang et al. in 1995 [33], Giacinto et al., in [15], propose a dynamic classifier selection based on multiple classifier behavior (MCB), named *DS-MCB* (see Algorithm 5). They estimate the MCB

using a similarity function to measure the degree of similarity of the output profiles of all base classifiers. First, a local region Ψ is defined as the k-nearest neighbors of the unknown pattern in the training set. Then, the similarity function is used as a filter to preselect from Ψ, the samples for which the classifiers present similar behavior to that observed for the unknown sample t. The remaining samples are used to select the most accurate classifier by using local classifier accuracy (OLA). Finally, if the selected classifier is significantly better than the others in the pool based on a defined threshold value, then it is used to classify the unknown sample. Otherwise, all classifiers are combined using the majority voting rule (MVR).

Algorithm 5. DS-MCB method.

Input the set of classes Ω; the pool of classifiers C; the datasets Tr and Te; the neighborhood size K;
Output c_t^*, the most promising classifier for each unknown sample t in Te;
for each testing sample t in Te **do**
  Compute the vector MCB_t as the class labels assigned to t by all classifiers in C;
  Find Ψ as the K nearest neighbors of the test sample t in Tr;
  for each sample ψ_j in Ψ **do**
    Compute MCB_{ψ_j} as the class label assigned to ψ_j by all classifiers in C;
    Compute Sim as the similarity between MCB_t and MCB_{ψ_j};
    if ($Sim > SimilarityThreshold$) *then*
      $\Psi' = \Psi' \cup \psi_j$;
    end if
  end for
  for each classifier c_i in C **do**
    Calculate OLA_i as the local classifier accuracy of c_i on Ψ';
  end for
  Select the best classifier $c_t^* = \arg\max_i\{OLA_i\}$;
  if (c_t^* is significantly better than the other classifiers on Ψ') **then**
    Use the classifier c_t^* to classify t
  else
    Apply MVR using all classifiers in C to classify t;
  end if
end for

The core of this algorithm is the vector named MCB (Multiple Classifier Behavior) which can be defined as $MCB_\psi = \{C_1(\psi), C_2(\psi), \ldots, C_M(\psi)\}$. It contains the class labels assigned to the sample ψ by the M classifiers in the pool. The measure of similarity Sim can be defined as

$$Sim(\psi_1, \psi_2) = \frac{1}{M}\sum_{i=1}^{M} T_i(\psi_1, \psi_2) \tag{7}$$

where

$$T_i(\psi_1, \psi_2) = \begin{cases} 1 & \text{if } C_i(\psi_1) = C_i(\psi_2) \\ 0 & \text{if } C_i(\psi_1) \neq C_i(\psi_2) \end{cases} \tag{8}$$

Another interesting method in this category is the *DSA-C* proposed by Cavalin et al. in [34] and [17]. Different from the previously described works, the *DSA-C* method may select one or a subset of ensembles. To that end, first they computed the output profiles of each available ensemble using a validation set. Different approaches for estimating the similarity between output profiles were used, while the selection of ensembles was done by choosing those with output profiles most similar to the output profile estimated for the testing pattern.

Nabiha et al. [16] proposed the dynamic selection of ensembles (*DECS-LA*) by calculating the reliability of each base classifier over a validation set during the testing phase. The reliability of each classifier is derived from its confusion matrix obtained over the validation set. In the selection step, each base classifier is evaluated considering its accuracy on a local region close to the unknown pattern combined with its reliability, which may be considered as a kind of probability that the local behavior is correct.

3.1.5. Oracle-based measures

To some extent, the methods here use the concept of the oracle, i.e., the one who may provide wise counsel. In the linear random oracle proposed by Kuncheva [18], each classifier in the pool has a subset with two sub-classifiers and an oracle. The oracle in their work is a random linear function that is responsible for deciding which of two possible sub-classifiers will be used for an unknown pattern. After consulting the oracle of each base classifier, the sub-classifiers selected are combined.

Algorithm 6. KNORA-Eliminate (KNE) method.

Input pool of classifiers C; meta-space sVa where for each sample is assign the classifiers that correctly recognize it; the testing set Te, and the neighborhood size K;
Output EoC_t^*, an ensemble of classifiers for each testing sample t in Te;
for each testing sample t in Te **do**
  k=K;
  while $k > 0$ **do**
    Find Ψ as the k nearest neighbors of the test sample t in sVa;
    for each classifier c_i in C **do**
    if (c_i correctly recognizes all samples in Ψ) **then**
      $EoC_t^* = EoC_t^* \cup c_i$;
    end if
    end for
    if ($EoC_t^* == \emptyset$) **then**
      $k = k - 1$;
    else
      break;
    end if
  end while
  if ($EoC_t^* == \emptyset$) **then**
    Find the classifier c_i that correctly recognizes more samples in Ψ;
    Select the classifiers able to recognize the same amount of samples of c_i to compose the ensemble EoC_t^*;
  end if
  Use the ensemble EoC_t^* to classify t;
end for

Algorithm 7. KNORA-Union (KNU) method.

Input pool of classifiers C; meta-space sVa where for each sample is assign the classifiers that correctly recognize it; the testing set Te, and the neighborhood size K;
Output EoC_t^*, an ensemble of classifiers for each testing sample t in Te;
for each testing sample t in Te **do**
  Find Ψ as the K nearest neighbors of the test sample t in sVa;
  for each sample ψ_i in Ψ **do**
    for each classifier c_i in C **do**

if (c_i correctly recognize ψ_i) **then**
$EoC_t^* = EoC_t^* \cup c_i$;
end if
end for
end for
Use the ensemble EoC_t^* to classify t;
end for

Another key work in this category is the k-nearest-oracles (*KNORA*) method proposed in [19]. The oracles are represented by the k-nearest neighbors of the unknown pattern in a validation set, where the classifiers that correctly classify each sample are known. This validation set is a kind of meta-space (*sVa* in Algorithms 6 and 7). It is used as the source of "oracles". By finding the k-nearest-neighbors of the test pattern t in *sVa*, we can select the classifiers that correctly recognize these neighbors to classify t. Thus, the oracles may suggest the classifiers that must be used to recognize the unknown pattern. The authors evaluated different schemes to select the classifiers suggested by the oracles. The most promising strategies were Knora-Eliminate (*KNE*) and Knora-Union (*KNU*). The former selects only those classifiers which are able to correctly recognize the entire neighborhood of the testing pattern (see Algorithm 6), while the later selects all classifiers that are able to correctly recognize at least one sample in the neighborhood. As we can see in Algorithm 7, the Knora-Union strategy considers that a classifier can participate in the ensemble more than once if it correctly classifies more than one neighbor.

3.2. Group-based measures

The methods in this category combine the accuracy of the base classifiers with some information related to the interaction among them, such as diversity, ambiguity or complexity.

3.2.1. Diversity-based measures

In 2003, Shin et al. in [20] used a clustering process based on the coefficients of their set of base logistic regression classifiers to create clusters of classifiers. Two clusters of classifiers were selected on the local region of the feature space close to the unknown pattern, one based on accuracy and the other on diversity. The definition of the local regions was done based on the NN-rule on the validation set during the testing phase. In fact, they modified the *DS-LA* approach proposed in [11] by considering the selection of ensembles of classifiers both in terms of accuracy and error diversity.

Santana et al., in [21], combined accuracy and diversity to build ensembles. The classifiers were sorted in decreasing order of accuracy and in increasing order of diversity. Two variations were presented. In the *DS-KNN*, accuracy and diversity are calculated in the local region defined by the k-nearest neighbors of the unknown pattern in the validation set, while in the *DS-Cluster*, the partitioning process is done during the training phase, when a clustering process is used to divide the validation set into clusters where the most promising classifiers will be associated. The diversity is calculated in a pairwise fashion using the double fault diversity measure, the idea being to select the classifiers more diverse among those more accurate (see Algorithm 8).

Algorithm 8. DS-KNN method.

Input the pool of classifiers C; the datasets *Va* and *Te*; the neighborhood size K; the number of classifiers to be selected N' and N'';
Output EoC_t^*, an ensemble of classifiers for each unknown sample t in *Te*;
for each testing sample t in *Te* **do**
Find Ψ as the K nearest neighbors of the sample t in *Va*;
for each classifier c_i in C **do**
Compute A_i as the accuracy of c_i on Ψ;
end for
for each classifier c_i in C **do**
for each classifier c_j in C **do**
if $(i \neq j)$ **then**
Compute D_{ij} as the diversity between c_i and c_j on Ψ;
end if
end for
end for
Create R_1 as the rank of classifiers in C by decreasing order of accuracy A;
Create R_2 as the rank of the classifiers in C by increasing order of diversity D;
Based on R_1, select the N' most accurate classifiers in C to compose the ensemble EoC;
Based on R_2, select the N'' $(N'' < N')$ most diverse classifiers in EoC to compose EoC_t^*;
Use the ensemble EoC_t^* to classify t;
end for

Lysiak et al. [22] considered the use of the diversity measure in the approach based on the randomized reference model proposed in [14], the proposed *DES-CD* method, first selects the most accurate classifier to start the ensemble, and then other classifiers are added to the ensemble as they improve the ensemble diversity.

