Classifier Subset Selection to Construct Multi-Classifiers by means of Estimation of Distribution Algorithms

Andoni Arruti[a], Iñigo Mendialdua[b], Ekaitz Jauregi[b], Elena Lazkano[b], Basilio Sierra[b]

[a]*Department of Computer Architecture and Technology*
University of the Basque Country (UPV/EHU), andoni.arruti@ehu.es
[b]*Department of Computer Science and Artificial Intelligence*
University of the Basque Country, http://www.sc.ehu.es/ccwrobot

Abstract

This paper proposes a novel approach to select the individual classifiers to take part in a Multiple-Classifier System. Individual classifier selection is a key step in the development of multi-classifiers. Several works have shown the benefits of fusing complementary classifiers. Nevertheless, the selection of the base classifiers to be used is still an open question, and different approaches have been proposed in the literature. This work is based on the selection of the appropriate single classifiers by means of an evolutionary algorithm. Different base classifiers, which have been chosen from different classifier families, are used as candidates in order to obtain variability in the classifications given. Experimental results carried out with 20 databases from the UCI Repository show how adequate the proposed approach is; Stacked Generalization multi-classifier has been selected to perform the experimental comparisons.

Keywords: Machine Learning, Multiple-Classifier Systems, Evolutionary Computation, Classifier Subset Selection

1. Introduction

The Machine Learning (ML) research area, and more specifically Supervised Classification, addresses the problem of building, from correctly classified datasets, models able to deal with new unclassified cases and assign them a predicted class; it is worth mentioning that good classification accuracy is expected from the classifier.

It has been experimentally observed that the construction of a perfect classifier, using a single paradigm, is often impossible. Therefore, designers have tried to combine several classifiers, and this idea has led to the development of Multiple-Classifier Systems (MCS), which attempt to combine the advantages of different individual classifiers, sometimes built using different training paradigms, to obtain better results. Classifier combination is a viable alternative to using a single classifier, and has become an established research area, thriving mostly on heuristic solutions. Some theoretical results are also available but only for special cases, usually assuming independent classifier outputs [1, 2, 3, 4].

Intuitively, it makes sense that a combination of classifiers provides better results than a single decision maker. However, this depends on how independent and diverse the individual classifiers are [5], and thus the diversity among the selected classifiers is one of the key design features within a successful multi-classifier.

There are different MCS strategies. Some approaches organize the different classifiers in a tree [6], other approaches create various classifiers by using different subsets of features to train them [7], and certain approaches divide the multi-class classification problems in several two-class sub-problems, in the so called Class-Binarization strategies [8] [9]. The most common strategies found in the literature are Boosting [10], Bagging [11] and Stacked Generalization (SG) [12].

In Bagging and Boosting, diversity is achieved by manipulating the training examples in order to generate multiple hypotheses. The base classifier is trained several times, each time with a different subset of the training examples, thus creating different classifiers. Finally there should be a method to combine the outputs of this set of classifiers.

Stacked Generalization [13] basically follows a layered architecture. At the level-0, classifiers are trained using the original dataset and each classifier outputs a prediction for each token. Successive layers receive as inputs the predictions of the previous layer, and at the level-1, a single classifier, also called meta-classifier, outputs the final

prediction. The overall performance not only depends on the individual classifiers used at the level-0, but also on the correct selection of classifiers at other levels. One problem of Stacked Generalization is how to obtain the right combination of level-0 classifiers and the meta-classifier, especially in relation to each specific dataset.

Two difficulties arise when a MCS has to be developed: the selection of base classifiers, and how to combine their individual decisions.

By combining the outputs of a team of classifiers, we aim at a more accurate decision than that of the best member of the team. The assumption is that developers should introduce diversity in the ensemble, and therefore enhance the performance. There are different ways to tackle output combination; a weighted majority vote is the standard combination method for ensembles but there are other combination methods which could also be successful. Many combination methods and algorithms have been developed, including the use of a meta-classifier, which is the option selected to construct a Stacked Generalization model.

The difficulty in choosing a suitable combination method for the problem at hand has been recognized and highlighted numerous times in the literature. Some theoretical works rely on simplifications and assumptions, and consider mostly special cases [14, 15, 16]. However, even a discipline as mature as pattern recognition does not offer strict guidelines about how to approach a data set and which classifier to select for it; many experimental studies have been published in the search for such guidelines [17, 18]. This study also belongs to this experimental group.

This paper presents a methodology to incorporate into a Multiple-Classifier System a mechanism which attempts to adapt the structure of the MCS to a given classification problem. This approach has the following properties:

1. To be a global optimisation technique which has been shown to be successful in complex domains, such as the space of the possible configurations for a Multiple-Classifier System given a pool of individual classifiers.
2. To provide an easy way to express the problem of optimisation of the multiple-classifier structure.
3. To combine the individual classifier opinions in the form established by the MCS itself (voting, meta-classifier,...); the searching process itself selects the appropriate base classifiers taking into account the mode in which the final decision is taken.

