Analyzing Dynamic Ensemble Selection Techniques Using Dissimilarity Analysis

Rafael M. O. Cruz, Robert Sabourin, and George D. C. Cavalcanti

École de Technologie Supérieure, Université du Québec,
1100 Notre-Dame Ouest, Montréal, Canada
Centro de Informática, Universidade Federal de Pernambuco
Recife, Brazil
{cruz,Robert.Sabourin}@livia.etsmtl.ca
gdcc@cin.ufpe.br
http://www.livia.etsmtl.ca

Abstract. In Dynamic Ensemble Selection (DES), only the most competent classifiers are selected to classify a given query sample. A crucial issue faced in DES is the definition of a criterion for measuring the level of competence of each base classifier. To that end, a criterion commonly used is the estimation of the competence of a base classifier using its local accuracy in small regions of the feature space surrounding the query instance. However, such a criterion cannot achieve results close to the performance of the Oracle, which is the upper limit performance of any DES technique. In this paper, we conduct a dissimilarity analysis between various DES techniques in order to better understand the relationship between them and as well as the behavior of the Oracle. In our experimental study, we evaluate seven DES techniques and the Oracle using eleven public datasets. One of the seven DES techniques was proposed by the authors and uses meta-learning to define the competence of base classifiers based on different criteria. In the dissimilarity analysis, this proposed technique appears closer to the Oracle when compared to others, which would seem to indicate that using different bits of information on the behavior of base classifiers is important for improving the precision of DES techniques. Furthermore, DES techniques, such as LCA, OLA, and MLA, which use similar criteria to define the level of competence of base classifiers, are more likely to produce similar results.

Keywords: Ensemble of classifiers, dynamic ensemble selection, dissimilarity analysis, meta-learning.

1 Introduction

In recent years, ensembles of Classifiers (EoC) have been widely studied as an alternative for increasing efficiency and accuracy in pattern recognition [1,2]. Classifier ensembles involve two basic approaches, namely, classifier fusion and dynamic ensemble selection. With classifier fusion approaches, each classifier in the ensemble is used, and their outputs are aggregated to give the final prediction. However, such techniques [1,3] present two main problems: they are based on the assumption that the base classifiers commit independent errors, which rarely occurs to find in real pattern recognition applications.

N. El Gayar et al. (Eds.): ANNPR 2014, LNAI 8774, pp. 59–70, 2014.

On the other hand, Dynamic Ensemble Selection (DES) techniques [4] rely on the assumption that each base classifier[1] is an expert in a different local region of the feature space. DES techniques work by measuring the level of competence of each base classifier, considering each new test sample. Only the most competent(s) classifier(s) is(are) selected to predict the class of a new test sample. Hence, the key issue in DES is defining a criterion for measuring the level of competence of a base classifier. Most DES techniques [5,6,7,8] use estimates of the classifier's local accuracy in small regions of the feature space surrounding the query instance as search criteria to carry out the ensemble selection. However, in our previous work [7], we demonstrated that this criterion is limited, and cannot achieve results close to the performance of the Oracle, which represents the best possible result of any combination of classifiers [2]. In addition, as reported by Ko et al. [5], addressing the behavior of the Oracle is much more complex than applying a simple neighborhood approach, and the task of figuring out its behavior based merely on the pattern feature space is not an easy one.

To tackle this issue, in [9] we proposed a novel DES framework in which multiple criteria regarding the behavior of a base classifier are used to compute its level of competence. In this paper, we conduct a dissimilarity analysis between different DES techniques in order to better understand their relationship. The analysis is performed based on the difference between the levels of competence of a base classifier estimated by the criterion embedded in each DES technique. All in all, we compare the DES criteria of seven state-of-the-art DES techniques, including our proposed meta-learning framework. In addition, we also formalize the Oracle as an ideal DES technique (i.e., a DES scheme which selects only the classifiers of the pool that predict the correct class for the query instance) to be used in the analysis.

The dissimilarities between different DES criteria are computed in order to generate a dissimilarity matrix, which is then, used to project each DES technique onto a two-dimensional space, called the Classifier Projection Space (CPS) [10]). In the CPS, each DES technique is represented by a point, and the distance between two points corresponds to their degree of dissimilarity. Techniques that appear close together present similar behavior (i.e., they are more likely to produce the same results), while those appearing far apart in the two-dimensional CPS can be considered different. Thus, a spatial relationship is achieved between different techniques. The purpose of the dissimilarity analysis is twofold: to understand the relationship between different DES techniques (i.e., whether or not the criteria used by DES techniques present a similar behavior), and in order to determine which DES technique presents a behavior that is closer to the behavior of the ideal DES scheme (Oracle).

This paper is organized as follows: Section 2 introduces the DES techniques from the literature that are used in the dissimilarity analysis. The proposed meta-learning framework is described in Section 3. Experiments are conducted in Section 4, and finally, our conclusion is presented in the last section.