3.2.2. Ambiguity-based measures

The methods in this category, which are different from diversity, use consensus. One of the pioneers of DS, Srihari et al. propose a classifier selection strategy in 1994 [23], based on the consensus of the classifiers on the top choices.

In the same vein, the authors in [35] and [24] describe the $\overline{A}$ and the *DSA* methods, respectively. Both methods select the ensemble of classifiers from a population of highly accurate ensembles with the lowest ambiguity among its members. The algorithm of the *DSA* method is presented in Algorithm 9. With the use of consensus, the authors observed an increase in the generalization performance since the level of confidence of classification had increased. For each test pattern, they selected, from a pool of diverse ensembles, the one showing the highest consensus in terms of the outputs provided by its members. To this end, the ambiguity of the ith classifier of the ensemble EoC_j for the sample ψ was determined as

$$a_i(\psi) = \begin{cases} 0 & \text{if } c_i(\psi) = EoC_j(\psi) \\ 1 & \text{otherwise} \end{cases} \tag{9}$$

while the ambiguity A of the ensemble EoC_j, considering the neighborhood Ψ, was calculated as denoted in Eq. (10), in which $|\Psi|$ and $|EoC_j|$ are the cardinalities of these sets. As one may see in Eq. (9) each classifier output is compared with the ensemble output, which represents the combined decision of their classifiers.

$$A = \frac{1}{|\Psi| \cdot |EoC_j|} \sum_{i \in EoC_j} \sum_{\psi \in \Psi} a_i(\psi) \tag{10}$$

Algorithm 9. DSA method.

Input the set of classes Ω; the pool of classifiers C; the datasets *Va* and *Te*; the neighborhood size K;
Output EoC_t^*, an ensemble of classifiers for each unknown sample t in *Te*;
$EoC' = OptimizationProcess(C, Va, \Omega)$; /* it generates a pool of N optimized ensembles */

Please cite this article as: A.S. Britto Jr. et al., Dynamic selection of classifiers—A comprehensive review, Pattern Recognition (2014), http://dx.doi.org/10.1016/j.patcog.2014.05.003

for each testing sample t in Te **do**
  if (all N ensembles in EoC' agree on the label of t) **then**
    classify t;
  else
    for each EoC'_i in EoC' **do**
      Compute A_i as the ambiguity of the ensemble EoC'_i by using Eq. (10);
    end for
    Select the best ensemble for t as $EoC^*_t = \arg\min_i\{A_i\}$;
    Use the ensemble EoC^*_t to classify t;
  end if
end for

In [17], the *DSA* method was improved through dynamic multi-stage organization and the use of contextual information. The authors organized the classification dynamically in different layers, according to the test patterns. In addition, they expanded the concept of consensus by considering additional information related to all classes involved instead of considering just the outputs of the most voted and the second most voted ones, in selecting the ensembles.

3.2.3. Data handling-based measure

Some authors used different pieces of group information. That is the case of the method proposed in [25], in which the authors use an adaptive classifier ensemble selection based on the group method of data handling theory (GMDH). A multivariate analysis theory for complex systems modeling firstly described in [36]. Their dynamic ensemble selection algorithm, named GDES, selects the ensemble with the optimal complexity for each test pattern from the initial pool of classifiers, also given the combination weights among the classifiers (see Algorithm 10). To that end, the algorithm deals with the pool of classifiers and a local region of the training set related to the k-nearest neighbors of the test pattern.

Algorithm 10. GDES-based method.

Input pool of classifiers C; the datasets Tr and Te, the neighborhood size K; the set of labels of the training samples L;
Output EoC^*_t, an ensemble of classifiers for each testing sample t in Te;
for each testing sample t in Te **do**
  Find Ψ as the K nearest neighbors of the test sample t on Tr;
  for each classifier c_i in C **do**
    $o_i = c_i(t)$, the output of the ith classifier for the sample t;
    for each sample ψ_i in Ψ **do**
      $O_i = c_i(\psi_i)$, the output of the ith classifier for the sample ψ_i;
    end for
  end for
  Compute $MOC(O_i)$ as the model with optimal complexity by using Ψ, L and the GMDH theory [36];
  Select the ensemble EoC^*_t and the weights for each classifier using $MOC(o_i)$;
  Use the ensemble EoC^*_t to classify t;
end for

3.3. Summary

Table 1 presents the main DS methods available in the literature and discussed in this paper. The first six columns lay out all the overall features of each method. In the table, we can find any given category based on the proposed taxonomy; the type of selection in terms of whether a single classifier or an ensemble of classifiers is selected; the kind of pool created (Homogeneous or Heterogeneous) and the number of classifiers in it. In addition, we can find out when and what kind of partitioning process is used; for instance, during the training phase based on clustering or other scheme, or during the testing phase based on the NN-Rule. The last five columns cover the experiments used to evaluate each method. They provide information related to the quantity and size of the datasets used in the experiments, as well as, the number of wins, ties and losses of the DS method against the SB, CC, SS, and other related methods. These numbers take into account the experiments reported in the original papers, as well as those done by other researchers.

The rationale behind the computing of the number of wins, ties and losses between these approaches is that using such information allows us to compare them even if they were originally evaluated through different experimental protocols. Based on these numbers, Fig. 3 let us compare the performance of each DS method with respect to SB, while Fig. 4 shows a similar comparison of DS against the CC results. In both cases, the data are not normalized in order to show the most cited methods, and consequently, the most evaluated ones.

Although such information may provide us with some insight on each specific method, the main objective of our study is to evaluate the DS approach in general. In that respect, Fig. 5 shows the overall performance of DS in terms of percentage of wins, ties, and losses when compared against the usual alternative approaches. The last bar (General) represents all these alternatives (SB, CC and SS) together. The DS approach has shown better results in 59%, 56% and 68% of the cases when compared to SB, CC, and SS, respectively.

These general statistics have proven to be positive for the DS approach. However, they do not give us any clue about the significance of the results or about when such an approach must be used. To accomplish that, a deeper analysis is done in the next section where a two-step methodology is executed.

4. A further analysis

A preliminary analysis of the last section based on several experiments available in the literature suggests that most often, DS will win when compared to the usual alternative approaches. However, it is worth emphasizing that the results also show that the "no free lunch" theorem [41] holds for such analyses in the sense that DS is not universally better. From all the experiments considered, it is clear that the only way one classification approach can outperform another is if it is specialized to the problem at hand.

To answer our research questions, a deeper analysis is necessary. To that end, a two-step methodology is executed in this section. Basically, the idea is to understand how significant the DS performance contribution is when compared to its alternative approaches, and we try to reveal how such a contribution could be related to the problem complexity. In the first step, different non-parametric statistic tests are used to evaluate the significance of the DS results when compared to SB, CC and SS, while in the second one, a set of complexity measures is used to describe the difficulty of a classification problem and relate it to the observed DS performance.

The experimental protocol considers the comparison of DS, SB, CC and SS approaches based on the computed wins, ties and losses in two sets of experiments. The first set (S1) is composed of all experiments in Table 1, while the second set (S2) is composed of experiments on 12 datasets that represent the intersection among all the studied works. Set S2 was constructed based on a careful search for experiments based on the same datasets divided into similar partitions for training and testing. We successfully found 12 datasets that appear in different works. Table 2 presents these datasets and their main features. The following works were the sources of experimental results for S2 [11,19,25,17,34,24,42].

Please cite this article as: A.S. Britto Jr. et al., Dynamic selection of classifiers—A comprehensive review, Pattern Recognition (2014), http://dx.doi.org/10.1016/j.patcog.2014.05.003

Table 1
Summary of the main features of the DS methods and the performance based on wins, ties, losses.

