Sesmero, M.P., Alonso-Weber, J.M., Gutiérrez, G., Ledezma, A., Sanchís, A. (2015). An ensemble approach of dual base learners for multi-class classification problems. Information Fusion, 24, pp. 122-136.

DOI: 10.1016/j.inffus.2014.09.002

An Ensemble Approach of Dual Base Learners for Multi-Class Classification Problems

M. Paz Sesmero*, Juan M. Alonso-Weber, German Gutierrez, Agapito Ledezma, Araceli Sanchis

Computer Science Department

Universidad Carlos III de Madrid

Avenida de la Universidad 30, Leganés 28911, Madrid (Spain)

msesmero@inf.uc3m.es, jmaw@ia.uc3m.es, {ggutierr, ledezma, masm}@inf.uc3m.es

* Corresponding author. Tel.: +34 91 624 9111. Fax: +34 91 624 9129. E-mail addresses: msesmero@inf.uc3m.es (M.Paz Sesmero)

Abstract

In this work, we formalise and evaluate an ensemble of classifiers that is designed for the resolution of multi-class problems. To achieve a good accuracy rate, the base learners are built with pairwise coupled binary and multi-class classifiers. Moreover, to reduce the computational cost of the ensemble and to improve its performance, these classifiers are trained using a specific attribute subset. This proposal offers the opportunity to capture the advantages provided by binary decomposition methods, by attribute partitioning methods, and by cooperative characteristics associated with a combination of redundant base learners. To analyse the quality of this architecture, its performance has been tested on different domains, and the results have been compared to other well-known classification methods. This experimental evaluation indicates that our model is, in most cases, as accurate as these methods, but it is much more efficient.

Highlights:

BCE is an ensemble focused on multi-class problems with a large number of features.

The base learners have a dual structure with a binary and a complementary classifier.

To reduce the computational cost, BCE includes a Feature Selection module.

BCE gives preference to the accuracy of the base learners over their diversity.

BCE is at least as accurate as other classification methods but is more efficient.

Keywords: Ensemble of classifiers – Multi-class Classification – Artificial Neural Networks – Feature Selection – Diversity – 5x2cv F-test.

1. Introduction

A considerable amount of research in machine learning has been devoted to developing methods that automate the classification tasks. Despite the variety and number of models that have been proposed, the construction of a perfect classifier for any given task is far from achieved [1]. An alternative to improving the accuracy of individual models has appeared during the last decades in the form of classifier ensembles, which are considered one of the most promising areas of research in supervised learning [2].

A specific kind of problem that has been devoted fewer attention concerns the application of ensembles to multi-class problems. Moreover, the lack of efficient solutions grows when the input space has a high dimensionality.

Due to most of the classification systems have been designed for resolving dichotomous problems, the approach to multi-class classification usually consists in decomposing the multiclass problem into several binary sub-problems. Nevertheless, when the learning algorithm that is implicit in the classifiers is easily adaptable to multi-class problems, the binary decomposition might not be the best approach.

In this paper, we present the Binary-Complementary Ensemble (BCE), a homogeneous ensemble of classifiers that is designed to resolve multi-class problems in which the number of

features that describe the examples is large. Given than in practical applications, training (space/time) complexity or the testing complexity can be factors as important as the accuracy; the main goal of the BCE architecture is improving the ensemble accuracy, especially in those problems with a high input dimensionality, while keeping the computational cost within reasonable bounds whenever possible.

The feasibility of the proposed ensemble has been empirically tested. This research makes a comprehensive analysis of the performance of the proposed ensemble on different domains, and the results are compared to other well-known classification methods.

This paper is organised as follows: Sections 2 and 3 provide a review of the literature on Classifier Ensembles and Feature Selection. Section 4 presents the architecture of BCE. Section 5 describes the data sets, the method and the measures used to evaluate BCE. Section 6 analyses the experimental results. Last, Section 7 presents concluding remarks and future work.

2. Ensemble of Classifiers

An ensemble of classifiers is a set of classifiers whose individual decisions are combined to obtain a system that hopefully outperforms every one of its members [2]. To achieve this goal, the members of the ensemble, known as base learners or base classifiers, must be both accurate and diverse. A classifier is accurate if its classification error is lower than that obtained when the classes are assigned in a random way. Two classifiers are diverse if they make errors on different instances [2].

An important trend for these systems is the search for diversity [3]. Some of the research has been focused on *heterogeneous classifier ensembles*, where the base learners are generated with different learning algorithms, such as artificial neural networks, decision trees, or nearest neighbour classifiers [4–6]. Another approach to achieve diversity is to *inject randomness into the learning algorithm*. For example, [7] show that training a series of Artificial Neural

Networks (ANN) on the same training set but with different initial weights can provide a set of classifiers whose behaviour can be quite different. Another method based on this approach is *Randomization* [8]. This method generates decision tress [9] in which the criterion used to expand a node is randomly selected among the 20 best candidates.

Alternatively, diversity can be achieved by using different training data sets to build individual classifiers. Such data sets can be obtained in several ways [10]:

- *Resampling the training examples*: This approach includes two of the most widely known methods to construct ensembles of classifiers: *Bagging* [11] and *Boosting* [12]. *Bagging* builds multiple versions of the training set by applying random sampling with replacement. Each new data set has the same cardinality as the original training set, but some instances are repeated while others are omitted. *Boosting* also resamples the original data set with replacement. This last system is based on a sequential training scheme in which the data set used for building each member of the ensemble depends on the performance of the previously trained classifiers. Therefore, in *Boosting*, misclassified examples are chosen more frequently than correctly predicted examples.

- *Manipulating the input features*: Another way to achieve diversity between classifiers is the quantitative or qualitative modification of the set of features that is used to describe the instances. The quantitative modifications reduce this number by searching appropriate feature subsets. This reduction can be accomplished by random selection [13] or by applying different feature selection methods, such as genetic algorithms [14,15], heuristic search techniques [16], or wrapper models [17]. The qualitative modifications involve a change in the feature space. This group includes the methods of non-linear transformations proposed in [18].

- *Manipulating the targets*: A last way for generating diverse classifiers is the manipulation of the classes or categories of the training examples. These techniques are especially useful in multi-class problems. In this situation, the principal alternative is transforming the original

problem into several binary sub-problems. These transformations can be accomplished in different ways: OAA – *One against All*– [19] (each classifier separates one class from the (*k-1*) remaining classes), OAO –*One against One*– [20] (all classes are confronted pairwise), and PAQ –*P against Q*– [21] (each classifier separates a subset *P* of classes from a subset *Q*, where *P* and *Q* are disjoint).

A representative method of the PAQ approach is ECOC –*Error Correcting Output Codes*– [22]. In ECOC, for each classifier *i* of the ensemble, the class set $C=\{c_1, c_2, \ldots c_k\}$, is randomly divided into two subsets, C_i^+ and C_i^-. Examples whose class is contained in C_i^+ are labelled as *"1"*, and examples whose class is contained in C_i^- are labelled as *"0"*. The training process delivers a set of binary classifiers that allow classifying new patterns by combining their outputs.

Another method based on PAQ decomposition is OAHO –*One Against Higher Order*– [23]. OAHO is based on a cascaded classifier architecture that ranks the *k* classes based on their number of training examples. The first classifier confronts the majority class (positive samples) against the remaining classes (negative samples). The successive classifiers repeat the same process, suppressing the previous majority class, i.e., taking the previous negative samples and confronting the next majority class against the remaining classes.

Most of the classification systems have been designed for dichotomous problems, and their extension to multi-class classification often leads to an increase in the computational cost or to a reduced system accuracy [24–26] . One way to address this difficulty is to divide the original problem into several binary sub-problems [23,27–32].

A drawback associated with some of the binary decomposition methods is that the mapping induced by the class recoding can provoke or increase the imbalance of the new classes [21]. Moreover, the dichotomous classifiers that integrate these models are trained only on partial knowledge and, in some of these architectures (OAA, OAHO), wrong decisions emitted by a

binary classifier are not rectifiable [33]. In this scenario, the system accuracy depends mainly on the accuracy of its members but not on their diversity. Therefore, for certain problems, binary decomposition might not be the best approach.

3. Feature Selection:

A drawback when dealing with real-world problems is the dimensionality of the data and the computational cost of the classification models. In these situations it can be useful to perform a Dimensionality Reduction based on Feature Selection techniques.

Feature Selection (FS) [34,35] has been applied in literature pursuing the following aims: decreasing the computational cost; increasing the data understanding and data visualization; and reducing the curse of dimensionality. However, the main purpose of Feature Selection is to increase the model accuracy, applying the idea that using as much as possible input information does not imply a better performance. Therefore, the Feature Selection is the procedure of selecting just the relevant information avoiding irrelevant and redundant information, and therefore reducing the computational complexity of the learning task.

It is worth applying FS when: input variables are irrelevant, there is no correlation to the output to be predicted (classification, clustering, or regression); and when some input variables are related to others. Besides, FS can be applied for any prediction task (classification [36], regression [37], clustering [36]), or supervised and unsupervised learning [38].

In order to carry out a Feature Selection procedure to any prediction system, a selection criterion has to be carefully chosen to fix a suitable feature subset. Hence, the criterion can be based on information acquired just from the input and targets data itself, or based on the model accuracy. Based on these criteria, the literature [34,35] establishes a taxonomy for FS methods: Filter [39], Wrapper [40] and Embedded methods [34].

Due to its computational efficiency, in this work Feature Selection is carried out applying a Filter method as *Correlation-Feature Subset Selection* [41]. This is not applied on the whole

feature set but on a selected feature subset obtained from the heuristic search known as Best First [42] ("greedy hill climbing augmented with a backtracking facility" [41]. This Feature Selection process is performed executing WEKA software [43].

4. Binary-Complementary Ensemble Architecture

As was previously mentioned, the usual alternative for solving multi-class classification problems is the decomposition of the initial problem into binary sub-problems. Nevertheless, when the classification algorithm that is implicit in the base learners is easily adaptable to multi-class problems, the binary decomposition might not be the best approach.

In [33] we started addressing the resolution of the multiclass classification problems with the proposal of a preliminary framework based on dual base learners. This system was tested on two real problems and the experimental results were rather promising. Subsequently, we realized that the diversity among the members of BCE could be improved with a modification of its architecture. So, BCE maintains a design in which the base learners are implemented with two coupled classifiers -a binary classifier (B_i) and a complementary classifier (C_i)- but now the architecture of the complementary classifier is quite different. In [33] the binary and the complementary classifiers were trained with the whole training set. In the current version of BCE, the complementary classifiers are trained only with the instances that have been labelled as negative for the corresponding binary classifier (see Fig. 1.a). This difference involves the construction of smaller, more accurate, and more diverse base classifiers, and is an important difference with the previous approximation.

It is necessary to achieve base learners whose output are a complete solution to the classification problem. Therefore, during the classification phase (Fig1.b) the answer given by B_i is included into the array given by C_i. So, the output of the *i-th* base learner is a one-dimensional-array

($\boldsymbol{Yi(x)}=\{y_1, y_2,.., y_k\}$), where component y_i comes from Bi and the other components come from C_i.