[1] The term base classifier refers to a single classifier belonging to an ensemble or a pool of classifiers.

2 Dynamic Ensemble Selection Techniques

The goal of dynamic selection is to find an ensemble of classifiers, $C' \subset C$ containing the best classifiers to classify a given test sample $\mathbf{x}_j$. This is different from static selection, where the ensemble of classifiers C' is selected during the training phase, and considering the global performance of the base classifiers over a validation dataset. In dynamic selection, the classifier competence is measured on-the-fly for each query instance $\mathbf{x}_j$.

The following DES techniques are described in this section: Overall Local Accuracy (OLA) [6], Local Classifier Accuracy (LCA) [6], Modified Local Accuracy (MLA) [8], KNORA-Eliminate [5], K-Nearest Output Profiles (KNOP) [11] and Multiple Classifier Behavior (MCB) [12].

For the definitions below, let $\theta_j = \{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$ be the region of competence of the test sample $\mathbf{x}_j$ (K is the size of the region of competence), defined on the validation data, c_i a base classifier from the pool $C = \{c_1, \ldots, c_M\}$ (M is the size of the pool), w_l the correct label of $\mathbf{x}_j$ and $\delta_{i,j}$ the level of competence of c_i for the classification of the input instance $\mathbf{x}_j$.

Overall Local Accuracy (OLA)

In this method, the level of competence $\delta_{i,j}$ of a base classifier c_i is simply computed as the local accuracy achieved by c_i for the region of competence θ_j. (Equation 1). The classifier with the highest level of competence $\delta_{i,j}$ is selected.

$$\delta_{i,j} = \sum_{k=1}^{K} P(w_l \mid \mathbf{x}_k \in w_l, c_i) \tag{1}$$

Local Classifier Accuracy (LCA)

This rule is similar to the OLA, with the only difference being that the local accuracy of c_i is estimated with respect to the output classes; w_l (w_l is the class assigned for $\mathbf{x}_j$ by c_i) for the whole region of competence, θ_j (Equation 2). The classifier with the highest level of competence $\delta_{i,j}$ is selected.

$$\delta_{i,j} = \frac{\sum_{\mathbf{x}_k \in w_l} P(w_l \mid \mathbf{x}_k, c_i)}{\sum_{k=1}^{K} P(w_l \mid \mathbf{x}_k, c_i)} \tag{2}$$

Modified Local Accuracy (MLA)

The MLA technique works similarly to the LCA. The only difference is that each instance $\mathbf{x}_k$ belonging to the region of competence θ_j is weighted by its Euclidean distance to the query sample $\mathbf{x}_j$. The classifier with the highest level of competence $\delta_{i,j}$ is selected.

KNORA-Eliminate (KNORA-E)

Given the region of competence θ_j, only the classifiers that achieved a perfect score, considering the whole region of competence, are considered competent for the classification of $\mathbf{x}_j$. Thus, the level of competence $\delta_{i,j}$ is either "competent", $\delta_{i,j} = 1$ or "incompetent", $\delta_{i,j} = 0$. All classifiers considered competent are selected.

Multiple Classifier Behavior (MCB)

Given the query pattern $\mathbf{x}_j$, the first step is to compute its K-Nearest-Neighbors $\mathbf{x}_k$, $k = 1, \ldots, K$. Then, the output profiles of each neighbor $\tilde{\mathbf{x}}_k$ are computed and compared to the output profile of the test instance $\tilde{\mathbf{x}}_j$ according to a similarity metric $D_{OutProf}$. If $D_{OutProf} > threshold$, the pattern is removed from the region of competence. The level of competence $\delta_{i,j}$ is measured by the recognition performance of the base classifier c_i over the filtered region of competence. The classifier with the highest level of competence $\delta_{i,j}$ is selected.

K-Nearest Output Profiles (KNOP)

This rule is similar to the KNORA technique, with the only difference being that KNORA works in the feature space while KNOP works in the decision space using output profiles. First, the output profiles' transformation is applied over the input $\mathbf{x}_j$, giving $\tilde{\mathbf{x}}_j$. Next, the similarity between $\tilde{\mathbf{x}}_j$ and the output profiles from the validation set is computed and stored in the set ϕ_j. The level of competence $\delta_{i,j}$ of a base classifier c_i for the classification of $\mathbf{x}_j$ is defined by the number of samples in ϕ_j that are correctly classified by c_i.

Oracle

The Oracle is classically defined in the literature as a strategy that correctly classifies each query instance $\mathbf{x}_j$ if any classifier c_i from the pool of classifiers C predicts the correct label for $\mathbf{x}_j$. In this paper, we formalize the Oracle as the ideal DES technique which always selects the classifier that predicts the correct label $\mathbf{x}_j$ and rejects otherwise. The Oracle as a DES technique is defined in Equation 3:

$$\begin{cases} \delta_{i,j} = 1, & \text{if } c_i \text{ correctly classifies } \mathbf{x}_j \\ \delta_{i,j} = 0, & \text{otherwise} \end{cases} \tag{3}$$

In other words, the level of competence $\delta_{i,j}$ of a base classifier c_i is 1 if it predicts the correct label for $\mathbf{x}_j$, or 0 otherwise.