Ref	Method	Category	Sel. type	Pool type(size)	Partitioning Phase/ Tech	Evaluated in	Datasets S=small L=Large	SB	CC	SS	Other DS
[12]	DSC-Rank	Ranking	CS	Het(2)	Training	–	1L	(6,0,0)	(4,0,2)	NA	NA
						[11]	3S	(1,0,2)	(2,0,1)	NA	(0,0,9)
[11]	DS-LA (OLA)	Accuracy	CS	Het(5)	Testing/NN-Rule	–	3S	(2,0,1)	(3,0,0)	NA	(4,0,5)
						[13]	3S	(3,0,0)	(0,0,3)	NA	(4,1,4)
						[37]	2S	(13,0,5)	NA	(6,0,12)	(0,0,18)
						[21]	2S	(1,1,1)	(1,0,2)	NA	(0,2,1)
						[19]	6S/1L	(27,6,22)	(37,3,15)	(0,0,1)	(95,36,89)
						[14]	6S	(6,1,5)	(5,1,6)	NA	(5,2,41)
						[32]	22S	(17,0,27)	(9,0,35)	NA	(85,9,170)
[11]	DS-LA (LCA)	Accuracy	CS	Het(5)	Testing/NN-Rule	–	3S	(3,0,0)	(3,0,0)	NA	(8,0,1)
						[13]	3S	(3,0,0)	(1,0,2)	NA	(3,0,6)
						[38]	5S	(3,1,6)	(3,0,7)	NA	(8,0,12)
						[35]	3S	NA	(2,0,1)	(1,0,2)	(0,0,3)
						[24]	5S/2L	(21,0,3)	(17,1,6)	(4,0,4)	(4,0,20)
						[19]	6S/1L	(20,8,27)	(30,4,21)	(1,0,0)	(98,29,93)
						[25]	6S	(5,1,0)	(5,1,0)	NA	(5,4,3)
[11]	DS-MR	Ranking	CS	Het(5)	Testing/NN-Rule	–	3S	(2,0,1)	(2,0,1)	NA	(5,0,4)
[13]	A Priori	Probabilistic	CS	Het(5)	Testing/NN-Rule	–	3S	(3,0,0)	(0,0,3)	NA	(0,1,8)
						[19]	6S/1L	(19,8,28)	(28,4,23)	(0,0,1)	(82,34,104)
[13]	A Posteriori	Probabilistic	CS	Het(5)	Testing/NN-Rule	–	3S	(3,0,0)	(0,0,3)	NA	(9,0,0)
						[19]	6S/1L	(16,7,32)	(24,3,28)	(1,0,0)	(51,23,146)
[15]	DS-MCB	Behavior	CS	Het(3)	Testing/NN-Rule	–	2S	(2,0,0)	(2,0,0)	NA	NA
						[14]	6S	(6,1,5)	(6,0,6)	NA	(12,2,34)
						[32]	22S	(18,0,26)	(9,0,35)	NA	(88,6,170)
[37]	DS-MLA	Accuracy	CS	Het(3,4)	Testing/NN-Rule	–	2S	(1,1,1)	(1,0,2)	NA	(1,2,0)
						[14]	6S	(8,0,4)	(8,0,4)	NA	(19,2,27)
						[32]	22S	(20,0,24)	(9,0,35)	NA	(125,8,131)
[21]	DS-KNN	Diversity	ES	Het(10,15)	Testing/NN-Rule	–	2S	(18,0,0)	NA	(13,0,5)	(18,0,0)
[21]	DS-Cluster	Diversity	ES	Het(10,15)	Training/Clustering	–	2S	(18,0,0)	NA	(16,0,2)	(18,0,0)
[35]	$\overline{A}$	Ambiguity	ES	Hom(100)	Training/Opt	–	3S	NA	(2,0,1)	(3,0,0)	(3,0,0)
[19]	KNORA	Oracle	ES	Hom(10,100)	Testing/NN-Rule	–	6S/1L	(33,8,14)	(48,3,5)	(1,0,0)	(150,26,44)
						[14]	6S	(10,0,2)	(8,0,4)	NA	(37,1,10)
						[32]	22S	(25,1,18)	(7,0,37)	NA	(131,5,128)
						[39]	6S	(4,1,1)	(6,0,0)	NA	(0,4,8)
						[40]	2S	(2,0,0)	(1,1,0)	(0,0,2)	(1,0,5)
[24]	DSA	Ambiguity	ES	Hom(100)	Training/Opt	–	5S/2L	(21,0,3)	(21,0,3)	(8,0,0)	(20,0,4)
						[17]	5S/2L	(11,0,8)	(10,1,8)	(10,0,9)	(0,0,19)
[25]	GDES	Data Handling	ES	Hom(10)	Training/Opt	–	6S	(2,0,0)	(8,0,0)	NA	(4,0,0)
[14]	DCS-M2	Probabilistic	ES	Hom(50)/Het (11)	Testing/allVs	–	6S	(12,0,0)	(12,0,0)	NA	(43,1,4)
[32]	DES-CS	Probabilistic	ES	Hom(50)/Het (10)	Training/allVs	–	22S	(39,0,5)	(37,0,7)	NA	(242,0,22)
						[22]	6S	(11,0,1)	(10,1,1)	NA	(5,2,5)
[22]	DES-CD	Diversity	ES	Hom(20)/Het(9)	Training/Opt	–	6S	(12,0,0)	(11,0,1)	NA	(5,2,5)
[40]	OP	Oracle	ES	Hom(100)	Testing/NN-Rule	–	2S	(2,0,0)	(2,0,0)	(2,0,0)	(5,0,1)
[16]	DECS-LA	Behavior	ES	Het(10)	Testing/NN-Rule	–	1L	(2,0,0)	NA	(2,0,0)	NA
[17]	DSA-C	Ambiguity	ES	Hom(100)	Training/Opt	–	5S/2L	(18,0,1)	(19,0,0)	(18,0,1)	(19,0,0)

Hom(20)=pool of 20 homogeneous classifiers, Het(10)=pool of 10 heterogeneous classifiers, Hom (10,100) pools with 10 and 100 classifiers, CS=classifier selection, ES=ensemble selection, Opt=optimization, allVs=all validation samples.

4.1. Significance of the results

In the first step of the methodology used, two non-parametric tests followed by a post hoc test are executed. The objective is to answer our first research question related to the significance of the DS results when compared to a single-based classifier.

The first non-parametric statistic is the simple and well-known sign test [43]. It was calculated on the computed wins, ties and losses reported in Table 1, i.e., the set of experiments S1. Let us consider the comparison of DS and SB. The number of wins, ties and losses is 467, 45 and 273, respectively, amounting to 785 experiments. First, we must add the ties to both wins and losses. Thus, we have 512 wins and 318 losses. In this case, the null hypothesis (H_0) is that the DS and SB approaches are equally successful. To reject the null hypothesis, and show that DS performance is significantly better than SB, DS must satisfy Eq. (11), in which n is the total number of experiments, n_w is the computed DS wins, and Z_α represents the z statistic at significance level α.

$$n_w \geq \frac{n}{2} + Z_\alpha \frac{\sqrt{n}}{2} \tag{11}$$

If we consider $\alpha = 0.05$, then $Z_\alpha = 1.645$. With this setup, the null hypothesis is rejected since $512 > 416$. The final conclusion is that with a significance level of $\alpha = 0.05$, DS performs better than the SB approach. A similar evaluation may be done by considering the comparison of DS against CC and SS. The computed DS wins, ties and losses are (403, 23, 308) and (84, 0, 39), against CC and SS, respectively. In both cases, the null hypothesis is rejected. When compared against CC, the results are ($426 > 389.2$), while for SS, they are ($84 > 70.6$).

In addition to this simple non-parametric test, we performed a more comprehensive evaluation using the experiments on set S2.

Please cite this article as: A.S. Britto Jr. et al., Dynamic selection of classifiers—A comprehensive review, Pattern Recognition (2014), http://dx.doi.org/10.1016/j.patcog.2014.05.003