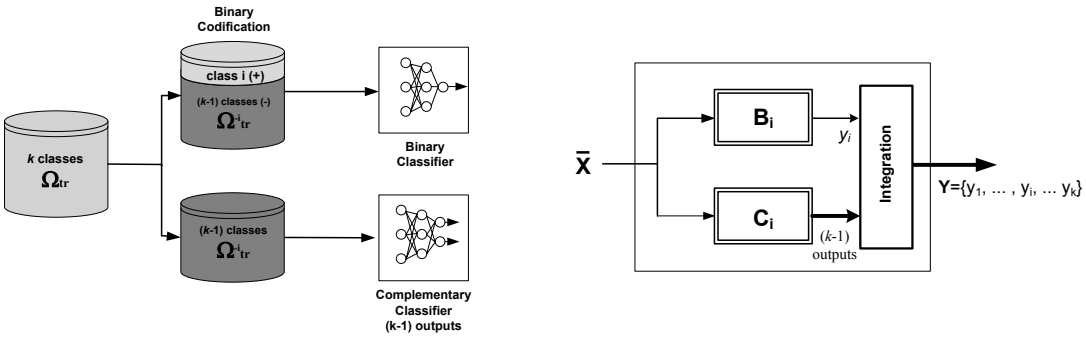

Fig. 1. a) Construction scheme of the i^{th} base learner (left). b) Performance scheme of the i^{th} base learner (right).

The Binary-Complementary decomposition attempts to ensure that the base learners are diverse and highly accurate. Nevertheless, this duality combined with the characteristics of the application domains (multi-class and a large number of attributes) has an impact on the computational cost of the BCE. To reduce this inconvenience, a module of feature selection is included in BCE.

Next sections cover the design of the individual base learners of BCE and the strategy employed for generating the output ensemble.

4.1 Design of the Binary Classifiers

The binary classifiers (Fig. 2) that compose the base learners of BCE are analogous to the classifiers used in the OAA architecture [19]. Therefore, all of them are trained on a unique data set but using different output codifications. Specifically, to build the i^{th} classifier, the original training data set Ω_{tr} is transformed into $\Omega^i_{tr}=\{\Omega^{+i}_{tr} \cup \Omega^{-i}_{tr}\}$, where Ω^{+i}_{tr} (labelled as "1") contains all of the examples of class i, and Ω^{-i}_{tr} (labelled as "0") contains the examples that belong to the remaining classes.

Once the different training sets (Ω^i_{tr}) have been built, the most relevant features are selected. This task is accomplished with the double objective of promoting diversity among the classifiers and improving the learning task, in terms of the accuracy and the computational cost. After analysing several feature selection methods [44] that are included in the Weka tool [45], Correlation-based Feature Selection [46] combined with Best First [47,48] (CFS+BF) were chosen as the search strategy.

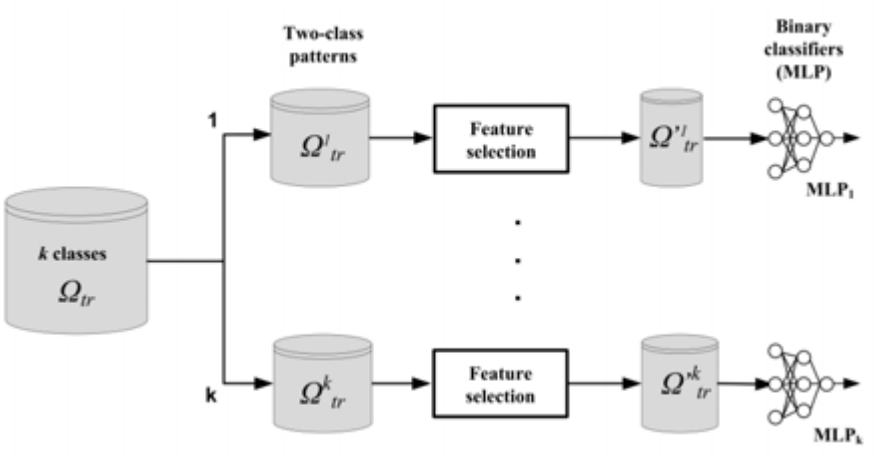

Fig. 2. Binary Classifier Design.

Therefore, each one of the binary classifiers is built using a specific output target and a specific input feature subset. In this work, these classifiers are implemented as MLPs trained with the Back-Propagation algorithm [49]. The details about their architecture, topology and parameters are described with more detail in section 5.2.1.

4.2 Design of Complementary Classifiers

The complementary classifiers are designed with the purpose of determining the class of the examples labelled as negative by the corresponding binary classifier B_i. If the binary classifier is reliable, then these examples will belong to one of the *k-1* classes that are considered to be negative during the learning phase. This step determines that, in contrast with the architecture introduced in [33] where complementary classifiers were trained with examples belonging to all

classes, in BCE the *i-th* complementary classifier is trained only with the instances that have been considered as negative by the *i-th* binary classifier, i.e., from $\{\Omega^{-i}_{tr}\}$.

Similar to in the binary classifier implementation, a process to determine the most relevant feature subset is performed before the complementary classifiers are constructed. Because $\{\Omega^{-i}_{tr}\} \neq \{\Omega^{-j}_{tr}\}$ and the used feature selection algorithm (CFS+BF) depends on the examples contained in these subsets, each complementary classifier is built on different subsets of patterns and features. Consequently, each of the complementary classifiers will learn a different hypothesis and, therefore, it is likely that their misclassified sample sets have a lower correlation.

Fig. 3 shows the construction scheme of the complementary classifiers.

Fig. 3. Construction of the complementary classifier associated with *class i.*

4.3. Base Learner Combination Method

To obtain the output of BCE, the next step is to explain the strategy employed for combining the outputs of the base learners.

MLPs provide continuous-valued outputs within the [0 - 1] range. Thus, within the *i-th* base module (M_i), the binary classifier (B_i) outputs a single real value between *0* and *1*. On the other hand, the complementary classifier (C_i) gives as output a *one-dimensional array* of (*k-1*) real values, where each component is related to one of the *k- 1* training classes. To obtain the output given by M_i, several options can be adopted [33]. The simplest approach is to include the answer of B_i into the *array* given by C_i. As a result, each base module produces an output ($\boldsymbol{Yi(x)}=\{y_1, y_2,.., y_k\}$), where component y_i comes from *Bi* and the other components come from

C_i. The combined output can be interpreted as an indicator of the degree of confidence that M_i gives to each class.

Because the base learners that integrate BCE provide a complete solution to the classification problem, the final integration scheme will follow a parallel architecture. This scenario leads to a partial redundancy among the individual decisions of the base modules, which suggests that the final decision should be made in a cooperative way.

In this type of model, the final decision can be obtained with a combination of individual classifications, either through a mathematical function (e.g., average, weighted voting) or through a *metaclassifier* [2, 4]. For both, its simplicity and its effectiveness in a large and complex data set [50] the BCE output is calculated by averaging the outputs that are associated with each class and choosing the class that attains the maximum value. Mathematically, the process is described through Eq. (1):

$$C(x) = \arg\max_{i=1}^{k} \left(\frac{\sum_{j=1}^{L} y_{ij}}{L} \right) \qquad (1)$$

where: y_{ij} is the i^{th} output of the j^{th} base module, k *i*s the number of categories, and L is the number of base modules. In BCE, k=L.

Once the structural characteristics of BCE have been presented, the next sections show the experimental analysis that is performed.

5. Experimental Setup

This section describes the data sets (Sec. 5.1) and the method and the measures (Sec. 5.2) used to evaluate BCE.

5.1. Selected Data Sets

For testing the viability of the proposed architecture we have selected 13 datasets from different sources. Table 1 compiles the main characteristics of these data sets.

Table 1. Description of the data sets used. The data sets are sorted by the number of features.

Data set	Number of Instances	Number of Features	Number of Classes	Num. Instances maj/min class	Imbalance Ratio	Source
VOWEL	990	12	11	90/90	1.00	[51,52]
SEGMENTATION	2310	18	7	330/330	1.00	[51,52]
SATIMAGE	6435	36	6	1533/626	2.45	[51,52]
TEXTURE	5500	40	11	500/500	1.00	[51,52]
SYNTHETIC	600	60	6	100/100	1.00	[51]
OPTDIGITS	5620	64	10	572/554	1.03	[51,52]
AUTOMOBILE	159	75	6	48/3	3.05	[51,52]
LIBRAS	360	90	15	24/24	1.00	[51,52]
SEMEION	1592	256	10	162/155	1.04	[51]
IMBALANCED SEMEION	1236	256	10	162/40	4.05	[51,53]
SPLICE	3190	287	3	1655/768	2.10	[51,52]
MNIST	60000	784	10	6742/5421	1.24	[54]
ASISTENTUR	1006	1024	9	478/22	21.73	[53]

5.2. Performance Evaluation

To test how well BCE works on solving classification tasks, its performance is compared to that obtained by other well-known classification systems:

1. A single one-layer MLP with k output units (k=number of classes).
2. An OAA scheme [25] modelled with k MLP.
3. *Bagging* [11] with MLP as base classifiers.
4. ECOC [22] with MLP as base classifiers.

5.2.1. Designing the Comparison

For all of the models, each base learner is a one-hidden-layer MLP trained with the *Back-Propagation* algorithm. According to Zhang [55], the importance of finding the adequate parameters for an optimal generalization capacity is more determining in the case of a simple ANN that in the case of an ANN ensemble. For each problem, the parameter search has been

performed on a single ANN based on a cross validation scheme. The selected parameter values have been also used for the ANNs in the ensembles. The *activation function* is the *logistic function,* both for the hidden and output units. The weights have been initialized with random uniform values in the interval [-1, 1]. The *learning rate*, the number of hidden nodes and the number of iterations are summarised in Table 2.

It is worth mentioning that the large number of iterations required in ASISTENTUR is a consequence of the highly imbalanced class distribution. In imbalanced data sets, the decreasing rate of the net error for the minority class is very low and, therefore, the number of iterations required by the standard *Back-Propagation* algorithm increases [19,56].

Table 2. Parameters of the evaluated models.

	Number of base classifiers			Number of Hidden units	Number of Iterations	Learning Rate
	BCE/OAA	Bagging	ECOC			
VOWEL	11	15	14	20	500	0.050
SEGMENTATION	7	15	63	10	500	0.025
SATIMAGE	6	15	31	15	600	0.050
TEXTURE	11	15	14	20	300	0.250
SYNTHETIC	6	15	31	15	300	0.025
OPTDIGITS	10	15	15	30	400	0.050
AUTOMOBILE	5	15	15	20	500	0.025
LIBRAS	15	15	15	20	300	0.250
SEMEION	10	15	15	20	300	0.025
IMBALANCED SEMEION	10	15	15	20	300	0.025
SPLICE	3	15	3	20	200	0.025
MNIST	10	15	15	100	500	0.025
ASISTENTUR	9	15	15	30	2000	0.025

For both OAA and BCE, the number of base learners is equal to the number of classes of the corresponding problem. ECOC has been constructed using the error-correcting codes proposed by [22], and therefore, the number of columns in the code determines the number of base classifiers. For determining the number of base learners of *Bagging*, we have attempted to reach

a compromise between the 50 replicas suggested by Breiman [11], the 10 suggested by Quinlan [57], and the computational cost of training an ANN. This number is quite close to the number of replicas proposed by Optiz & Maclin [58], who assert than whenever *Bagging* is implemented with Neural Networks, the largest error-reduction rate occurs when using between 10 and 15 base classifiers.