3 Dynamic Ensemble Selection Using Meta-Learning

A general overview of the proposed meta-learning framework is depicted in Figure 1. It is divided into three phases: Overproduction, Meta-training and Generalization. Each phase is detailed in the following sections.

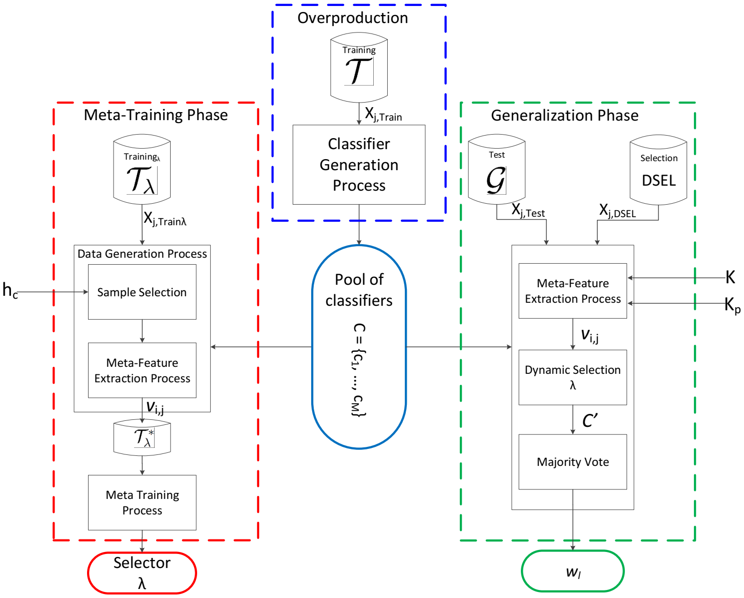


Fig. 1. Overview of the proposed framework. It is divided into three steps 1) Overproduction 2) Meta-training and 3) Generalization. [Adapted from [9]]

3.1 Overproduction

In this step, the pool of classifiers $C = \{c_1, \ldots, c_M\}$, where M is the pool size, is generated using the training dataset $\mathcal{T}$. The Bagging technique [13] is used in this work in order to build a diverse pool of classifiers.

3.2 Meta-Training

In this phase, the meta-features are computed and used to train the meta-classifier λ. As shown in Figure 1, the meta-training stage consists of three steps: sample selection, the meta-features extraction process and meta-training. A different dataset $\mathcal{T}_\lambda$ is used in this phase to prevent overfitting.

Sample Selection. We focus the training of λ on cases in which the extent of consensus of the pool is low. Thus, we employ a sample selection mechanism based on a threshold h_C, called the consensus threshold. For each $\mathbf{x}_{j,train_\lambda} \in \mathcal{T}_\lambda$, the degree of consensus of the pool, denoted by $H\left(\mathbf{x}_{j,train_\lambda}, C\right)$, is computed. If $H\left(\mathbf{x}_{j,train_\lambda}, C\right)$ falls below the threshold h_C, $\mathbf{x}_{j,train_\lambda}$ is passed down to the meta-features extraction process.

Meta-Features Extraction. In order to extract the meta-features, the region of competence of $\mathbf{x}_{j,train_\lambda}$, denoted by $\theta_j = \{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$ must be first computed. The region of competence is defined in the $\mathcal{T}_\lambda$ set using the K-Nearest Neighbor algorithm.

Then, $\mathbf{x}_j$ is transformed into an output profile, $\tilde{\mathbf{x}}_j$ by applying the transformation T, ($T : \mathbf{x}_j \Rightarrow \tilde{\mathbf{x}}_j$), where $\mathbf{x}_j \in \Re^D$ and $\tilde{\mathbf{x}}_j \in Z^M$ [11]. The output profile of a pattern $\mathbf{x}_j$ is denoted by $\tilde{\mathbf{x}}_j = \{\tilde{\mathbf{x}}_{j,1}, \tilde{\mathbf{x}}_{j,2}, \ldots, \tilde{\mathbf{x}}_{j,M}\}$, where each $\tilde{\mathbf{x}}_{j,i}$ is the decision yielded by the classifier c_i for $\mathbf{x}_j$. The similarity between $\tilde{\mathbf{x}}_j$ and the output profiles of the instances in $\mathcal{T}_\lambda$ is obtained through the Euclidean distance. The most similar output profiles are selected to form the set $\phi_j = \{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_{K_p}\}$, where each output profile $\tilde{\mathbf{x}}_k$ is associated with a label $w_{l,k}$. Next, for each base classifier $c_i \in C$, five sets of meta-features are calculated:

- f_1 - **Neighbors' hard classification:** First, a vector with K elements is created. For each instance $\mathbf{x}_k$, belonging to the region of competence θ_j, if c_i correctly classifies $\mathbf{x}_k$, the k-th position of the vector is set to 1, otherwise it is 0. Thus, K meta-features are computed.
- f_2 - **Posterior probability:** First, a vector with K elements is created. Then, for each instance $\mathbf{x}_k$, belonging to the region of competence θ_j, the posterior probability of c_i, $P(w_l \mid \mathbf{x}_k)$ is computed and inserted into the k-th position of the vector. Consequently, K meta-features are computed.
- f_3 - **Overall local accuracy:** The accuracy of c_i over the whole region of competence θ_j is computed and encoded as f_3.
- f_4 - **Output profiles classification:** First, a vector with K_p elements is generated. Then, for each member $\tilde{\mathbf{x}}_k$, belonging to the set of output profiles ϕ_j, if the label produced by c_i for $\mathbf{x}_k$ is equal to the label $w_{l,k}$ of $\tilde{\mathbf{x}}_k$, the k-th position of the vector is set to 1, otherwise it is 0. A total of K_p meta-features are extracted using output profiles.
- f_5 - **Classifier's Confidence:** The perpendicular distance between the input sample $\mathbf{x}_{j,train_\lambda}$ and the decision boundary of the base classifier c_i is calculated and encoded as f_5. f_5 is normalized to a $[0-1]$ range using the Min-max normalization.

A vector $v_{i,j} = \{f_1 \cup f_2 \cup f_3 \cup f_4 \cup f_5\}$ is obtained at the end of the process. It is important to mention that a different vector $v_{i,j}$ is generated for each base classifier c_i. If c_i correctly classifies $\mathbf{x}_{j,train_\lambda}$, the class attribute of $v_{i,j}$, $\alpha_{i,j} = 1$ (i.e., $v_{i,j}$ corresponds to the behavior of a competent classifier), otherwise $\alpha_{i,j} = 0$. $v_{i,j}$ is stored in the meta-features dataset (Figure 1).

Training. With the meta-features dataset, $\mathcal{T}_\lambda^*$, on hand, the last step of the meta-training phase is the training of the meta-classifier λ. The dataset $\mathcal{T}_\lambda^*$ is divided on the basis of 75% for training and 25% for validation. A Multi-Layer Perceptron (MLP) neural network with 10 neurons in the hidden layer is considered as the selector λ. The training process for λ is performed using the Levenberg-Marquadt algorithm, and is stopped if its performance on the validation set decreases or fails to improve for five consecutive epochs.

3.3 Generalization

Given an input test sample $\mathbf{x}_{j,test}$ from the generalization dataset $\mathcal{G}$, first, the region of competence θ_j and the set of output profiles ϕ_j, are calculated using the samples from the dynamic selection dataset D_{SEL} (Figure 1). For each classifier $c_i \in C$, the five subsets of meta-features are extracted, returning the meta-features vector $v_{i,j}$. Next, $v_{i,j}$ is passed down as input to the meta-classifier λ, which decides whether c_i is competent enough to classify $\mathbf{x}_{j,test}$. In this case, the posterior probability obtained by the meta-classifier λ is considered as the estimation of the level of competence $\delta_{i,j}$ of the base classifier c_i in relation to $\mathbf{x}_{j,test}$.

After each classifier of the pool is evaluated, the majority vote rule [2] is applied over the ensemble C', giving the label w_l of $\mathbf{x}_{j,test}$. Tie-breaking is handled by choosing the class with the highest a posteriori probability.

4 Experiments

We evaluated the generalization performance of the proposed technique using eleven classification datasets, nine from the UCI machine learning repository, and two artificially generated using the Matlab PRTOOLS toolbox[2]. The experiment was conducted using 20 replications. For each replication, the datasets were randomly divided on the basis of 25% for training ($\mathcal{T}$), 25% for meta-training $\mathcal{T}_\lambda$, 25% for the dynamic selection dataset (D_{SEL}) and 25% for generalization ($\mathcal{G}$). The divisions were performed while maintaining the prior probability of each class. The pool of classifiers was composed of 10 Perceptrons. The values of the hyper-parameters K, K_p and h_c were set as 7, 5 and 70%, respectively. They were selected empirically based on previous publications [7,9].

4.1 Results

Table 1. Mean and standard deviation results of the accuracy obtained for the proposed meta-learning framework and the DES systems in the literature. The best results are in bold. Results that are significantly better ($p < 0.05$) are underlined.