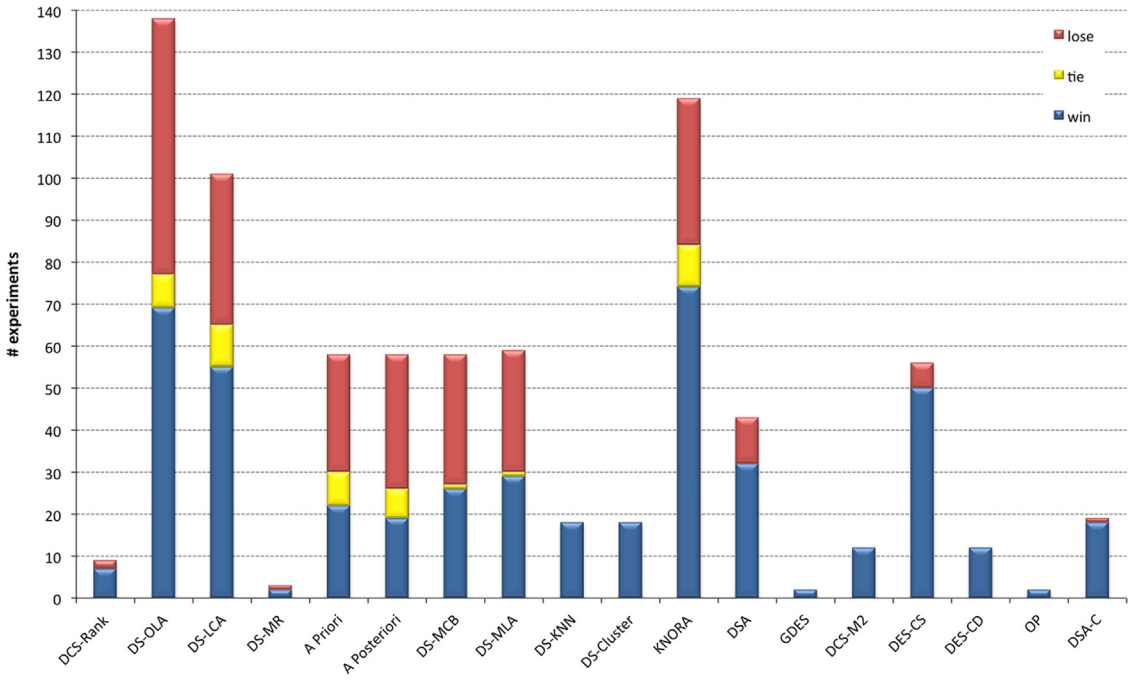

Fig. 3. Performance of the main DS methods in terms of number of wins, ties and losses with respect to the single best (SB) classifier.

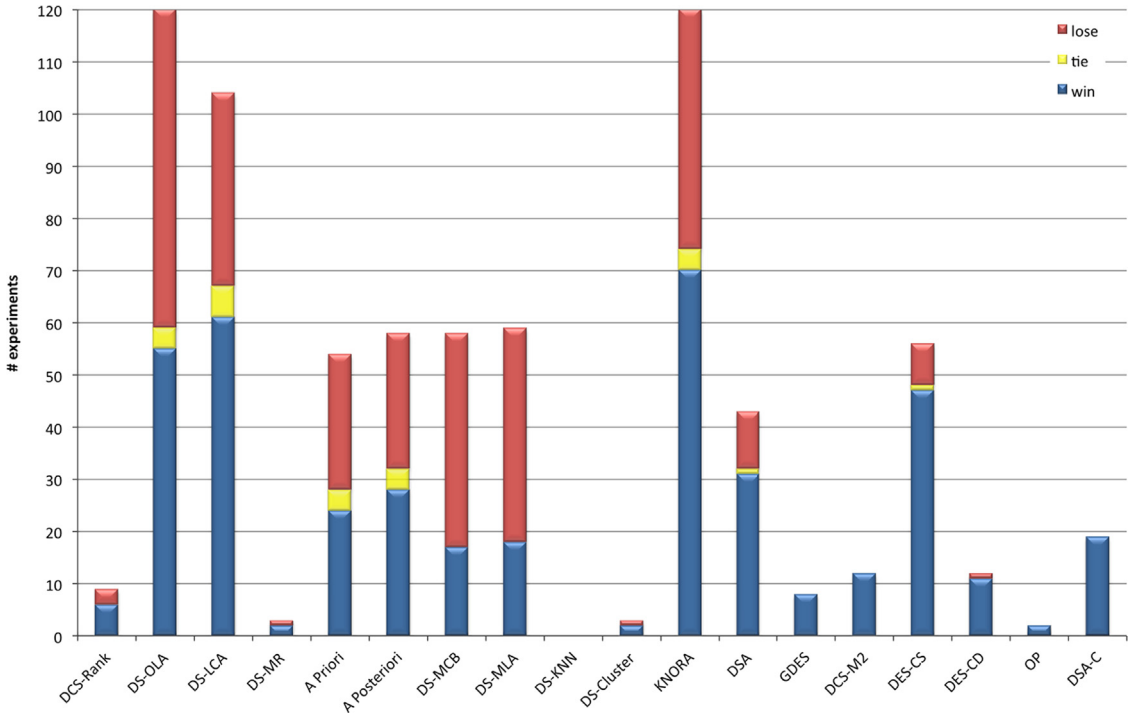

Fig. 4. Performance of the main DS methods in terms of number of wins, ties and losses with respect to the combination of all classifiers (CC) in the pool.

For each dataset, we found 10 experiments in which the SB, CC and DS approaches were compared. Except for the SH dataset for which only nine experiments were found. Unfortunately, not enough results were found to allow a comparison against the SS approach.

With the set of experiments for each dataset on hand, we reformulated our null-hypothesis (H_0) to state that the three classification approaches, SB, CC and DS, perform equally well. In other words, there is no significant difference between their results.

Friedman's test (F_t) [43] was performed and the results are shown in Table 3. Since we are comparing three approaches, the degree of freedom is 2. The level of significance (α) was defined as 0.05, while the corresponding critical value (p_α) is 5.99. In the same table, beyond the average rank of each classification approach, it is possible to find out whether or not the null-hypotheses is rejected.

As we may see, at a level of significance of $\alpha = 0.05$, there is enough evidence to conclude that, for the majority of the datasets, there is a significant difference in the accuracy among the three classification approaches, except for the three datasets (TE, SA and FE), where $F_t < p_\alpha$, and consequently, the null-hypothesis was not rejected.

Friedman's test only shows whether or not there is a significant difference between the three approaches, but it does not show

Please cite this article as: A.S. Britto Jr. et al., Dynamic selection of classifiers—A comprehensive review, Pattern Recognition (2014), http://dx.doi.org/10.1016/j.patcog.2014.05.003

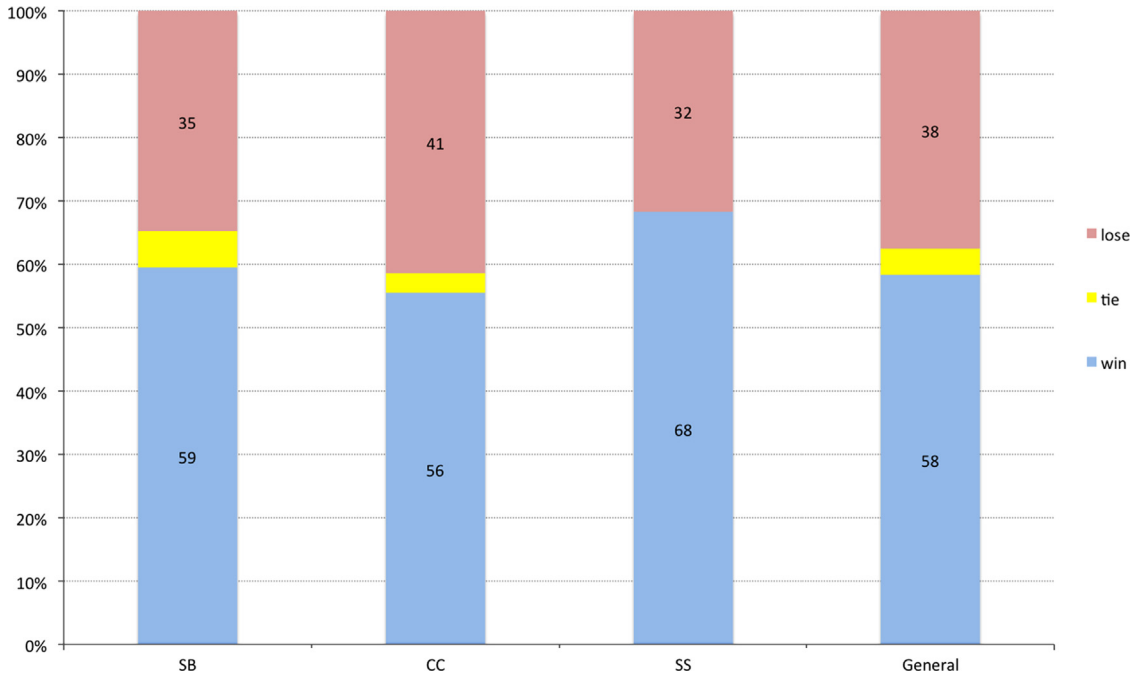

Fig. 5. Performance of the DS methods in terms of percentage of wins, ties and loses when compared to the single best classifiers (SB), the fusion of all classifiers (CC), static selection approach (SS), and any alternative solution (General).

Table 2
Datasets used and their main features.