5.2.2. Ensemble Integration Method

As was mentioned before, the output of BCE is obtained by averaging the outputs associated with each class and choosing the class that attains the maximum value (Eq. 1). To avoid errors in the class assignation that are attributable to the integration method [59], the output of *Bagging*[1] is obtained as in BCE or, more precisely, using Eq. (1). For the single ANN and for the OAA approach, the predicted class corresponds to the unit that attains the highest output value. Finally, for ECOC, the class of a sample **x**, *C(**x**)*, is computed following Eq. (2):

$$C(x) = \min_i \sum_{j=1}^{L} \left| f_j(x) - w_{ij} \right| \qquad (2)$$

where: $f_j(x)$ is the output value of the *j-th* binary classifier and w_{ij} = 1/0 if class c_i belongs to one of the categories that the *j-th* classifier considers as positive/negative during the learning phase ($c_i \in C_j^+ / c_i \in C_j^-$).

5.2.3. Ensemble Performance Evaluation

To statistically evaluate the performance of BCE, we have used the combined *5x2 cv F-test* [60]. This test relies on performing five runs of a *two-fold cross-validation* [61]. In each run, the original data set is randomly partitioned into two subsets with the same cardinality, which are alternatively used as training and testing sets. If $p_i^{(j)}$ is the difference between the error rates of the two classifiers on fold *j* of run *i,* and $s_i^2 = (p_i^{(1)} - \bar{p}_i)^2 + (p_i^{(2)} - \bar{p}_i)^2$ is the estimated variance

[1] Experimentally we have noticed that simple average delivers better results than majority vote.

on run i (where $\bar{p}_i = (p_i^{(1)} + p_i^{(2)})/2$)), then the statistic given in Eq. (3) approximately follows an F distribution with 10 and 5 *degrees of freedom.*

$$F - test = \frac{\sum_{i=1}^{5}\sum_{j=1}^{2}\left(p_i^{(j)}\right)^2}{2\sum_{i=1}^{5} s_i^2} \quad (3)$$

Therefore, we reject the hypothesis that the two classifiers have the same error rate if the *F-test* is equal to or greater than the tabulated critical value for the one-tailed *F distribution* at the prespecified level of significance. At the *0.05* level, this value is equal to *4.735* [62].

Furthermore, for each model, we represent and analyse the relationship between the accuracy of the ensemble and i) the observed mean accuracy of the members in the ensemble and ii) the maximum accuracy of any base learner.

5.2.4. Diversity Evaluation

To analyse the influence of the diversity of the base classifiers on the ensemble accuracy, some well-known measures of diversity [16,63] will be computed: *fail/non-fail disagreement*, the *Q statistic*, the *correlation coefficient* and the *kappa statistic*. Table 3 shows a summary of these measures and the relationship between the obtained value and the diversity between the ensemble members (the greater/lower the value is, the more diverse the classifiers are). They all are pairwise measures because they quantify the diversity between each pair of classifiers. Therefore, the diversity of the ensemble is the averaged value over all of the pairs of classifiers in the ensemble, as given in Eq. (4):

$$M_{av} = \frac{2}{L(L-1)}\sum_{i=1}^{L-1}\sum_{j=i+1}^{L} M_{i,j} \quad (4)$$

where L is the number of base classifiers in the ensemble.

To study the relationship between the diversity and accuracy, the connection between these four measures of diversity and the gain of the ensemble is computed [64]. The gain is defined as the difference between the ensemble accuracy and the mean accuracy of its members.

In the following subsection, there is a description of the experimental results obtained with each implemented system.

Table 3. Summary of the 4 pairwise diversity measures. Monotonically increasing/decreasing measures are identified with an ascending/descending arrow.

Name	Symbol	Definition	↑/↓
fail/non-fail disagreement measure	*dis*	$\frac{N^{01}+N^{01}}{N^{11}+N^{10}+N^{01}+N^{00}}$	↑
Q statistic	Q	$\frac{N^{11}N^{00}-N^{01}N^{10}}{N^{11}N^{00}+N^{01}N^{10}}$	↓
correlation coefficient	ρ	$\frac{N^{11}N^{00}-N^{01}N^{10}}{\sqrt{\left(N^{11}+N^{10}\right)\left(N^{01}+N^{00}\right)\left(N^{11}+N^{01}\right)\left(N^{10}+N^{00}\right)}}$	↓
kappa degree-of-agreement statistic	κ	$\frac{\frac{\sum_{i=1}^{k}N_{ii}}{N}-\sum_{i=1}^{k}\left(\frac{N_{i*}}{N}\frac{N_{*i}}{N}\right)}{1-\sum_{i=1}^{k}\left(\frac{N_{i*}}{N}\frac{N_{*i}}{N}\right)}$	↓

where:

N is the cardinality of the test set.
k is the number of classes.
$\boldsymbol{N^{ab}}$ is the number of instances in the data set, classified correctly (a=1) or incorrectly (a=0) by the classifier *i*, and correctly (b=1) or incorrectly (b=0) by the classifier *j*.
$\boldsymbol{N_{ij}}$ is the number of instances in the data set, labelled as class *i* by the first classifier and as class *j* by the second classifier.

6. Experimental Results

This section contains the experimental results obtained in this work. First, in section 6.1, the accuracy and the computational cost of BCE are measured and compared with other well-established systems. Second, in section 6.2, the diversity of the base learners and its relationship

with the ensemble accuracy are analysed. Finally, the results are summarised and analysed in section 6.3.

6.1. Ensemble Performance

To evaluate the performance of the proposed system, BCE is tested against the standard classification methods specified in section 5.2. For all of the domains and all of the classification methods, 5 runs of 2-fold *stratified* cross validation (*5x2cv*) have been performed. In each run, the original data set has been randomly partitioned into two subsets, $S_i^{(1)}$ and $S_i^{(2)}$, which have the same cardinality and keep the same class distribution of the examples as in the original data set. Moreover, to control the variations that result from the inherent ANN randomness, each classification model has been trained on each data set ten times, each time with a different initial weight set. After excluding the extreme cases (minimum and maximum values), the mean error is considered to be the *actual error* of the model on each test set. Hence, the standard error E is computed following Eq. (5):

$$E = \frac{1}{5}\sum_{i=1}^{5}\frac{1}{2}\left(e_i^{(1)} + e_i^{(2)}\right) \qquad (5)$$

where $e_i^{(1)}$ is the *actual error* of the model when it is trained on $S_i^{(1)}$ and tested on $S_i^{(2)}$, and $e_i^{(2)}$ is the *actual error* of the model when it is trained on $S_i^{(2)}$ and tested on $S_i^{(1)}$.

Table 4 shows the accuracy (*1.0 - standard error*) for the evaluated classification models. Additionally, the results of the statistical comparison (considering the *F-test* and a level of significance of 0.05) between BCE and the other standard classifiers are shown.

Table 4. Accuracy values (in %) for the evaluated models. The ✓/× symbol indicates that, according to the F-test, the standard classifier is significantly better/worse than BCE. The ~ symbol indicates that the standard classifier and BCE are statistically equivalent. The best values in each domain are marked in bold.

Data Set	BCE	Standard Classifiers			
		ANN	OAA	Bagging	ECOC
VOWEL	73.93	79.13 ~	86.29 ✓	83.82 ✓	**87.05** ✓
SEGMENTATION	93.08	**93.73** ✓	92.89 ×	93.67 ✓	92.88 ~
SATIMAGE	86.14	87.16 ✓	**86.21 ~**	87.44 ✓	85.35 ×
TEXTURE	98.63	99.52 ✓	**99.49** ✓	99.65 ✓	99.56 ✓
SYNTHETIC	**97.02**	96.26 ~	94.58 ×	96.85 ~	96.61 ~
OPTDIGITS	**95.37**	92.25 ×	94.17 ×	95.36 ~	94.61 ~
AUTOMOBILE	**69.29**	64.28 ~	64.42 ~	65.13 ~	63.33 ~
LIBRAS	71.39	72.99 ~	73.15 ~	**77.08** ~	76.33 ~
SEMEION	90.12	86.10 ×	87.09 ×	**90.56** ~	88.06 ×
IMBALANCED SEMEION	**90.70**	84.71 ×	85.70 ×	89.12 ×	87.07 ×
SPLICE	94,79	93,96 ×	94,25 ~	**95,41** ~	86,41 ×
MNIST	96,91	95,26 ×	96,56 ×	96,38 ×	**96,95** ~
ASISTENTUR	**94.69**	93.27 ×	92.87 ×	94.36 ~	94.31 ~
win/tie/loss		6/4/3	7/4/2	2/7/4	4/7/2

The results in Table 4 show that BCE:

- Offers the best accuracy rate on five (SYNTHETIC, OPTDIGITS, AUTOMOBILE, IMBALANCED_SEMEION, and ASISTENTUR) of the thirteen data sets.
- Significantly outperforms OAA in 7 data sets, ANN in 6 data sets, ECOC in 4, and *Bagging* in 2.
- Is significantly better than any of the other models on the IMBALANCED SEMEION data set.
- When the number of features in the data sets is large (greater than 60), it is never significantly worse than the other classifiers. Nevertheless, when the number of features in the data sets is less than 40 the classical ensembles outperform BCE. This is the case of the VOWEL, SEGMENTATION, SATIMAGE and TEXTURE datasets.

To analyse the influence of the Feature Selection process in the results obtained for VOWEL, SEGMENTATION, SATIMAGE and TEXTURE, Table 5 shows the accuracy of BCE when it is trained and tested using the full feature space – to avoid ambiguity, this ensemble is called BCE*. Additionally, Appendix 1 gathers the results obtained with the rest of the datasets. This

study includes a) the results obtained when all the classification models (including BCE) are implemented using the whole feature set and b) the results obtained when the base learners of all classification models are implemented using the feature subsets obtained by applying BF+CFS.

Table 5. Accuracy values (in %) when all the classifiers (included BCE) are built using the full feature space. The ✓/× symbol indicates that, according to the F-test, the standard classifier is significantly better/worse than BCE*. The ~ symbol indicates that the standard classifier and BCE* are statistically equivalent. The best values in each domain are marked in bold.