Database	Proposed	KNORA-E	MCB	LCA	OLA	MLA	KNOP	Oracle
Pima	**77.74(2.34)**	73.16(1.86)	73.05(2.21)	72.86(2.98)	73.14(2.56)	73.96(2.31)	73.42(2.11)	95.10(1.19)
Liver Disorders	**68.83 (5.57)**	63.86(3.28)	63.19(2.39)	62.24(4.01)	62.05(3.27)	57.10(3.29)	65.23(2.29)	90.07(2.41)
Breast Cancer	**97.41(1.07)**	96.93(1.10)	96.83(1.35)	97.15(1.58)	96.85(1.32)	96.66(1.34)	95.42(0.89)	99.13(0.52)
Blood Transfusion	**79.14(1.88)**	74.59(2.62)	72.59(3.20)	72.20(2.87)	72.33(2.36)	70.17(3.05)	77.54(2.03)	94.20(2.08)
Banana	**90.16(2.09)**	88.83(1.67)	88.17(3.37)	89.28(1.89)	89.40(2.15)	80.83(6.15)	85.73(10.65)	94.75(2.09)
Vehicle	**82.50(2.07)**	81.19(1.54)	80.20(4.05)	80.33(1.84)	81.50(3.24)	71.15(3.50)	80.09(1.47)	96.80(0.94)
Lithuanian Classes	**90.26(2.78)**	88.83(2.50)	89.17(2.30)	88.10(2.20)	87.95(1.85)	77.67(3.20)	89.33(2.29)	98.35 (0.57)
Sonar	**79.72(1.86)**	74.95(2.79)	75.20(3.35)	76.51(2.06)	74.52(1.54)	74.85(1.34)	75.72(2.82)	94.46(1.63)
Ionosphere	**89.31(0.95)**	87.37(3.07)	85.71(2.12)	86.56(1.98)	86.56(1.98)	87.35(1.34)	85.71(5.52)	96.20(1.72)
Wine	**96.94(4.08)**	95.00(1.53)	95.55(2.30)	95.85(2.25)	96.16(3.02)	96.66(3.36)	95.00(4.14)	100.00(0.21)
Haberman	**76.71(3.52)**	71.23(4.16)	72.86(3.65)	70.16(3.56)	72.26(4.17)	65.01(3.20)	75.00(3.40)	97.36(3.34)

In Table 1, we compare the recognition rates obtained by the proposed meta-learning framework against dynamic selection techniques explained in this paper: Overall Local Accuracy (OLA) [6], Local Classifier Accuracy (LCA) [6], Modified Local Accuracy (MLA) [8], KNORA-Eliminate [5], K-Nearest Output Profiles (KNOP) [11] and the Multiple Classifier Behavior (MCB) [12]. We compare each pair of results using the

[2] www.prtools.org

Kruskal-Wallis non-parametric statistical test with a 95% confidence interval. The results of the proposed framework over the Pima, Liver Disorders, Blood Transfusion, Vehicle, Sonar and Ionosphere datasets are statistically superior to the result of the best DES from the literature. For the other datasets, Breast, Banana and Lithuanian, the results are statistically equivalent.

4.2 Dissimilarity Analysis

In this section, we conduct a dissimilarity analysis between distinct DES techniques. The analysis is performed based on the difference between the level of competence $\delta_{i,j}$ estimated by each DES technique for a given base classifier c_i, for each query sample $\mathbf{x}_j$ (Section 2). The goal of the dissimilarity analysis is twofold: to understand the behavior of different DES techniques (i.e., whether or not the criterion used by DES techniques present a similar behavior), and in order to see which DES criterion is closer to the behavior of the criterion used by the ideal DES scheme (Oracle) for the estimation of the competence level of a base classifier.

Given 8 dynamic selection techniques, the first step of the dissimilarity analysis is to compute the dissimilarity matrix D. This matrix D is an 8×8 symmetrical matrix, where each element $d_{A,B}$ represents the dissimilarity between two different DES techniques, A and B. Given that $\delta_{i,j}^{A}$ and $\delta_{i,j}^{B}$ are the levels of competence of c_i in relation to $\mathbf{x}_j$ for the techniques A and B, respectively, the dissimilarity $d_{A,B}$ is calculated by the difference between $\delta_{i,j}^{A}$ and $\delta_{i,j}^{B}$ (Equation 4).

$$d_{A,B} = \frac{1}{NM} \sum_{j=1}^{N} \sum_{i=1}^{M} \left(\delta_{i,j}^{A} - \delta_{i,j}^{B} \right)^{2} \tag{4}$$

where N and M are the size of the validation dataset and the pool of classifiers, respectively.

For each dataset considered in this work, a dissimilarity matrix (e.g., D_{Pima}, D_{Liver}) is computed, with the mean dissimilarity values over 20 replications. Then, the average dissimilarity matrix $\bar{D}$ is obtained by computing the mean and standard deviation of the eleven dissimilarity matrices. Table 2 shows the average dissimilarity matrix $\bar{D}$. Both the average and the standard deviation values are presented. Each line or column of the dissimilarity matrix can be seen as one axe in the $8th$ dimensional space. Each axe in this space represents the distance to a specific DES technique, for instance, the first axe represents the distance to the proposed meta-learning framework; the second represents the distance to the KNORA technique and so forth.