Dataset	Source	# Classes	# Features	# Samples
Wine (W)	UCI	3	13	178
Liver-Disorders (LD)	UCI	2	6	345
Wisconsin Breast Cancer (WC)	UCI	2	30	569
Pima Diabetes (PD)	UCI	2	8	768
Image Segmentation (IS)	UCI	7	19	2310
Ship (SH)	Stalog	8	11	2545
Texture (TE)	Stalog	11	40	5500
Satimage (SA)	UCI	6	36	6435
Feltwell (FE)	Stalog	5	15	10,944
Letters (LR)	UCI	26	16	20,000
Nist Letter (NL)	NIST-SD19	26	132	67,192
Nist Digit (ND)	NIST-SD19	10	132	75,089

Table 3
The results of Friedman's test (F_t) for each dataset, considering the degree of freedom=2, the significance level α=0.05 and the critical value p_α = 5.99. The average ranks for the Single Best Classifier (SB), the Combination of All Classifiers (CC) and the Dynamic Selection (DS).

Dataset	Average ranks			F_t	Null-hypothesis
	SB	CC	DS		
W	2.050	2.900	1.050	17.15	Rejected
LD	2.100	2.750	1.150	12.95	Rejected
WC	1.750	2.900	1.350	12.95	Rejected
PD	1.900	2.900	1.200	24.33	Rejected
IS	1.750	2.800	1.450	10.05	Rejected
SH	2.889	1.667	1.444	10.88	Rejected
TE	2.300	2.200	1.500	3.80	Accepted
SA	2.200	2.100	1.700	0.67	Accepted
FE	1.800	2.600	1.600	5.60	Accepted
LR	2.700	2.100	1.200	11.40	Rejected
NL	3.000	1.950	1.050	19.05	Rejected
ND	3.000	1.800	1.200	16.80	Rejected

where the differences may be. To that end, we have performed a post hoc Nemenyi test [43], which compares the three approaches (SB, CC and DS) in a pairwise fashion. Fig. 6 shows a graphical representation of post hoc Nemenyi test results of the compared approaches for each dataset with the ranks given in Table 3. The numbers above the main line represent the average ranks, while CD is the critical difference for statistical significance. The methods with no significant difference are connected by lines. The CD is 1.10 for the SH dataset, and 1.05 for all other datasets. The performance of two classifiers is significantly different if the corresponding average ranks differ by at least the critical difference.

As already detected with Friedman's test for the FE, SA and TE datasets, there is no significant difference between the three approaches. On the other hand, for four datasets (SH, LR, ND and NL), we may observe a significant and positive difference between DS and SB; positive in the sense that DS is "better" than SB. For the same datasets, DS did not present a significant difference when compared to the CC approach. However, it is worth noting that we are not considering any other possible parameter of comparison between DS and CC, such as a reduction in terms of number of classifiers, for instance.

For another set of datasets (IS, WC, PD, LD and W), we observed no significant difference between DS and SB, but it could be seen that DS is significantly better than CC in those cases. This would suggest that the pools generated were composed of many weak classifiers.

Fig. 7 was obtained by plotting the differences between the DS and SB ranks used for the post hoc Nemenyi test. In this case, which is different from the usual graphical representation adopted in Fig. 6, the numbers above the main line represent the difference between ranks. Thus, high numbers mean a more significant difference between DS and SB. The objective is to show a graphical representation of the impact of the DS for each dataset when compared with the corresponding SB approach. As can be seen, the best impact was observed for the dataset NL (1.95) and the worst case was the dataset FE (0.2). In addition, this figure shows that different base classifiers were used in the experiments.

Please cite this article as: A.S. Britto Jr. et al., Dynamic selection of classifiers—A comprehensive review, Pattern Recognition (2014), http://dx.doi.org/10.1016/j.patcog.2014.05.003

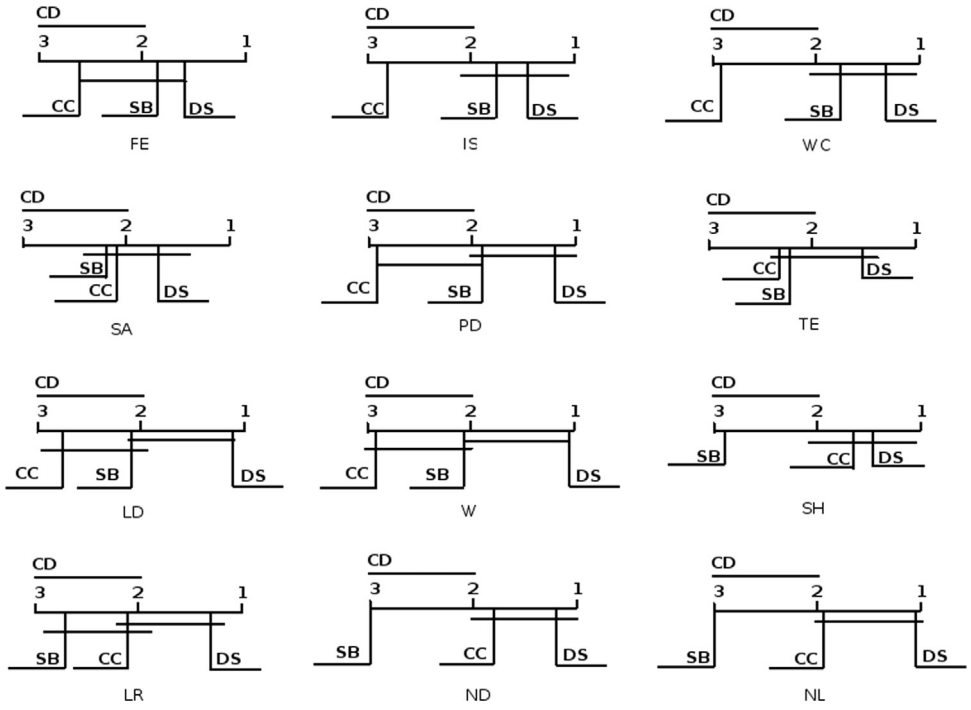

Fig. 6. Graphical representation of post hoc Nemenyi test results of compared methods for each dataset with ranks given in Table 3. For each dataset, the numbers on the main line represent the average ranks and the CD is the critical difference for statistical significance. Methods with no significant difference are connected by additional lines.

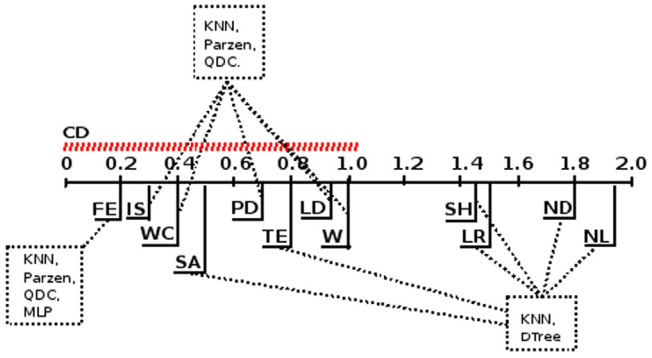

Fig. 7. The differences between the DS and SB ranks. For each dataset, the numbers on the main line represent the rank difference between DS and SB approaches. The CD is the critical difference for statistical significance.

From the results observed, we may confirm that, in general, there exists a significant difference between the performance of DS, SB, CC and SS. The results of the sign test have shown that. In addition, Friedman's test showed that in 9 from 12 datasets there is a significant difference between DS, SB and CC. A deeper pairwise analysis using the post hoc Nemenyi test, showed that in four of the remaining nine datasets, there was a significant performance contribution using the DS approach.

4.2. Classification difficulty

This section describes the second step of our methodology. The objective is to answer our second research question as to whether or not there is a relation between the classification complexity and the observed DS performance.

A first attempt to empirically characterize the DS approach as an interesting alternative to deal with complex problems was carried out in [17]. With this in mind, the authors divided the large digit dataset available on NIST-SD19 into subsets of different sizes to create five scenarios, varying from few samples for training (5000) to a large training set (180,000). In their experiments, two monolithic classifiers based on SVM and MLP were compared against their DS method. They observed that when enough data is available, the trained monolithic classifiers perform better than the proposed DS method. Thus, they suggested that the DS approach is more suitable when enough data is not available to represent the whole variability of the learned pattern.

Inspired by that observation, we decided to go further by investigating evidence of a clear correlation between the performance of DS methods and the classification difficulty. We then implemented a set of complexity measures for classification problems [3], composed of two measures of overlap between single feature values (F1 and F2), two measures of separability of classes (N2 and N3) and one measure related to the dimensionality of the dataset (T2). The measures used are described below based on their generalization to problems with multiple classes, as in [4].