We present a Multiple-Classifier System which incorporates an automatic self-configuration scheme based on Estimation of Distribution Algorithms [19] (an evolutionary computation approach). Our main interest is focused on the constituent classifiers of the resulting multi-classifier.

To show the behaviour of the proposed method, 20 datasets have been selected from the UCI repository [20] and 10 standard classifiers are used by the new Classifier Subset Selection (CSS) proposed method to construct a Stacked Generalization MCS. Experiments are carried out comparing the results of the 10 single classifiers, state-of-the-art multi-classifiers (Boosting, Bagging and StackingC) and the Stacked Generalization classifiers constructed with the whole set of 10 base classifiers in the level-0. Results obtained show the goodness of the approach, as the new paradigm statistically outperforms the remaining classifiers used.

The rest of the paper is organized as follows: In Section 2 related work in the area of multi-classifiers is presented. Section 3 describes the proposed approach and section 4 the experimental setup. Finally, Section 6 depicts some conclusions and future work lines.

2. Related work

Several papers can be found in the literature about construction and use of Multiple-Classifier Systems. In this section we reflect three main aspects: different MCS which have been used in different tasks; in the next subsection the revision is focused on the Stacked Generalization paradigm, as it will be that used in the experimental phase, and in the third subsection we revise the use of evolutionary algorithms in order to obtain better multi-classifiers.

2.1. Multiple-Classifier Systems

Combination of classifiers has been widely used as a useful approach in several Machine Learning tasks [21].

In the field of people detection, several authors have used multi-classifier approaches: [22] use Histograms of Oriented Gradients (HOGs) and Local Receptive Fields (LRFs), which are provided by a convolutional neural network, and are classified by Multi-Layer Perceptrons (MLPs) and Support Vector Machines (SVMs) combining classifiers by majority vote and fuzzy integral; [23, 24, 25] present a MCS to manage image based classification problems; Batista

et al. [26], take advantage of unigrams, bigrams and trigrams to design a Multiple-Classifier System for Sentiment Analysis and Opinion Mining.

In [27], to improve the performance of classification, three classifiers that have the best results among all applied methods are combined; on the other hand, [28] present a Multiple-Classifier System based on color and texture information for face image segmentation; Haibo et al. [29], present a hybrid MSC to improve the precision of remote sensing image classification. Taking the characteristic of abstract level and measurement level into consideration, the optimal sub-classifier, Bagging algorithm and the largest confidence algorithm are combined.

In [30], the problem of Multiple-Classifier System design is discussed and the reader is provided with a critical survey of the state-of-the-art. The main conclusion in this chapter is that optimal design is still an open problem. More information about Multiple-Classifier Systems can be found in [31].

2.2. Stacked Generalization

Ting and Witten [32] propose to extend Stacked Generalization using class probability distributions of the original classifiers. Moreover, they propose to use the Multi-Response Linear Regression (MRLR) as meta-classifier. Seewald [33] discovered that this new version worked correctly for two-class problems while it performed worse for multi-class problems. In order to solve this problem, he proposes a new method called StackingC where, for each class separately, a meta training set is created with the class probabilities associated with the class. In this case, he also use MRLR as meta-classifier.

As can be seen in the literature the Stacked Generalization multi-classifier has been applied in different types of problems: For example Ekbal et al. [34] use a Stacked Generalization multi-classifier for the extraction of biomedical entities in the forms of genes and gene product mentions in text and Ibarguren et al. [35] use a Stacked Generalization to real time recognition of the Fingerspelling Alphabet used by the deaf people.

2.3. Base classifier configuration by means of Evolutive Computation

Quian et al. investigate MSC as a multi-goal problem from a theoretical point of view [36]. Whereas Rahman et al. [37] propose a novel cluster oriented ensemble classifier generation method and a Genetic Algorithm based approach to optimize the parameters.

Impedovo et. al. [38] propose a new method for handwritten digit recognition where, for each individual classifier, a feature selection is applied. They consider the problem of feature selection as an optimization problem so they use the genetic algorithms in order to find the best performance of the combined classifier. On the other hand, in order to predict subcellular localization of apoptosis proteins, Ding and Zhang [39] propose a new method where each individual classifier is trained with different dimensions of protein sequences and they use the genetic algorithms to find the optimal weight factors.

Zhou et al. [40] has proved that ensembling some of the available classifiers may be better than ensembling all of them. Viewing that, Kim and Kang [41] propose a new method where they use the genetic algorithms for classifier selection in ensembles. In this case they concentrate on the selection process of an ensemble containing diverse classifiers.