Classifier Projection Space. The next step is to project the dissimilarity matrix $\bar{D}$ onto the Classifier Projection Space (CPS) for a better visualization of the relationship between all techniques. The CPS is an $\mathbb{R}^{n}$ space where each technique is represented as a point and the Euclidean distance between two techniques is equal to their dissimilarities [10]. Techniques that are similar to one another appear closer in the CPS while those with a higher dissimilarity are more distant. Thus, it is possible to obtain a spatial representation of the dissimilarity between all techniques. A two-dimensional

Table 2. The average dissimilarity matrix $\bar{D}$. The values are the mean and standard deviation computed over the eleven dissimilarity matrix.

	Meta-Learning	**KNORA**	**MCB**	**LCA**	**OLA**	**MLA**	**KNOP**	**Oracle**
Meta-Learning	0	0.36(0.06)	0.46(0.15)	0.40(0.07)	0.36(0.06)	0.40(0.04)	0.53(0.08)	0.54(0.03)
KNORA	0.36(0.06)	0	0.89(0.06)	0.42(0.01)	0.44(0.01)	0.71(0.04)	0.74(0.11)	0.68(0.01)
MCB	0.46(0.15)	0.89(0.06)	0	0.58(0.01)	0.89(0.06)	1.06(0.07)	0.75(0.03)	0.72(0.08)
LCA	0.40(0.07)	0.42(0.01)	0.58(0.01)	0	0.42(0.01)	0.45(0.02)	0.31(0.04)	0.60(0.06)
OLA	0.36(0.06)	0.44(0.01)	0.89(0.06)	0.42(0.01)	0	0.71(0.04)	0.74(0.11)	0.68(0.11)
MLA	0.40(0.04)	0.71(0.04)	1.06(0.07)	0.45(0.02)	0.71(0.04)	0	0.54(0.01)	0.63(0.07)
KNOP	0.53(0.08)	0.74(0.11)	0.75(0.03)	0.31(0.04)	0.74(0.11)	0.54(0.01)	0	0.86(0.12)
Oracle	0.54(0.03)	0.68(0.01)	0.72(0.08)	0.60(0.06)	0.68(0.11)	0.63(0.07)	0.86(0.12)	0

CPS is used for better visualization. To obtain a two-dimensional CPS, a dimensionality reduction of the dissimilarity matrix $\bar{D}$ in the $\mathbb{R}^8$ to $\tilde{D}$ in the $\mathbb{R}^2$ is required. This reduction is performed using Sammon mapping [14]; that is, a non-linear Multidimensional Scaling (MDS) projection onto a lower dimensional space such that the distances are preserved [10,14].

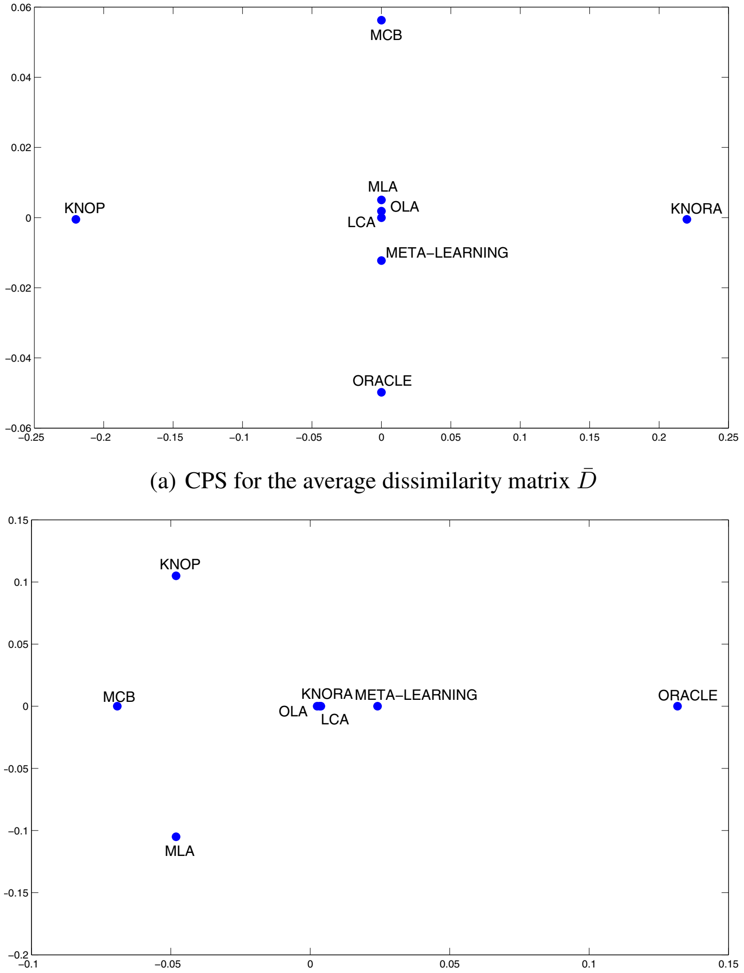
Given the dissimilarity matrix $\bar{D}$, a configuration X of m points in $\mathbb{R}^k, (k \leq m)$ is computed using a linear mapping, called classical scaling [14]. The process is performed through rotation and translation, such that the distances after dimensionality reduction are preserved. The projection X is computed as follows: first, a matrix of the inner products is obtained by the square distances $B = -\frac{1}{2}JD^2J$, where $J = I - \frac{1}{m}UU^T$, and I and U are the identity matrix and unit matrix, respectively. J is used as a normalization matrix such that the mean of the data is zero. The eigendecomposition of B is then obtained as, $B = Q\Lambda Q^T$, where Λ is a diagonal matrix containing the eigenvalues (in decreasing order) and Q is the matrix of the corresponding eigenvectors. The configuration of points in the reduced space is determined by the k largest eigenvalues. Therefore, X is uncorrelated in the $\mathbb{R}^k$, $X = Q_k\sqrt{\Lambda_k}$ space. In our case, $k = 2$.