Data Set	BCE*	Standard Classifiers			
		ANN	OAA	Bagging	ECOC
VOWEL	85.42	79.13 ×	86.29 ~	83.82 ~	**87.05** ✓
SEGMENTATION	**93.89**	93.73 ~	92.89 ×	93.67 ×	92.88 ×
SATIMAGE	87.33	87.16 ~	86.21 ×	**87.44 ~**	85.35 ×
TEXTURE	99.63	99.52 ~	99.49 ~	**99.65 ~**	99.56 ~
win/tie/loss		1/3/0	2/2/0	1/3/0	2/1/1

The results in Table 5 show that, in domains in which the number of attributes is relatively small (between 12 and 40), the binary-complementary decomposition is a good approach whenever the Feature Selection process is omitted. A conclusion is that Feature Selection is useless for domains with a low number of attributes.

In addition to the accuracy, another important aspect to account for while evaluating the performance of the different classifiers is the overall computation time. Given two or more algorithms that are statistically equivalent, the simpler algorithm is chosen, namely the approach with the lower running time [65].

Figure 4 shows the computation time of the implemented classification models. In this study we have excluded those data sets with a low number of attributes in which the Feature Selection makes no sense (VOWEL, SEGMENTATION, SATIMAGE and TEXTURE)

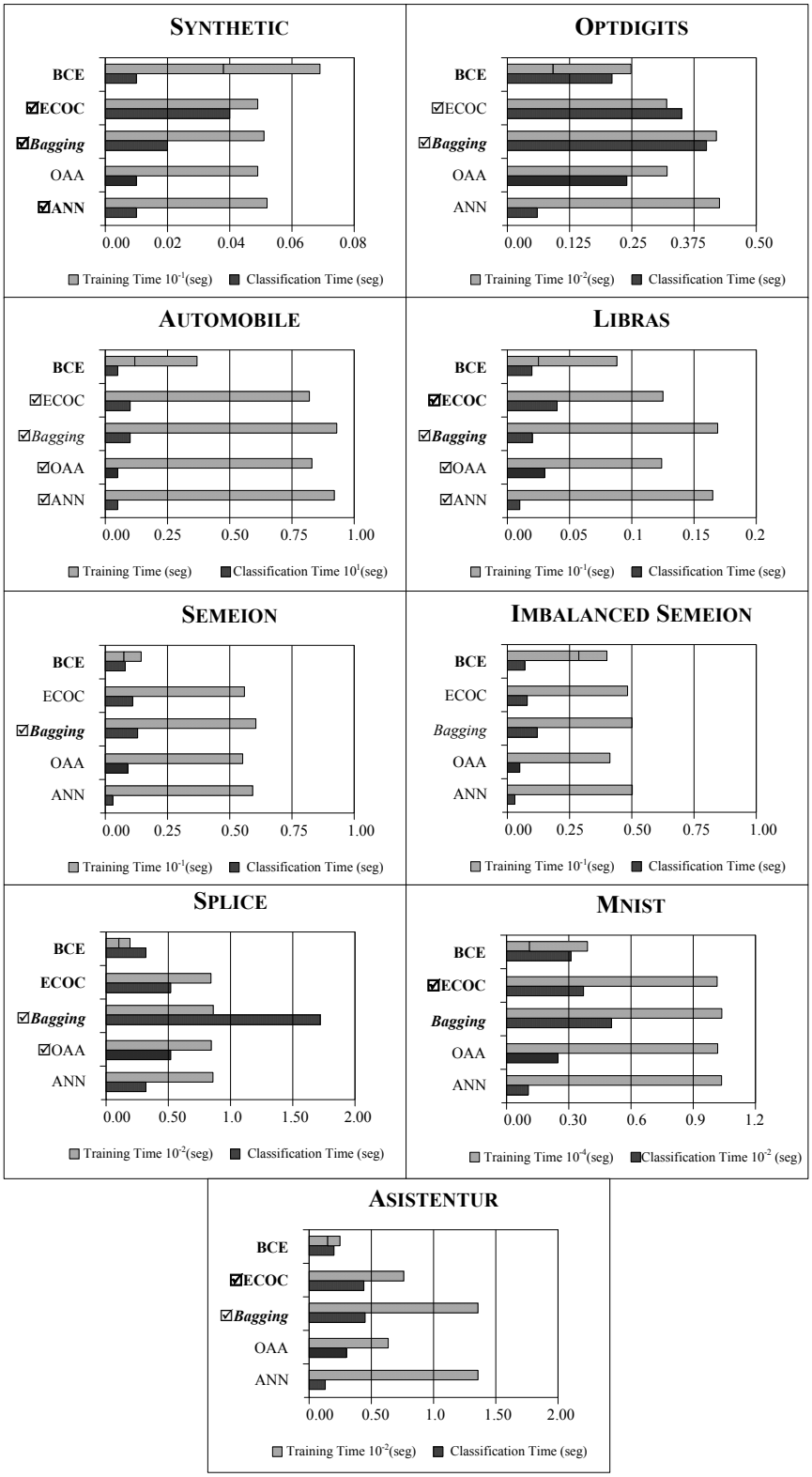

Fig.4. Ensemble classification time and training time of a base learners on an Intel Core i7-2600 CPU @ 3.40GHz. The ☑symbol indicates those systems that are statistically equivalent to BCE in terms of the accuracy. The training time for BCE is divided into two consecutive bars that correspond to the complementary and the binary classifiers that compose the base learner.

When evaluating BCE only according to its training time, it outperforms those systems that are statistically equivalent. The only exception occurs in the SYNTHETIC dataset. Here, the training time of BCE appears to be higher than the training time of both ECOC and *Bagging*. However, considering that i) the base modules of BCE are composed by two MLPs that can be trained in a parallel way, and ii) the number of base learners of BCE is lower than for *Bagging* and ECOC, it can be said that BCE outperforms both *Bagging* and ECOC. None of the ensembles appears to provide a substantial improvement over a single ANN. However, because of the parallel structure of BCE, it is possible to achieve a high computational efficiency on multiprocessor systems.

If we consider only the classification time, BCE outperforms ECOC, Bagging and the OAA architecture. Compared with the single ANN, the classification time of BCE is slightly higher for the OPTDIGITS and MNIST datasets, but in these cases the single ANN offers a higher error rate.

Considering both the accuracy and computational complexity, it can be concluded that BCE is, in general, better than the traditional ensembles (OAA, *Bagging* and ECOC) and, in the worst case, it is equally precise but much more efficient.

To estimate the quality of BCE and to check whether it outperforms every one of its members, Fig. 5 shows the relationship between the accuracy of the ensemble and i) the observed mean accuracy of the members in the ensemble and ii) the maximum accuracy of any classifier. Each graph shows the values obtained using the cross-validation method described previously; therefore, there are 100 (10x5x2) points in each plot.

This graphical representation is shown only for those models in which the base learners are redundant in the sense that each learner provides a complete answer to the classification problem, in other words, in BCE and in *Bagging*. In OAA and ECOC, this representation is not feasible.

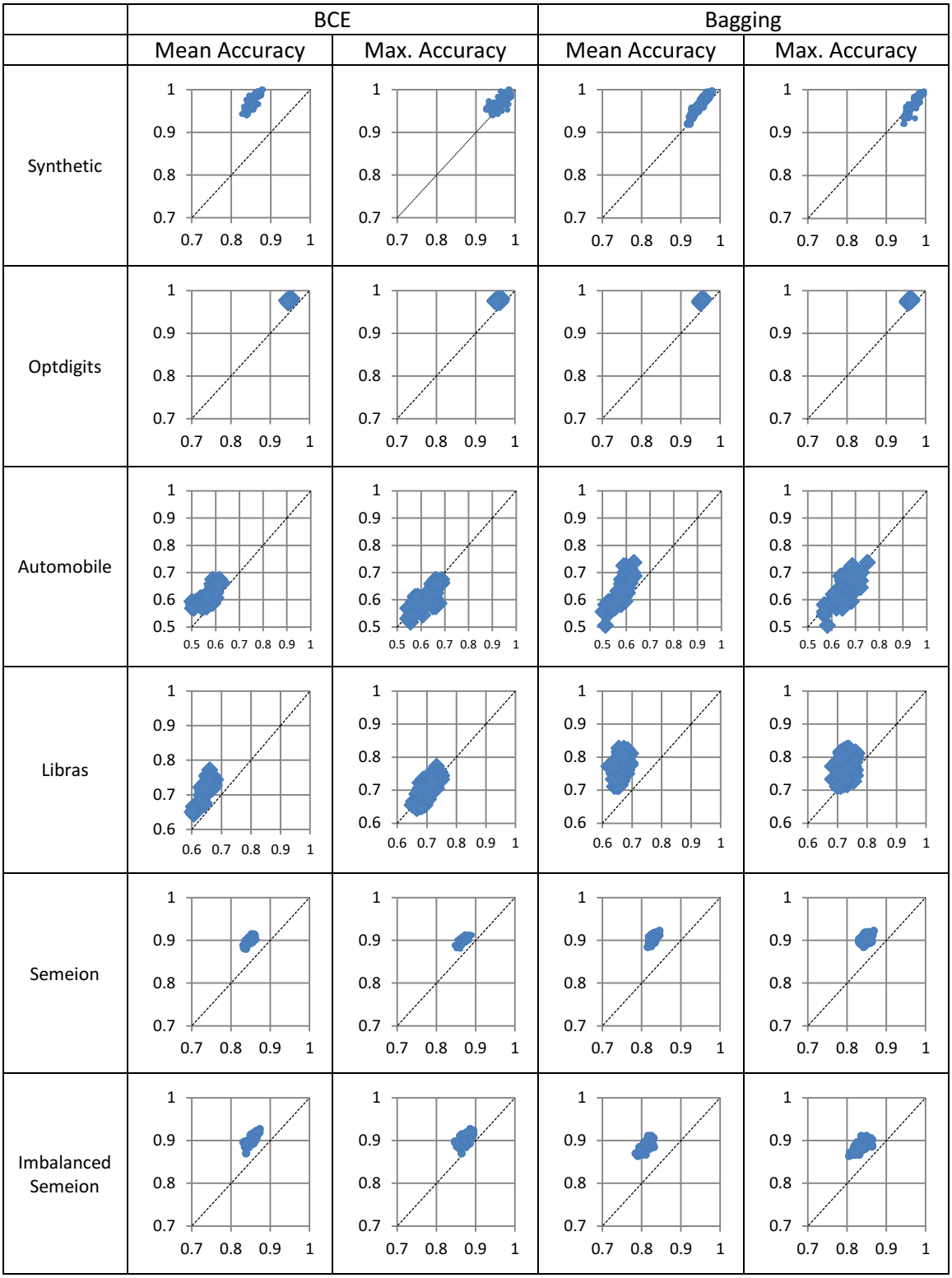

Fig. 5a.- Comparison of BCE/Bagging accuracy (y axis) and the medium/maximum accuracy of its base classifiers (x axis). Datasets: Synthetic, Optdigits, Automobile, Libras, Semeion, and Imbalanced Semeion.