1. *Fisher's Discriminant Ratio* (F 1): this well-known measure of class overlapping is calculated over each single feature dimension as denoted in Eq. (12), where M is the number of classes and μ is the overall mean, while n_i, μ_i and s_j^i are the number of samples, the mean and the jth sample of the class i, respectively. In this generalization of F1, Ψ is the Euclidian distance. A high F1 value indicates the presence of discriminating features and hence a classification problem easier.

$$F1 = \frac{\sum_{i=1}^{M} n_i \cdot \delta(\mu, \mu_i)}{\sum_{i=1}^{M} \sum_{i=1}^{n_i} \delta(s_j^i, \mu_i)} \tag{12}$$

2. *Volume of Overlap Region* (F 2): this measure conducts a pairwise calculation of the overlap between the conditional distribution of classes. As can be observed in Eq. (13), the overlap, considering two classes, c_i and c_j, and a T-dimension feature space, is calculated by finding the minimum and maximum values of each feature f_k for both classes. The ratio between the range of the feature values for each class is normalized by the length of the total range considering both

classes. The overlap region is estimated as the product of the normalized ratio obtained for all features. The generalization for multiple classes considers the sum of the overlapped regions calculated for each pair of classes. A small overlap (F2) value suggests a classification problem easier.

$$F2 = \sum_{(c_i,c_j)} \prod_{k=1}^{T} \frac{\min[\max(f_k,c_i),\max(f_k,c_j)] - \max[\min(f_k,c_i),\min(f_k,c_j)]}{\max[\max(f_k,c_i),\max(f_k,c_j)] - \min[\min(f_k,c_i),\min(f_k,c_j)]} \quad (13)$$

3. *Non-parametric Separability of Classes* (*N* 2, *N* 3): the first measure, referred to as N2 in the literature, compares the intraclass dispersion with the interclass separability, as denoted in Eq. (14). For this purpose, let $\eta_1^{intra}(s_i)$ and $\eta_1^{inter}(s_i)$ denote the intra- and inter-class nearest neighbors of the sample s_i, while Ψ represents the Euclidian distance. As can be observed, N2 calculates the ratio between the intra- and the inter-class dispersions. A small N2 value suggests high separability, and consequently, a classification problem easier. The second measure of separability (N3) is related to the estimated error rate of the 1-NN rule by using the leaving-one-out scheme.

$$N2 = \frac{\sum_i^N \delta(\eta_1^{intra}(s_i), s_i)}{\sum_i^N \delta(\eta_1^{inter}(s_i), s_i)} \quad (14)$$

4. *Density per Dimension* (*T* 2): this measure describes the density of spatial distributions of samples by computing the average number of instances per dimension. Referred to as T2 in the literature, it is calculated as shown in Eq. (15), where *N* and *T* are the number of samples and features, respectively, of a classification problem. Similar to F1, a high T2 value suggests a classification problem easier.

$$T2 = \frac{N}{T} \quad (15)$$

After implementing the previously described complexity measures, they were applied to the datasets described in Table 2. The value of each measure for these datasets is shown in Table 4.

As can be seen, the same problem may be taken as difficult with respect to one measure, but easy with respect to another. The reason is that the different measures consider different aspects of the classification problems. However, some interesting analysis may be done when they are combined. For instance, let us consider the measure values related to the Wine (W), Liver-Disorders (LD) and Texture (TE) datasets. Based on the class overlapping calculated over each feature dimension (measure F1), W should be considered as an easy problem. Its value for this measure is the second higher. However, the values of measures N3 and T2 for the same dataset show the contrary. Based on N3, the error rate of the NN classifier is high for W (close to 23%), while the number of samples per dimension is very low (around 13). Despite the fact that we have a small overlap of the feature range values among classes shown by the value F1, the dataset W has few samples, making it more difficult to be learned. For the LD dataset, all measures seem to agree on its complexity: it is in fact the most complex problem among those in Table 4. On the other hand, TE is a very easy problem.

Table 4
Results of the complexity measures for each dataset: ↑ means the higher easier, ↓ means the lower easier.

Dataset	F1↑	F2↓	N2↓	N3↓	T2↑
W	2.362	6.120E−05	0.018	0.230	13.692
LD	0.017	7.320E−02	0.853	0.377	57.500
WC	1.118	5.683E−11	0.031	0.084	18.967
PD	0.032	2.515E−01	0.838	0.320	96.000
IS	0.938	1.653E−04	0.071	0.033	121.579
SH	0.706	6.687E−02	0.293	0.095	231.364
TE	4.064	5.058E−06	0.127	0.009	135.500
SA	2.060	3.754E−04	0.215	0.091	178.750
FE	1.206	6.722E−02	0.107	0.011	729.600
LR	0.479	2.162E+00	0.228	0.038	1250.000
NL	0.642	9.080E−30	0.535	0.068	509.030
ND	0.626	1.257E−32	0.327	0.015	568.856

Although some interesting assumptions may be made based on these results, our question here is whether or not there is a relation between the observed DS performance and the classification difficulty.

In this perspective, except for the F2 measure, we carried out an analysis in which the complexity measures were combined in a pairwise fashion. The reason for excluding F2 was that it reflects the problem dimensionality, making it difficult to compare problems having overly different numbers of features. Fig. 8 plots the pairwise combinations F1 × N2, F1 × N3, F1 × T2, N2 × N3, N2 × T2, and N3 × T2. The datasets represented by red markers (SH, LR, ND and NL) in the plots are those for which we observed some significant contribution of the DS when compared to the corresponding SB approach. As can be seen, F1 × N2, F1 × N3 and F1 × T2 showed some interesting results. The datasets mentioned show low F1 values (difficult), in addition to low values of N2 (easy) and N3 (easy). We see this in the plot corresponding to F1 × N3. Moreover, in the plot corresponding to F1 × T2, it can be seen that the datasets where DS did not show a significant contribution present a low T2 value (difficult), except for one outlier (FE dataset).

The significant DS contribution observed for the datasets SH, LR, ND and NL may be explained using F1, N3 and T2. For these datasets, F1 suggests a high difficulty related to the overlap among the ranges of the feature values per class ($F1 \leq 0.8$). On the other hand, N3 and T2 suggest that they are easy problems. A low N3 means an easy problem for an NN classifier, since this measure represents the NN error rate using a leave-one-out strategy. A high T2 implies that there are more samples to deal with the problem variability. Thus, a low F1 combined with a low N3 and high T2 is an interesting observation when the goal is to use a DS approach.

As with SH, LR, ND and NL, the PD and LD datasets show a very low F1. However, they show a very high N3 and a very low T2. This means that they are difficult for the three measures. For this type of dataset, the DS approach seems unsuitable. The same assumption may be made for datasets with high F1 values.

Fig. 9 presents the evaluated datasets plotted in the space formed by the measures F1, N3 and T2. The SH, LR, ND and NL datasets are represented by red markers. It is possible to see that these datasets appear close to the origin in terms of the F1 and N3 axes, and usually present a high T2 value.

Thus, from this analysis, we may conclude that a relation exists between the data complexity and the observed contribution of the DS approach. In addition, we can also conclude that this relation is based on some intrinsic aspects of the classification problem, more than just the size of the problem (number of samples, classes and features).

5. Conclusion and future works

In this paper, we have presented the state of the art of DS methods, proposing a taxonomy for them. In addition, we have revisited the main basic concepts related to DS and presented the algorithms of some key methods available in the literature. This review has shown that different selection schemes have been