The CPS projection is obtained by applying Sammon mapping over the matrix X. The mapping is performed by defining a function, called stress function $\mathcal{S}$ (Equation 5), which measures the difference between the original dissimilarity matrix $\bar{D}$ and the distance matrix of the projected configuration, $\tilde{D}$, where $\tilde{d}(i,j)$ is the distance between the classifiers i and j in the projection X.

$$\mathcal{S} = \frac{1}{\sum_{i=1}^{m-1}\sum_{j=i+1}^{m} d(i,j)^2} \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \left(d(i,j) - \tilde{d}(i,j)\right) \tag{5}$$

The two-dimensional CPS plot is shown in Figure 2. Figure 2(a) shows the average CPS plot obtained considering the average dissimilarity matrix $\bar{D}$, while Figure 2(b) shows an example of the CPS plot obtained for the Liver Disorders dataset D_{Liver}.

An important observation that can be drawn from Figure 2(a) is that the LCA, OLA and MLA appear close together in the dissimilarity space. Which means, that the criteria used by these three techniques to estimate the level of competence of a base classifiers present similar behaviors when averaged over several classification problems. Thus, they are very likely to achieve the same results [15]. This can be explained by



(a) CPS for the average dissimilarity matrix $\bar{D}$

(b) CPS for the dissimilarity matrix D_{Liver}, obtained for the Liver disorders dataset

Fig. 2. Two-dimensional CPS plot for the average dissimilarity matrix $\bar{D}$ and for the dissimilarity matrix obtained for the Liver disorders dataset D_{Liver}. It is important to mention that the axes of the CPS plot cannot be interpreted alone. Only the Euclidean distances between the points count.

the fact that these three techniques are based on the same information (the classification accuracy over a defined local region in the feature space), with little difference regarding the use of a posteriori information by the LCA technique or weights for the MLA technique.

The meta-learning framework appears closer to the Oracle in the two-dimensional CPS (Figures 2(a) and (b)). In addition, the meta-learning framework is also closer to the techniques from the local accuracy paradigm (LCA, OLA and MLA) than to any other DES technique, which can be explained by the fact that three out of the five meta-features comes from estimations of the local regions (f_1, f_2 and f_3).

Table 3 presents the dissimilarity measure for each DES technique in relation to the Oracle. Results show that the proposed meta-learning framework is closer to the behavior of the Oracle as it presents the lowest dissimilarity value on average, 0.54. The LCA technique comes closer, with an average dissimilarity value of 0.60. Thus, we suggest that the use of multiple criteria to estimate the level of competence of a base classifier results in a DES technique that obtains a estimation of the level of competence of a base classifier closer to that provided by an ideal DES scheme (Oracle).

Table 3. Mean and standard deviation of the dissimilarity between each DES technique from the Oracle for each classification problem. The smallest dissimilarity values are highlighted.

Database	**Meta-Learning**	**KNORA-E**	**MCB**	**LCA**	**OLA**	**MLA**	**KNOP**
Pima	**0.32(0.04)**	0.43(0.01)	0.47(0.08)	0.36(0.06)	0.43(0.01)	0.44(0.07)	0.41(0.02)
Liver Disorders	**0.50(0.04)**	0.61(0.01)	0.67(.008)	0.56(0.06)	0.61(0.01)	0.60(0.07)	0.51(0.02)
Breast Cancer	**0.59(0.35)**	1.22(0.10)	1.20(0.10)	0.69(0.01)	1.20(0.10)	0.77(0.03)	1.20(0.10)
Blood Transfusion	**0.33(0.03)**	0.40(0.01)	0.46(0.01)	0.36(.003)	0.40(0.01)	0.44(0.08)	0.4(0.01)
Banana	0.33(0.10)	0.29(0.01)	0.36(0.01)	**0.24(0.01)**	0.29(0.01)	0.36(0.01)	0.34(0.01)
Vehicle	**0.36(0.07)**	0.49(0.01)	0.48(0.02)	**0.36(0.04)**	0.49(0.01)	0.37(0.05)	0.47(0.02)
Lithuanian Classes	0.47(0.14)	0.49(0.02)	0.56(0.02)	**0.39(0.04)**	0.49(0.02)	0.54(0.01)	0.51(0.03)
Sonar	**0.58(0.10)**	0.91(0.04)	0.88(0.01)	0.70(0.01)	0.91(0.04)	0.85(0.02)	0.84(0.06)
Ionosphere	**0.62(0.22)**	0.89(0.05)	0.88(0.06)	0.70(0.07)	0.89(0.05)	0.68(0.02)	0.88(0.06)
Wine	1.03(0.20)	0.88(0.11)	0.98(0.11)	**0.73(0.02)**	0.88(0.11)	0.93(0.06)	0.82(0.14)
Haberman	**0.79(0.04)**	0.89(0.05)	1.01(0.05)	0.82(0.02)	0.89(0.05)	0.92(0.04)	0.86(0.06)
Mean	**0.54(0.05)**	0.68(0.01)	0.72(0.08)	0.60(0.06)	0.68(0.11)	0.63(0.07)	0.86(0.12)