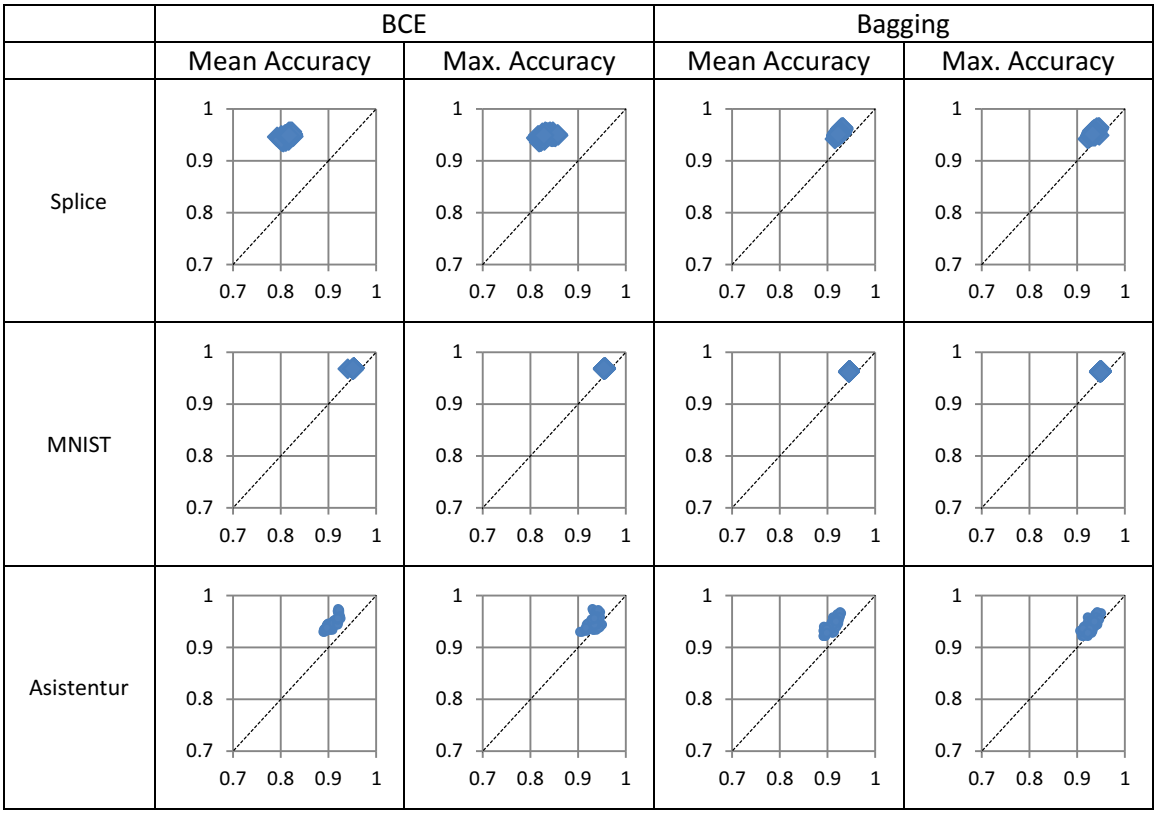

Fig.5b.- Comparison of BCE/Bagging accuracy (y axis) and the medium/maximum accuracy of its base classifiers (x axis). Dataset: Splice, MNIST, and Asistentur.

For those data sets in which BCE or *Bagging* produce ensembles that are better than every one of its base modules, all of the points on the graph in Fig. 5 lie above the dashed diagonal line. Nevertheless, the values plotted in Fig. 5 show that, sometimes, both BCE and *Bagging* are worse than the best of their members. This scenario appears in the SYNTHETIC dataset, where for BCE and *Bagging,* 14 and 61 out of 100 points are below the diagonal; in the AUTOMOBILE dataset where for BCE and *Bagging*, 61 and 70 of 100 points are below the diagonal; and in the LIBRAS dataset where for BCE 31 points are below the diagonal. In all of the other domains, both *Bagging* and BCE are more accurate than the best of their members and are significantly more accurate than the average. This fact combined with the improvement of *Bagging* and BCE over the single classifier (ANN) proves the utility of the ensembles of classifiers.

To improve the readability of the graphs in Fig. 5, Fig. 6 plots the ensemble accuracy, the interval defined by the accuracy of all of the base learners and the median of these last values.

Taking the IMBALANCED SEMEION data set as an example, the base modules of BCE are more accurate than the *Bagging* base learners. Both *Bagging* and BCE are more accurate than any of their members. BCE is significantly more accurate than *Bagging*.

Taking ASISTENTUR as a reference, the base learners of both *Bagging* and BCE have, on average, a similar accuracy; however, some of the base modules of BCE are less accurate than the *Bagging* base learners. BCE is slightly more accurate than *Bagging*.

The study of the achieved accuracy shows that, in general, BCE is similar to *Bagging*, but the reduction in the computational cost (Fig. 4) makes BCE more efficient.

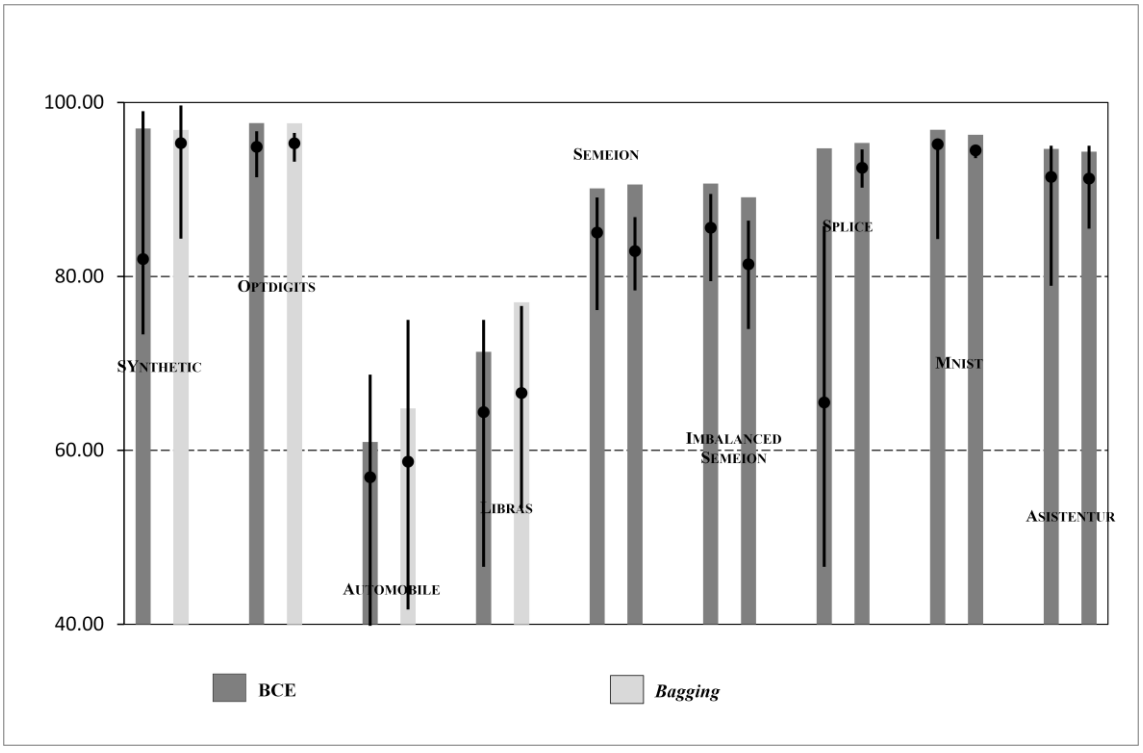

Fig. 6 Ensemble Accuracy (*bar*), interval defined by the accuracy of the base learners (*line*) and median of these last values (*point*).

6.2. Diversity Evaluation

One way to understand the behaviour of ensemble methods is to measure the correlation between the diversity and the gain of the ensemble (Ac_{Ens} - Ac_{Mean}) [16]. Because of the fact that this relationship is not linear [64], the statistical dependence of these parameters is quantified using *Spearman's rank correlation coefficient* (RCC), which is described through Eq. (6):

$$RCC = 1 - 6\sum_{i=1}^{N} \frac{\left(Rank(x_i) - Rank(y_i)\right)^2}{N\left(N^2 - 1\right)} \qquad (6)$$

where: N is the number of observations, X *and* Y are the measured variables (diversity and gain), and $Rank(x_i)$ is the position of x_i when the values of X are sorted in order of increasing magnitude.

For RCC values equal or near to 0, the relationship between the accuracy gain and the diversity of the ensembles is non-existent or low. If the variations of the diversity and the gain are strongly coupled, then the RCC will output a high absolute value. The RCC should have a negative sign for those measures that are monotonically decreasing functions (Q, ρ, and κ). Otherwise, it should have a positive sign.

Table 6 presents Spearman's rank correlation coefficients between the 4 diversity measures summarised in Table 3 and the gain of the ensemble for *Bagging* and BCE. The best results are shown in bold. It is worth mentioning that the BCE values showed for VOWEL, SEGMENTATION, SATIMAGE and TEXTURE were obtained excluding the Feature Selection Module.

Upon computing RCC, it appears to be interesting to analyse whether the obtained value is large enough to allow a conclusion as to whether the relationship between the measurement variables is statistically significant or not. To evaluate the significance of RCC [62], we assume that, under the null hypothesis, the statistic given through Eq. 7:

$$t = RCC\sqrt{\frac{N-2}{1-RCC^2}} \qquad (7)$$

is distributed following *Student's t distribution* with *N-2* degrees of freedom. Therefore, we reject the hypothesis that the relationship between the two variables is null when *t* is greater than the tabulated critical value for the two-tailed *Student's t distribution* at the prespecified level of significance. At the 0.05 level and for *N*=100 (the number of built ensembles), this value ($t_{0.05}$) is equal to 1.984.

Table 6. Spearman's rank correlation coefficient (RCC) for the total ensemble diversity and the gain of the ensemble. The × symbol indicates that, according to the T-test, the relationship between the considered diversity measure and the gain of the ensemble is not statistically significant. Bold face type indicates the best values.