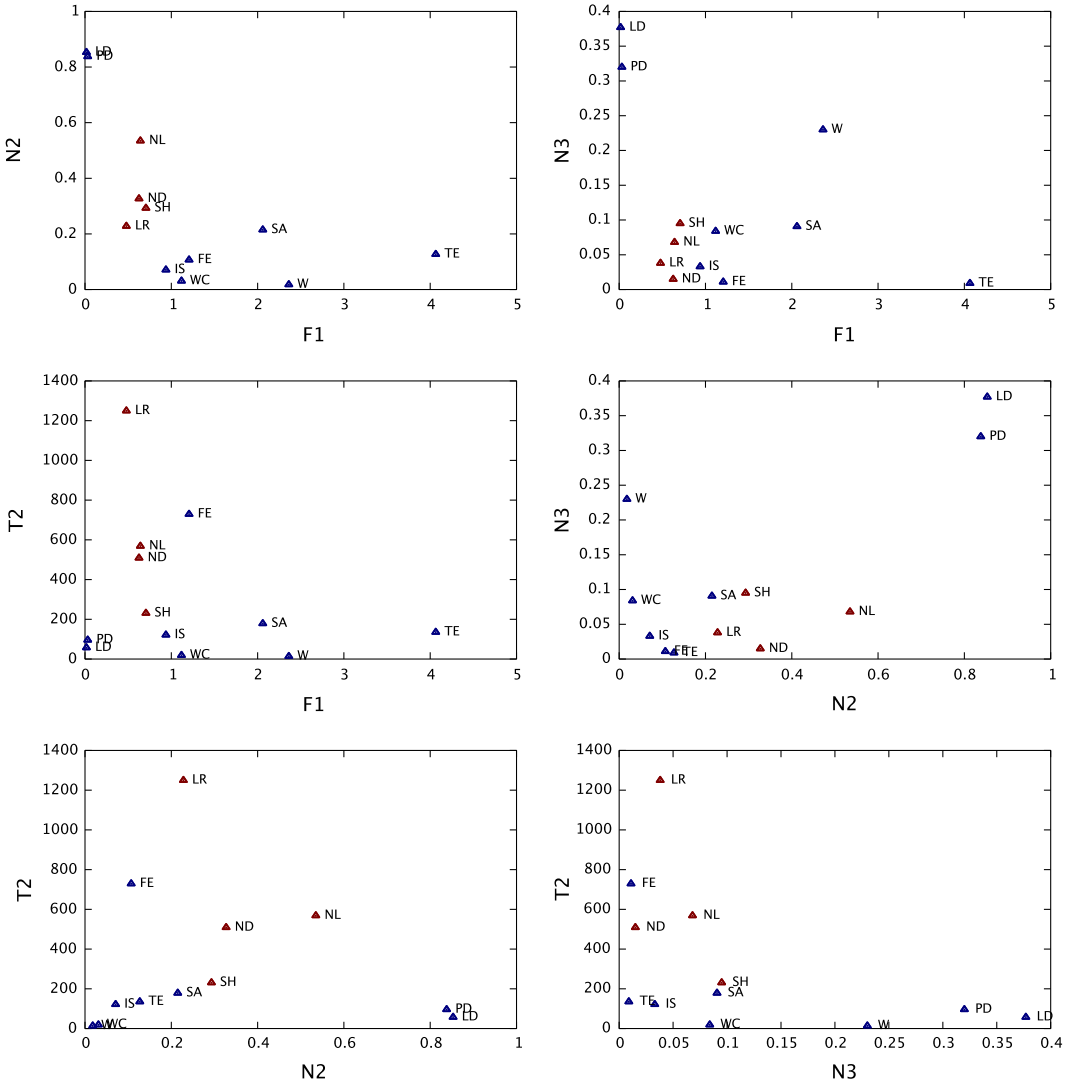

Fig. 8. Pairwise combination of the complexity measures F1, N2, N3 and T2, considering the datasets presented in Table 4.

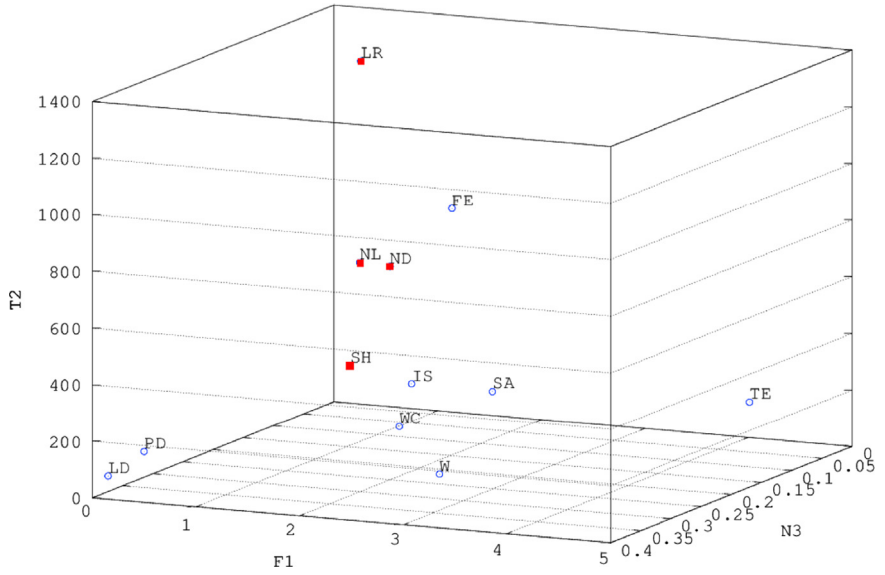

Fig. 9. 3D graphical representation of the datasets based on the measures F1, N3 and T2. The SH, LR, ND and NL are in red. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

Please cite this article as: A.S. Britto Jr. et al., Dynamic selection of classifiers—A comprehensive review, Pattern Recognition (2014), http://dx.doi.org/10.1016/j.patcog.2014.05.003

proposed, and basically differ in terms of the source of information used to evaluate the competence of the available classifiers in the pool for a given unknown sample.

As expected, the study performed does not allow us to point out the best DS method. On the contrary, it is seen that there is no evidence that one specific method may surpass all the others for any classification problem. However, it can be observed that simpler selection schemes (KNORA and DS-LCA) may provide similar, or sometimes even better, classification performances than the sophisticated ones (GDES, DSA and DSA-C).

Despite the importance of the literature review presented, the research questions addressed in this paper were related to when DS is applied. A further analysis of the reported results of the studied DS methods showed that for some classification problems, the DS contribution is statistically significant. In addition, it showed that there is some evidence of a relation between the DS performance contribution and the corresponding complexity of the classification problem. Thus, we may conclude that it is possible to predict when to use or not DS.

As observed, the DS has shown better results for classification problems presenting low F1 values, combined with high N3 and T2 values. Such an observed relation between the problem complexity and the DS contribution confirms the Ho et al. expectation [3]. They suggested that complexity measures may be used as a guide for static or dynamic selection of classifiers. In our case, we suggest that they may be used to determine when to apply DS. However, as also cautioned in their work, the extrapolation of the observations must be done with extreme care. The reason for this is that the observations are based on a tiny set of classification problems. Further work can be done, considering these three complexity measures and a huge set of classification problems in order to model a meta-classifier dedicated to determining whether or not to use a DS approach for a specific classification problem.

Conflict of interest

None declared.

Acknowledgments

This research has been supported by the Brazilian National Council for Scientific and Technological Development (CNPq) and by the Research Foundation of the Parana state (Fundação Araucária).