5 Conclusion

In this paper, we conducted a study about the dissimilarity between different DES techniques. These dissimilarities are computed in order to generate a dissimilarity matrix. Through Sammon Mapping, the dissimilarity matrix is embedded in a two-dimensional space, called the Classifier Projection Space (CPS), where the Euclidean distance between two feature representations reflects their dissimilarity.

Based on the visual representation provided by the CPS, we can draw two conclusions:

- The proposed technique is closer to the Oracle in the dissimilarity space, which indicates that the use of different types of information about the behavior of base classifiers is indeed necessary in order to achieve a DES technique that is closer to the Oracle.
- Techniques that use the same kind of information to compute the level of competence of the base classifiers, such as LCA, OLA and MLA, are more likely to present the same results when their performance is averaged over several problems.

Future works in this topic include: i) The design of new sets of meta-features; ii) Carrying out a comparison of different meta-features vectors in order to achieve a set of features that can better address the behavior of the Oracle; and, iii) Increasing the number of classification problems in the analysis.

References

1. Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On combining classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence 20, 226–239 (1998)
2. Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms. Wiley-Interscience (2004)
3. Cruz, R.M.O., Cavalcanti, G.D.C., Ren, T.I.: Handwritten digit recognition using multiple feature extraction techniques and classifier ensemble. In: 17th International Conference on Systems, Signals and Image Processing, pp. 215–218 (2010)
4. de Souza Britto Jr., A., Sabourin, R., Oliveira, L.: Dynamic selection of classifiers a comprehensive review. Pattern Recognition (in press, 2014), `http://dx.doi.org/10.1016/j.patcog.2014.05.003`
5. Ko, A.H.R., Sabourin, R., Britto Jr., A.S.: From dynamic classifier selection to dynamic ensemble selection. Pattern Recognition 41, 1735–1748 (2008)
6. Woods, K., Kegelmeyer Jr., W.P., Bowyer, K.: Combination of multiple classifiers using local accuracy estimates. IEEE Transactions on Pattern Analysis Machine Intelligence 19, 405–410 (1997)
7. Cruz, R.M.O., Cavalcanti, G.D.C., Ren, T.I.: A method for dynamic ensemble selection based on a filter and an adaptive distance to improve the quality of the regions of competence. In: International Joint Conference on Neural Networks, pp. 1126–1133 (2011)
8. Smits, P.C.: Multiple classifier systems for supervised remote sensing image classification based on dynamic classifier selection. IEEE Transactions on Geoscience and Remote Sensing 40(4), 801–813 (2002)
9. Cruz, R.M.O., Sabourin, R., Cavalcanti, G.D.: On meta-learning for dynamic ensemble selection. In: International Conference on Pattern Recognition (in press, 2014)
10. Pękalska, E., Duin, R.P.W., Skurichina, M.: A discussion on the classifier projection space for classifier combining. In: Roli, F., Kittler, J. (eds.) MCS 2002. LNCS, vol. 2364, pp. 137–148. Springer, Heidelberg (2002)
11. Cavalin, P.R., Sabourin, R., Suen, C.Y.: Dynamic selection approaches for multiple classifier systems. Neural Computing and Applications 22(3-4), 673–688 (2013)
12. Giacinto, G., Roli, F.: Dynamic classifier selection based on multiple classifier behaviour. Pattern Recognition 34, 1879–1881 (2001)
13. Breiman, L.: Bagging predictors. Machine Learning 24, 123–140 (1996)
14. Cox, T.F., Cox, M.A.A.: Multidimensional Scaling, 2nd edn. Chapman and Hall (2000)
15. Cruz, R.M.O., Cavalcanti, G.D., Tsang, I.R., Sabourin, R.: Feature representation selection based on classifier projection space and oracle analysis. Expert Systems with Applications 40(9), 3813–3827 (2013)