	Q (↓)		ρ (↓)		κ (↓)		*des* (↑)	
	Bagging	BCE	*Bagging*	BCE	*Bagging*	BCE	*Bagging*	BCE
VOWEL*	**-0.695**	-0.601	**-0.679**	-0.523	**-0.556**	-0.468	**0.555**	0.470
SEGMENTATION*	-0.209	**-0.878**	-0.230	**-0.867**	-0.220	**-0.946**	0.239	**0.948**
SATIMAGE*	-0.537	**-0.862**	-0.554	**-0.900**	-0.463	**-0.912**	0.467	**0.917**
TEXTURE*	-0.634	**-0.784**	**-0.559**	-0.529	-0.588	**-0.971**	0.644	**0.982**
SYNTHETIC	**-0.564**	-0.403	**-0.564**	-0.506	-0.091×	**-0.182×**	0.105×	**0.195×**
OPTDIGITS	-0.449	**-0.698**	-0.079×	**-0.508**	-0.625	**-0.781**	0.663	**0.809**
AUTOMOBILE	**-0.378**	-0.31	**-0.407**	-0.308	-0.26	**-0.447**	0.258	**0.456**
LIBRAS	**-0.745**	-0.573	**-0.744**	-0.557	**-0.75**	-0.419	**0.752**	0.422
SEMEION	**-0.505**	-0.479	**-0.471**	-0.384	-0.411	**-0.473**	0.475	**0.525**
IMBALANCED SEMEION	**-0.668**	-0.536	**-0.626**	-0.514	**-0.495**	-0.433	**0.584**	0.478
SPLICE	-0.431	**-0.697**	-0.339	**-0.741**	-0.282	**-0.825**	0.302	**0.834**
MNIST	**-0.374**	-0.324	**-0.329**	-0.274	-0.304	**-0.360**	**0.522**	0.506
ASISTENTUR	-0.455	**-0.460**	**-0.288**	-0.229	-0.361	**-0.508**	0.507	**0.573**

The relationship between the diversity (Q, ρ, κ, *des*) and the *gain* of the ensemble (Ac_{Ens} - Ac_{Mean}) is illustrated in Fig. 7. Each point in the scatter plot represents an ensemble of classifiers.

Fig. 7 shows that, in most cases, the measured values of the diversity are very distant from the best possible theoretical values. They are high when they should be low (Q, ρ and κ), and they are low when they should be high.

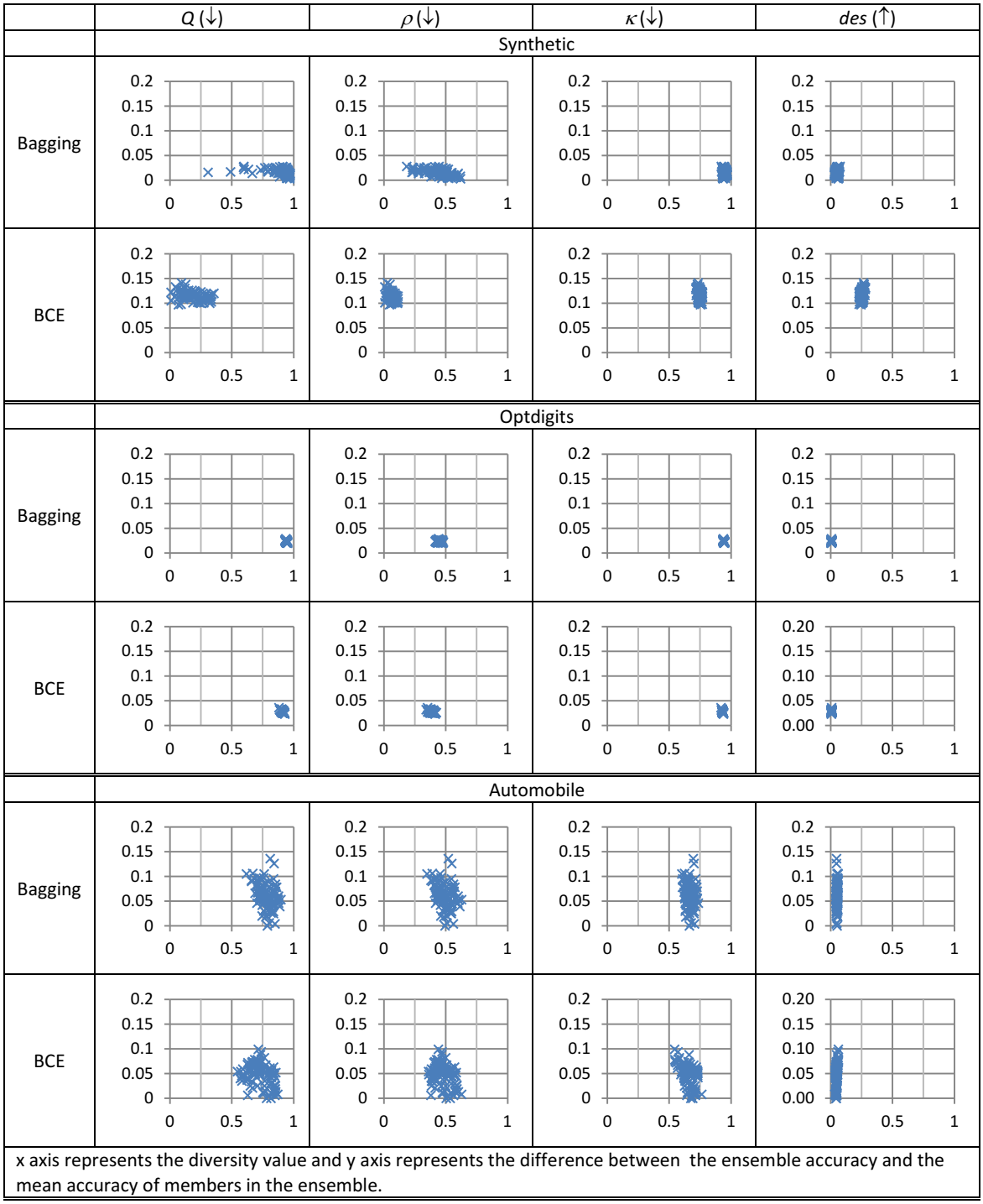

Fig. 7a Relationship between ensemble diversity (x axis) and the difference between the ensemble accuracy and the mean accuracy of the members in the ensemble (y axis). Datasets: Synthetic, Optdigits, and Automobile.

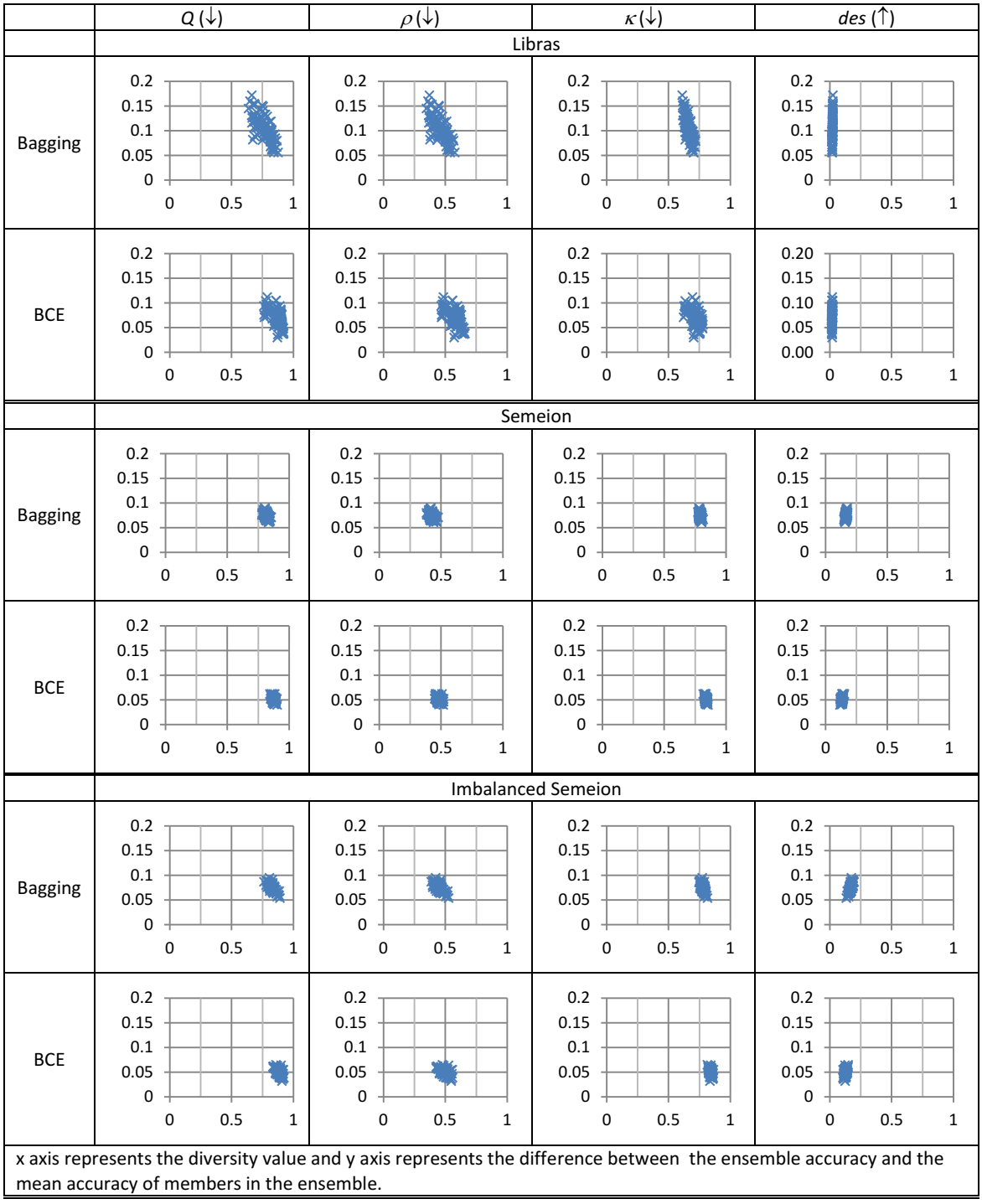

Fig. 7b Relationship between ensemble diversity (x axis) and the difference between the ensemble accuracy and the mean accuracy of the members in the ensemble (y axis). Datasets: Libras, Semeion, and Imbalanced Semeion.