References

[1] L.K. Hansen, P. Salamon, Neural network ensembles, IEEE Trans. Pattern Anal. Mach. Intell. 12 (October (10)) (1990) 993–1001.
[2] T.G. Dietterich, Ensemble methods in machine learning, in: Multiple Classifier Systems, Lecture Notes in Computer Science, vol. 1857, Springer, Berlin, Heidelberg, 2000, pp. 1–15.
[3] T.K. Ho, M. Basu, Complexity measures of supervised classification problems, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002) 289–300.
[4] J.S. Sanchez, R.A. Mollineda, J.M. Sotoca, An analysis of how training data complexity affects the nearest neighbor classifiers, Pattern Anal. Appl. 10 (July (3)) (2007) 189–201.
[5] G.D.C. Cavalcanti, T.I. Ren, B.A. Vale, Data complexity measures and nearest neighbor classifiers: a practical analysis for meta-learning, in: IEEE 24th International Conference on Tools with Artificial Intelligence (ICTAI), 2012, vol. 1, 2012, pp. 1065–1069.
[6] L.I. Kuncheva, C.J. Whitaker, Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy, Mach. Learn. 51 (May (2)) (2003) 181–207.
[7] G. Kumar, K. Kumar, The use of artificial-intelligence-based ensembles for intrusion detection: a review, Appl. Comput. Int. Soft Comput. 2012 (2012) 1–20.
[8] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.
[9] R.E. Schapire, Y. Freund, P. Bartlett, and W.S. Lee, Boosting the margin: a new explanation for the effectiveness of voting methods, in: Proceedings of 14th International Conference on Machine Learning, Nashville, USA, 1997, pp. 322–330.
[10] T.K. Ho, The random subspace method for constructing decision forests, IEEE Trans. Pattern Anal. Mach. Intell. 20 (8) (1998) 832–844.
[11] K. Woods, W.P. Kegelmeyer Jr, K. Bowyer, Combination of multiple classifiers using local accuracy estimates, IEEE Trans. Pattern Anal. Mach. Intell. 19 (4) (1997) 405–410.
[12] M. Sabourin, A. Mitiche, D. Thomas, G. Nagy, Classifier combination for hand-printed digit recognition, in: 1993 Proceedings of the Second International Conference on Document Analysis and Recognition, 1993, pp. 163–166.
[13] G. Giacinto, F. Roli, Methods for dynamic classifier selection, in: 1999 Proceedings of 10th International Conference on Image Analysis and Processing, 1999, pp. 659–664.
[14] M. Kurzynski, T. Woloszynski, R. Lysiak, On two measures of classifier competence for dynamic ensemble selection—experimental comparative analysis, in: 2010 International Symposium on Communications and Information Technologies (ISCIT), 2010, pp. 1108–1113.
[15] G. Giacinto, F. Roli, Dynamic classifier selection based on multiple classifier behaviour, Pattern Recognit. 34 (2001) 1879–1881.
[16] A. Nabiha, F. Nadir, New dynamic ensemble of classifiers selection approach based on confusion matrix for arabic handwritten recognition, in: 2012 International Conference on Multimedia Computing and Systems (ICMCS), 2012, pp. 308–313.
[17] P.R. Cavalin, R. Sabourin, C.Y. Suen, Dynamic selection approaches for multiple classifier systems, Neural Comput. Appl. 22 (3–4) (2013) 673–688.
[18] L.I. Kuncheva, J.J. Rodriguez, Classifier ensembles with a random linear oracle, IEEE Trans. Knowl. Data Eng. 19 (4) (2007) 500–508.
[19] A.H.R. Ko, R. Sabourin, A.S. Britto Jr., From dynamic classifier selection to dynamic ensemble selection, Pattern Recognit. 41 (5) (2008) 1718–1731.
[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 (1) (2003) 63–68.
[21] A. Santana, R.G.F. Soares, A.M.P. Canuto, M.C.P. de Souto, A dynamic classifier selection method to build ensembles using accuracy and diversity, in: 2006 Ninth Brazilian Symposium on Neural Networks, SBRN '06, 2006, pp. 36–41.
[22] R. Lysiak, M. Kurzynski, T. Woloszynski, Probabilistic approach to the dynamic ensemble selection using measures of competence and diversity of base classifiers, in: Emilio Corchado, Marek Kurzynski, Michal Wozniak (Eds.), Hybrid Artificial Intelligent Systems, Lecture Notes in Computer Science, vol. 6679, Springer, Berlin, Heidelberg, 2011, pp. 229–236.
[23] T.K. Ho, J.J. Hull, S.N. Srihari, Decision combination in multiple classifier systems, IEEE Trans. Pattern Anal. Mach. Intell. 16 (1) (1994) 66–75.
[24] E.M. dos Santos, R. Sabourin, P. Maupin, A dynamic overproduce-and-choose strategy for the selection of classifier ensembles, Pattern Recognit. 41 (October (10)) (2008) 2993–3009.
[25] J. Xiao, C. He, Dynamic classifier ensemble selection based on GMDH, in: 2009 International Joint Conference on Computational Sciences and Optimization, CSO 2009, vol. 1, 2009, pp. 731–734.
[26] A.K. Jain, R.P.W. Duin, J. Mao, Statistical pattern recognition: a review, IEEE Trans. Pattern Anal. Mach. Intell. 22 (1) (2000) 4–37.
[27] J. Kittler, M. Hatef, R.P.W. Duin, J. Matas, On combining classifiers, IEEE Trans. Pattern Anal. Mach. Intell. 20 (1998) 226–239.
[28] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, G.E. Hinton, Adaptative mixtures of local experts, Neural Comput. 3 (1) (1991) 79–87.
[29] .Jacobs Robert A, Bias/variance analyses of mixtures-of-experts architectures, Neural Comput. 9 (February (2)) (1997) 369–383.
[30] L. Kuncheva, Combining Pattern Classifiers, Methods and Algorithms, Wiley, New York, 2004.
[31] L.A. Rastrigin, R.H. Erenstein, Method of collective recognition, Energoizdat, 1981 (in Russian).
[32] T. Woloszynski, M. Kurzynski, A probabilistic model of classifier competence for dynamic ensemble selection, Pattern Recognit. 44 (2011) 2656–2668 (Semi-Supervised Learning for Visual Content Analysis and Understanding).
[33] Y.S. Huang, C.Y. Suen, A method of combining multiple experts for the recognition of unconstrained handwritten numerals, IEEE Trans. Pattern Anal. Mach. Intell. 17 (1) (1995) 90–94.
[34] P.R. Cavalin, R. Sabourin, C.Y. Suen, Dynamic selection of ensembles of classifiers using contextual information, in: Proceedings of the 9th International Conference on Multiple Classifier Systems, Springer-Verlag, Berlin, Heidelberg, 2010, MCS'10, pp. 145–154.
[35] E.M. dos Santos, R. Sabourin, P. Maupin, Ambiguity-guided dynamic selection of ensemble of classifiers, in: 2007 10th International Conference on Information Fusion, 2007, pp. 1–8.
[36] A.G. Ivakhnenko, Heuristic self-organization in problems of engineering cybernetics, Automatica 6 (2) (1970) 207–219.
[37] P.C. Smits, Multiple classifier systems for supervised remote sensing image classification based on dynamic classifier selection, IEEE Trans. Geosci. Remote Sens. 40 (4) (2002) 801–813.
[38] L.I. Kuncheva, Switching between selection and fusion in combining classifiers: an experiment, Trans. Syst. Man Cybern. Part B 32 (April (2)) (2002) 146–156.
[39] J. Xiao, L. Xie, C. He, X. Jiang, Dynamic classifier ensemble model for customer classification with imbalanced class distribution, Expert Syst. Appl. 39 (February (3)) (2012) 3668–3675.

[40] L. Batista, E. Granger, R. Sabourin, Dynamic selection of generative-discriminative ensembles for off-line signature verification, Pattern Recognit. 45 (April (4)) (2012) 1326–1340.
[41] D.H. Wolpert, The lack of a priori distinctions between learning algorithms, Neural Comput. 8 (October (7)) (1996) 1341–1390.
[42] G. Giacinto, F. Roli, Adaptive selection of image classifiers, in: Alberto Bimbo (Ed.), Image Analysis and Processing, Lecture Notes in Computer Science, vol. 1310, Springer, Berlin, Heidelberg, 1997, pp. 38–45.
[43] N. Japkowicz, M. Shah, Evaluating Learning Algorithms: A Classification Perspective, Cambridge University Press, New York, NY, USA, 2011.

A.S. Britto Jr. received M.Sc. degree in Industrial Informatics from the Centro Federal de Educação Tecnológica do Paraná (CEFET-PR, Brazil) in 1996, and Ph.D. degree in Computer Science from the Pontifícia Universidade Católica do Paraná (PUCPR, Brazil) in 2001. In 1989, he joined the Informatics Department of the Universidade Estadual de Ponta Grossa (UEPG, Brazil). In 1995, he also joined the Computer Science Department of the Pontifícia Universidade Católica do Paraná (PUCPR) and, in 2001, the Post-graduate Program in Informatics (PPGIa). His research interests are in the areas of document analysis and handwriting recognition.

R. Sabourin joined in 1977 the Physics Department of the Montreal University where he was responsible for the design, experimentation and development of scientific instrumentation for the Mont Mégantic Astronomical Observatory. His main contribution was the design and the implementation of a microprocessor-based fine tracking system combined with a low-light level CCD detector. In 1983, he joined the staff of the École de Technologie Supérieure, Université du Québec, in Montréal where he cofounded the Department of Automated Manufacturing Engineering where he is currently Full Professor and teaches Pattern Recognition, Evolutionary Algorithms, Neural Networks and Fuzzy Systems. In 1992, he joined also the Computer Science Department of the Pontifícia Universidade Católica do Paraná (Curitiba, Brazil) where he was co-responsible for the implementation in 1995 of a master program and in 1998 a Ph.D. program in applied computer science. Since 1996, he is a senior member of the Centre for Pattern Recognition and Machine Intelligence (CENPARMI, Concordia University). Dr. Sabourin is the author (and co-author) of more than 260 scientific publications including journals and conference proceedings. He was a co-chair of the program committee of CIFED'98 (Conférence Internationale Francophone sur l'Écrit et Le Document, Québec, Canada) and IWFHR'04 (9th International Workshop on Frontiers in Handwriting Recognition, Tokyo, Japan). He was nominated as a Conference co-chair of ICDAR'07 (9th International Conference on Document Analysis and Recognition) that has been held in Curitiba, Brazil, in 2007. His research interests are in the areas of handwriting recognition, signature verification, intelligent watermarking systems and bio-cryptography.

L.E.S. Oliveira received the B.S. degree in Computer Science from UnicenP, Curitiba, PR, Brazil, the M.Sc. degree in electrical engineering and industrial informatics from the Centro Federal de Educacao Tecnologica do Parana (CEFET-PR), Curitiba, PR, Brazil, and Ph.D. degree in Computer Science from Ecole de Technologie Superieure, Universite du Quebec in 1995, 1998, and 2003, respectively. From 2004 to 2009 he was a professor of the Computer Science Department at Pontifical Catholic University of Parana, Curitiba, PR, Brazil. In 2009 he joined the Federal University of Parana, Curitiba, PR, Brazil, where he is a professor of the Department of Informatics. His current interests include Pattern Recognition, Neural Networks, Image Analysis, and Evolutionary Computation.

Please cite this article as: A.S. Britto Jr. et al., Dynamic selection of classifiers—A comprehensive review, Pattern Recognition (2014), http://dx.doi.org/10.1016/j.patcog.2014.05.003