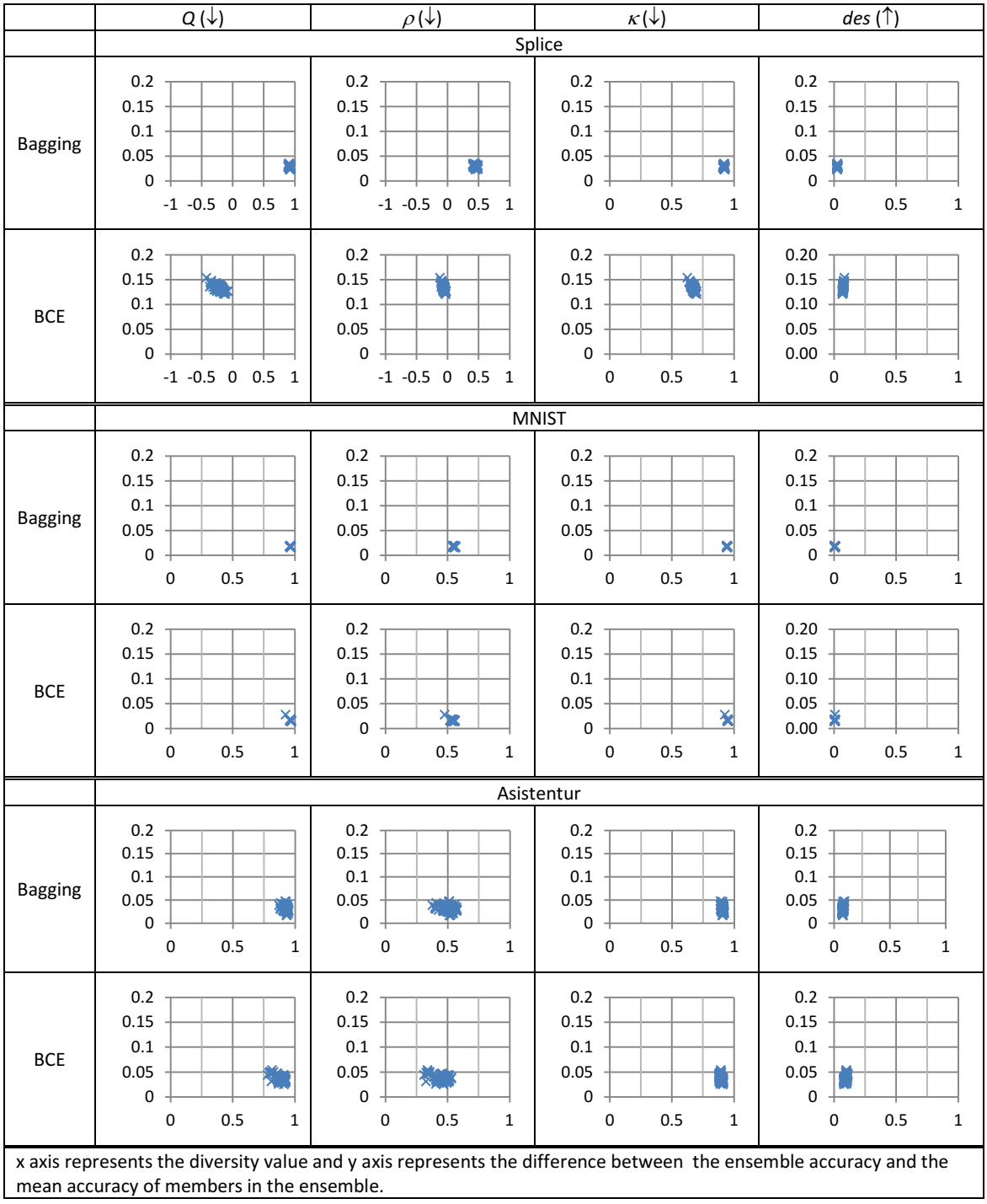

Fig. 7c.- Relationship between ensemble diversity (x axis) and the difference between the ensemble accuracy and the mean accuracy of the members in the ensemble (y axis). Dataset: Splice, MNIST, and Asistentur.

Considering the SYNTHETIC and the SPLICE datasets, BCE has a higher diversity than *Bagging* (the clouds of points for BCE are closer to the best possible theoretical values than for *Bagging*) and has a larger gain (elevated on the *y axis*). However, this improvement is not sufficient to achieve a statistically significant difference in the ensemble accuracy (see Table 4). On the other hand, for the case of LIBRAS and IMBALANCED SEMEION, *Bagging* is slightly more diverse than BCE and shows a higher gain. Nevertheless, with IMBALANCED SEMEION BCE is statistically more accurate than *Bagging,* whereas with LIBRAS both models are statistically equivalent. This finding appears to suggest that in these domains, the accuracy of the ensemble depends more on the accuracy of the base learners than on their diversity (see Fig. 6). With OPTDIGITS, AUTOMOBILE, SEMEION, MNIST and ASISTENTUR, *Bagging* and BCE show a similar diversity.

As reported in [66], these observations might be evidence that the diversity measures and the ensemble accuracy have a small relationship in real-life classification problems.

6.3. Summary of the Results

The experimental study shown in section 6.1 suggests that BCE is a feasible alternative for resolving multi-class classification problems. The results indicate (Table 4) that in five of the thirteen analysed domains, BCE offers a higher accuracy. When comparing BCE against the four other models in the domains with a high number of attributes (greater than 60), BCE is a good alternative for the classification task. On the other hand, with datasets with a relative low number of attributes, the results obtained with BCE are in some cases statistically inferior to those obtained with other classical methods. Nevertheless, when the Feature Selection Module is switched off (Table 5) the results of BCE are equivalent or even better than those obtained with the rest of the evaluated models.

When BCE is evaluated in terms of its testing time (Fig. 4), BCE is clearly more efficient than those systems that are statistically equivalent. When a comparison is performed in terms of the training time, it is observed that, in most cases BCE outperforms those ensembles that have an

equivalent error rate. However, considering the highly parallelisable nature of BCE and the number of base learners of the different ensembles (Table 2), it is possible to say than BCE outperforms these.

Considering the accuracy and the diversity of the base learners, the BCE members are usually very accurate but not very diverse (Fig. 7). Therefore, accepting a taxonomy in which the ensembles of classifiers are grouped according to the accuracy and the diversity of their base learners, BCE –as *Bagging*– is included in the category that gives preference to the accuracy over the diversity [67].

When quantifying the induced diversity of the ensembles using different measurements, the obtained values are similar for BCE and for *Bagging*. Finally, when computing the correlation between the diversity and the gain of the ensemble, it can be concluded that there is a significant relationship between the two parameters (Table 6).

7. Conclusions

In this work, we proposed BCE, an ensemble of classifiers based on ANN that efficiently resolves multi-class problems that have a feature space with a high dimensionality. In this model, each base learner is a dual arrangement that is composed of a binary and a complementary classifier. The Binary-Complementary decomposition that was initially proposed in [33], is now enhanced with a new complementary classifier architecture. This adaptation leads to the construction of more accurate and diverse base modules..

The dual character of the base modules has an impact on the computational cost of BCE. To lower this inconvenience, a feature selection process is applied before each classifier is constructed. This process depends on the instances that are contained in the training set and their class distribution. Therefore, each member of all the base learners is trained using a

specific data set and a specific feature subset. Consequently, it is likely that each of them learns a different hypothesis and, therefore, a degree of diversity can be achieved. Additional diversity is achieved by implementing the classifier as an ANN with randomly generated initial weight sets.

We have tested the performance of BCE in thirteen different domains, and to evaluate their performance we have compared BCE against a single ANN, OAA, ECOC, and *Bagging*. The experimental results indicate that BCE significantly outperforms OAA in 7 data sets, ANN in 6 data sets, ECOC in 4, and *Bagging* in 2. On the other hand, BCE is statistically worse than OAA and ECOC in 2 data sets, ANN in 3 and Bagging in 4. It is worth mentioning that these unfavourable results appears only in domains in which the number of features is relatively small. The accuracy values obtained when BCE is trained and tested using the full feature space (BCE*) show that these bad results are a consequence of the feature selection process, rather than due to some problem in the binary-complementary decomposition.

In respect to the computational cost, in domains with a large number of features, BCE is equal or more accurate than other traditional classifiers but clearly much more efficient. This question is more irrelevant when the number of features is low.

For the domains with a large number of features (>60), we have analyzed the relationship between the accuracy of BCE and i) the observed mean accuracy of its base learners and ii) the maximum accuracy of its base learners. Moreover, to analyze the influence of the diversity on the accuracy, the correlation between the former and the gain of BCE has been computed. This analysis shows that the base learners of BCE are accurate and diverse and that the relationship between diversity and the gain of the ensemble, while not strong, is statistically significant.

In the future, we intend to analyse the dependence between BCE and the construction methodology of the base learners. This work should include other learning algorithms combined with the construction of heterogeneous ensembles. Furthermore, achieving higher diversity rates is also an important aim that should entail an increased accuracy of the ensemble of classifiers.

Acknowledgments

This research was supported by the Spanish MICINN under projects TRA2010-20225-C03-01, TRA 2011-29454-C03-02, and TRA 2011-29454-C03-03.

Appendix 1

This appendix shows the accuracy of the evaluated models when all the base learners are implemented a) using the whole feature set and b) using the feature subsets obtained by applying BF+CFS

Table A. Accuracy values (in %) when all the classifiers (including BCE) are built using the full feature space. The ✓/× symbol indicates that, according to the F-test, the standard classifier is significantly better/worse than BCE*. The ~ symbol indicates that the standard classifier and BCE* are statistically equivalent. The best values in each domain are marked in bold.

Data Set	BCE*	Standard Classifiers			
		ANN	OAA	Bagging	ECOC
VOWEL	85.42	79.13 ×	86.29 ~	83.82 ~	**87.05** ✓
SEGMENTATION	**93.89**	93.73 ~	92.89 ×	93.67 ×	92.88 ×
SATIMAGE	87.33	87.16 ~	86.21 ×	**87.44 ~**	85.35 ×
TEXTURE	99.63	99.52 ~	99.49 ~	**99.65 ~**	99.56 ~
SYNTHETIC	96.48	96.26 ~	94.85 ~	**96.85** ~	96.61 ~
OPTDIGITS	**95.80**	92.25 ×	94.17 ×	95.36 ×	94.61 ×
AUTOMOBILE	**66.33**	64.28 ~	64.42 ×	65.13 ~	63.33 ~
LIBRAS	**77.62**	72.99 ×	73.15 ~	77.08 ~	76.33 ~
SEMEION	**91.11**	86.10 ~	87.09 ~	90.56 ~	88.06 ~
IMBALANCED SEMEION	**90.16**	84.71 ~	85.70 ~	89.12 ~	87.07 ~
SPLICE	94.96	93.96 ×	94.25 ×	**95.41** ✓	86.41 ×
MNIST	96.85	95.26 ~	96.56 ~	96.38 ~	**96.95** ~
ASISTENTUR	**95.03**	93.72 ~	92.87 ~	94.36 ~	94.31 ~
win/tie/loss		4/9/0	5/8/0	2/10/1	4/8/1

Table B. Accuracy values (in %) when all the classifiers are built using the feature subsets obtained by applying BF+CFS. The ✓/× symbol indicates that, according to the F-test, the standard classifier is significantly better/worse than BCE*. The ~ symbol indicates that the standard classifier and BCE* are statistically equivalent. The best values in each domain are marked in bold.

Data Set	BCE	Standard Classifiers			
		ANN*	OAA*	Bagging*	ECOC*
VOWEL	73.93	68.44 ×	61.74 ×	**78.03** ✓	57.14 ×
SEGMENTATION	93.08	**93.56** ~	88.35 ×	93.46 ~	87.70 ×
SATIMAGE	86.14	86.31 ~	81.95 ×	**87.91** ~	81.97 ×
TEXTURE	**98.63**	97.44 ×	92.78 ×	98.43 ~	95.33 ~
SYNTHETIC	97.01	96.01 ~	94.77 ×	96.85 ~	**97.04** ~
OPTDIGITS	**95.37**	91.58 ×	89.70 ×	94.87 ×	88.39 ×
AUTOMOBILE	**69.29**	67.55 ~	69.44 ~	69.00 ~	65.64 ~
LIBRAS	71.39	63.06 ×	56.23 ×	**75.66** ~	45.41 ×
SEMEION	**90.12**	85.96 ×	85.96 ×	86.68 ~	89.99 ×
IMBALANCED SEMEION	**90.70**	85.88 ×	88.32 ×	89.36 ×	85.20 ×
SPLICE	94.79	93.47 ~	94.13 ~	**94.86** ×	86.71 ×
MNIST	**96.91**	95.32 ×	93.82 ×	96.39 ×	94.70 ×
ASISTENTUR	**94.69**	93.34 ×	91.72 ×	94.15 ~	93.73 ~
win/tie/loss		8/5/0	11/2/0	3/9/1	10/3/0

References

[1] R. Ranawana, Multi-Classifier Systems: Review and a roadmap for developers, Int. J. Hybrid Intell. Syst. 3 (2006) 35–61.

[2] T.G. Dietterich, Ensemble Methods in Machine Learning, in: Mult. Classif. Syst. Lect. Notes Comput. Sci., Springer Berlin Heidelberg, 2000: pp. 1–15.

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

[4] D. Wolpert, Stacked Generalization*, Neural Networks. 5 (1992) 241–259.

[5] D. Bahler, L. Navarro, Methods for Combining Heterogeneous Sets of Classifiers, in: Proc. the17th Natl. Conf. Artif. Intell. Work. New Res. Probl. Mach. Learn., 2000.

[6] A. Ledezma, R. Aler, A. Sanchis, D. Borrajo, GA-stacking: Evolutionary stacked generalization, Intell. Data Anal. 14 (2010) 89–119.

[7] J.F. Kolen, J.B. Pollack, Backpropagation is Sensitive to Initial Conditions, Complex Syst. 4 (1990) 269–280.

[8] T.G. Dietterich, An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization, Mach. Learn. 40 (2000) 139–157.

[9] J. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[10] T.G. Dietterich, Machine-Learning Research: Four Current Directions, AI Mag. 18 (1997) 97–137.

[11] L. Breiman, Bagging Predictors, Mach. Learn. 24 (1996) 123–140.

[12] R.E. Schapire, The Strength of Weak Learnability, Mach. Learn. 5 (1990) 197–227.

[13] T.K. Ho, The Random Subspace Method for Constructing Decision Forests, IEEE Trans. Pattern Anal. Mach. Intell. 20 (1998) 832–844.

[14] D. Optiz, Feature Selection for Ensembles, in: Proc. 16th Int. Conf. Artif. Intell., 1999: pp. 379–384.

[15] L.S. Oliveira, R. Sabourin, F. Bortolozzi, C.Y. Suen, A Methodology for Feature Selection Using Multi-Objective Genetic Algorithms for Handwritten Digit String Recognition, Int. J. Pattern Recognit. Artif. Intell. 17 (2003) 903–929.

[16] A. Tsymbal, M. Pechenizkiy, P. Cunningham, Diversity in Search Strategies for Ensemble Feature Selection, Inf. Fusion. 6 (2005) 83–98.

[17] R. Bryll, R. Gutierrez-Osuna, F. Quek, Attribute Bagging: Improving Accuracy of Classifier Ensembles by Using Random Feature Subsets, Pattern Recognit. 36 (2003) 1291–1302.

[18] A.J.C. Sharkey, N.E. Sharkey, Combining diverse neural nets, Knowl. Eng. Rev. 12 (1997) 231–247.

[19] R. Anand, K.G. Mehrotra, C.K. Mohan, S. Ranka, An Improved Algorithm for Neural Network Classification of Imbalanced Training Sets, IEEE Trans. Neural Networks. 4 (1993) 962–969.

[20] T. Hastie, R. Tibshirani, Classification by Pairwise Coupling, Ann. Stat. 6 (1998) 451–471.

[21] G. Ou, Y.L. Murphey, Multi-class pattern classification using neural networks, Pattern Recognit. 40 (2007) 4–18.

[22] T.G. Dietterich, G. Bakiri, Solving Multiclass Learning Problems via Error-Correcting Output Codes, J. Artif. Intell. Res. 2 (1995) 263–286.

[23] Y.L. Murphey, H. Wang, G. Ou, OAHO: An Effective Algorithm for Multi-Class Learning from Imbalanced Data, in: Proc. Int. Jt. Conf. Neural Networks, 2007: pp. 406–411.

[24] R. Anand, K. Mehrotra, C.K. Mohan, S. Ranka, Efficient Classification for Multiclass Problems Using Modular Neural Networks, IEEE Trans. Neural Networks. 6 (1995) 117–124.

[25] E.L. Allwein, R.E. Schapire, Y. Singer, Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers, J. Mach. Learn. Res. 1 (2000) 113–141.

[26] E. Gelenbe, K.F. Hussain, Learning in the Multiple Class Random Neural Network, IEEE Trans. Neural Netw. 13 (2002) 1257–67.

[27] D.M.J. Tax, R.P.W. Duin, Using two-class classifiers for multiclass classification, in: Proc. the16th Int. Conf. Pattern Recognition., 2002: pp. 124–127.

[28] P. Kraipeerapun, C. Fung, K. Wong, Multiclass Classification using Neural Networks and Interval Neutrosophic Sets, in: Proc. 5th WSEAS Int. Conf. Comput. Intell. Man-Machine Sist. Cybern., Venice, 2006: pp. 123–128.

[29] O. Lézoray, H. Cardot, Comparing Combination Rules of Pairwise Neural Networks Classifiers, Neural Process. Lett. 27 (2008) 43–56.

[30] N. García-Pedrajas, D. Ortiz-Boyer, An empirical study of binary classifier fusion methods for multiclass classification, Inf. Fusion. 12 (2011) 111–130.

[31] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes, Pattern Recognit. 44 (2011) 1761–1776.

[32] T.H. Oong, N.A. Mat Isa, One-against-all ensemble for multiclass pattern classification, Appl. Soft Comput. 12 (2012) 1303–1308.

[33] M.P. Sesmero, J.M. Alonso-Weber, G. Gutiérrez, A. Ledezma, A. Sanchis, A new artificial neural network ensemble based on feature selection and class recoding, Neural Comput. Appl. 21 (2012) 771–783.

[34] I. Iguyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (2003) 1157–1182.

[35] G. Chandrashekar, F. Sahin, A survey on feature selection methods, Comput. Electr. Eng. 40 (2014) 16–28.

[36] H. Liu, L. Yu, Toward Integrating Feature Selection Algorithms for Classification and Clustering, IEEE Trans. Knowl. Data Eng. 17 (2005) 491–502.

[37] B. Frénay, G. Doquire, M. Verleysen, Is mutual information adequate for feature selection in regression?, Neural Networks. 48 (2013) 1–7.

[38] J. Tang, H. Liu, Unsupervised Feature Selection for Linked Social Media Data, in: Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., ACM, New York, NY, USA, 2012: pp. 904–912.

[39] R. Battiti, Using mutual information for selecting features in supervised neural net learning, Neural Networks, IEEE Trans. 5 (1994) 537–550.

[40] G.H. John, R. Kohavi, K. Pfleger, Irrelevant Features and the Subset Selection Problem, in: Mach. Learn. Proc. Elev. Int. Conf., Morgan Kaufmann, 1994: pp. 121–129.

[41] M.A. Hall, Correlation-based Feature Subset Selection for Machine Learning, University of Waikato, 1998.

[42] E. Rich, K. Knight, Artificial Intelligence, McGraw Hill, 1991.

[43] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The WEKA Data Mining Software: An Update, SIGKDD Explor. Newsl. 11 (2009) 10–18.

[44] M.P. Sesmero, J.M. Alonso-Weber, G. Gutierrez, A. Ledezma, A. Sanchis, Testing Feature Selection in Traffic Signs, in: Proc. 11th Int. Conf. Comput. Aided Syst. Theory - EUROCAST 2007, 2007: pp. 396–398.

[45] I.H. Witten, E. Frank, M.A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, Second Edi, Morgan Kaufmann, 2005.

[46] M.A. Hall, Correlation-based Feature Selection for Machine Learning, 1999.

[47] L. Xu, P. Yan, T. Chang, Best First Strategy for Feature Selection, in: 9th Int. Conf. Pattern Recognit., 1988: pp. 706–708.

[48] S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, 2003.

[49] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning Internal Representations by Error Propagation, in: D.E.R. James L. McClelland (Ed.), Parallel Distrib. Process. Explor. Microstruct. Cogn., 1985: pp. 318–362.

[50] N.C. Oza, K. Tumer, Classifier ensembles: Select real-world applications, Inf. Fusion. 9 (2008) 4–20.

[51] A. Frank, A. Asuncion, UCI Machine Learning Repository, Univ. California, Irvine, Sch. Inf. Comput. Sci. (2010).

[52] J. Alcalá-Fdez, L. Sánchez, S. García, S. Ventura, J.M. Garrell, J. Otero, et al., KEEL: a software tool to assess evolutionary algorithms for data mining problems, Soft Comput. 13 (2009) 307–318.

[53] M.P. Sesmero, Diseño. Análisis y Evaluación de Conjuntos de Clasificadores basados en Redes de Neuronas, Carlos III de Madrid, 2012.

[54] Y. LeCun, THE MNIST DATABASE of handwritten digits, (n.d.).

[55] G. Zhang, Neural networks for classification: a survey, IEEE Trans. Syst. Man, Cybern. Appl. Rev. 30 (2000) 451–462.

[56] L. Bruzzone, S.B. Serpico, Classification of imbalanced remote-sensing data by neural networks, Pattern Recognit. Lett. 18 (1997) 1323–1328.

[57] J. Quinlan, Induction of decision trees, Mach. Learn. 1 (1986) 81–106.

[58] D. Optiz, R. Maclin, Popular Ensemble Methods: An Empirical Study, J. Artif. Intell. Res. 11 (1999) 169–198.

[59] R.P.W. Duin, D.M.J. Tax, Experiments with Classifier Combining Rules, Mult. Classif. Syst. Lect. Notes Comput. Sci. 1857 (2000) 16–29.

[60] E. Alpaydin, Combined 5 × 2 cv F Test for Comparing Supervised Classification Learning Algorithms, Neural Comput. 11 (1999) 1885–1892.

[61] T.G. Dietterich, Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms., Neural Comput. 10 (1998) 1895–1923.

[62] D.J. Sheskin, Handbook of Parametric and Nonparametric Statistical Procedures, Second Edi, Chapman & Hall/CRC, 2000.

[63] L.I. Kuncheva, C.J. Whitaker, Ten Measures of Diversity in Classifier Ensembles: Limits for Two Classifiers, in: Proc. IEEE Work. Intell. Sens., 2001: pp. 10/1–10/10.

[64] L.I. Kuncheva, C.J. Whitaker, Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy, Mach. Learn. 51 (2003) 181–207.

[65] E. Alpaydin, Introduction to Machine Learning, Second Edi, MIT Press, 2010.

[66] C.A. Shipp, L.I. Kuncheva, Relationships between combination methods and measures of diversity in combining classifiers, Inf. Fusion. 3 (2002) 135–148.

[67] J.J. Rodríguez, L.I. Kuncheva, C.J. Alonso, Rotation forest: A new classifier ensemble method., IEEE Trans. Pattern Anal. Mach. Intell. 28 (2006) 1619–30.