In the literature we can also find some works that try to improve the performance of Stacked Generalization by selecting the best classifiers using different evolutionary computation strategies. Chen [42], [43], proposes a new ensemble construction method which applies Ant Colony Optimization (ACO) in the Stacked Generalization ensemble construction process to generate domain-specific configurations. Shunmugapriya and Kanmani [44], use Artificial Bee Colony(ABC) Algorithm as a meta-heuristic search algorithm to obtain a suitable Stacked Generalization model. To this end, two versions of the ABC algorithm are used. Ledezma et al. [45] use the genetic algorithms and show that selecting the right classifiers, their parameters and the meta-classifier is a critical issue.

3. Proposed approach

As explained in the introduction, the main goal of this work is optimal selection of base classifiers to construct multi-classifiers. In this section, the elements and processing sequence that constitute the proposed approach are explained in detail.

3.1. Combination of classifiers

To combine the results of the base classifiers, we use Stacked Generalization (SG) as Multiple-Classifier System. Stacked Generalization is a well known ensemble approach and is also called Stacking [12, 46]. While ensemble strategies such as Bagging or Boosting obtain the final decision after a vote among the predictions of the individual classifiers, SG applies another individual classifier to the predictions in order to detect patterns and improve performance of the vote.

As can be seen in Figure 1, SG is divided into two levels: in the level-0 each individual classifier makes a prediction independently, and in the level-1 these predictions are treated as the input values of another classifier, known as meta-classifier, which returns the final decision.

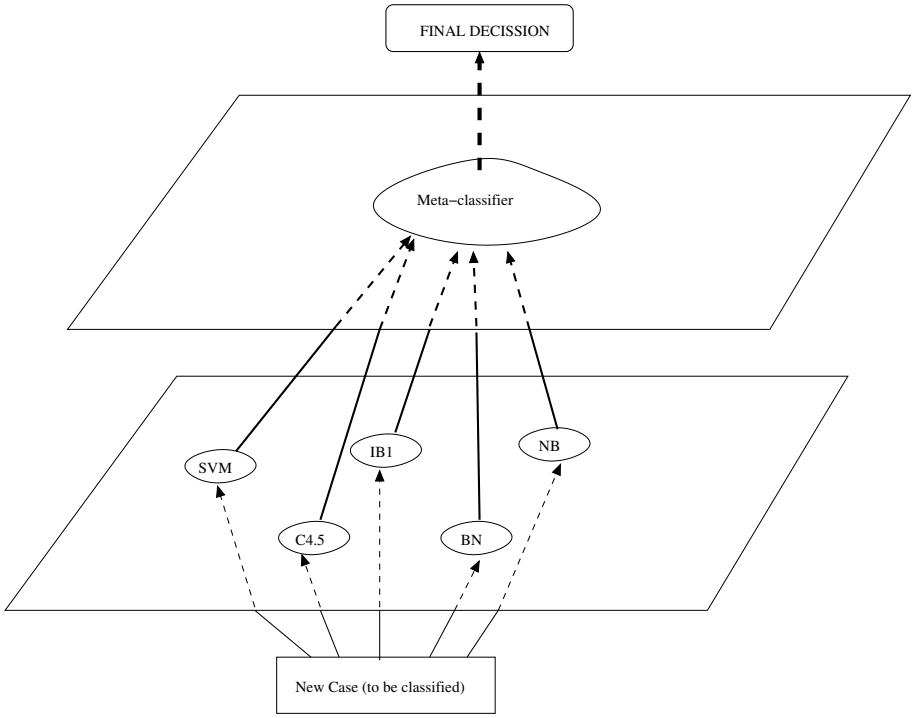

Figure 1: Stacked Generalization schemata

The data for training the meta-classifier is obtained after a validation process, where the outputs of the level-0 classifiers are taken as attributes and the class is the real class of the example.

3.2. Classifier Subset Selection (CSS) as a search problem

Although using many classifiers may seem more effective, our believe is that selecting a subset of them can reduce the computational cost and improve the accuracy, assuming that the selected classifiers are diverse and independent between them.

In this paper we propose a new multi-classifier paradigm, which extends the Staking Generalization approach, reducing the number of classifiers to be used in the final model. We call this new approach Classifier Subset Selection (CSS) and a graphical example is illustrated in Figure 2. As can be seen, we added to the multi-classifier an intermediate phase in which a subset of the level-0 classifiers is selected. The criterion to make the selection depends on the goal of the classification task, and in this case, we have decided to use classification accuracy. As can be seen in Figure 2, discarded classifiers -those with an X- are not used in the multi-classifier.

CSS can be contemplated in a similar way as Feature Subset Selection (FSS) in some ML problems. As reported by Aha and Bankert [47], the objective of FSS in Machine Learning is to reduce the number of features used to characterize a dataset so as to improve the performance of a learning algorithm on a given task. Our objective will

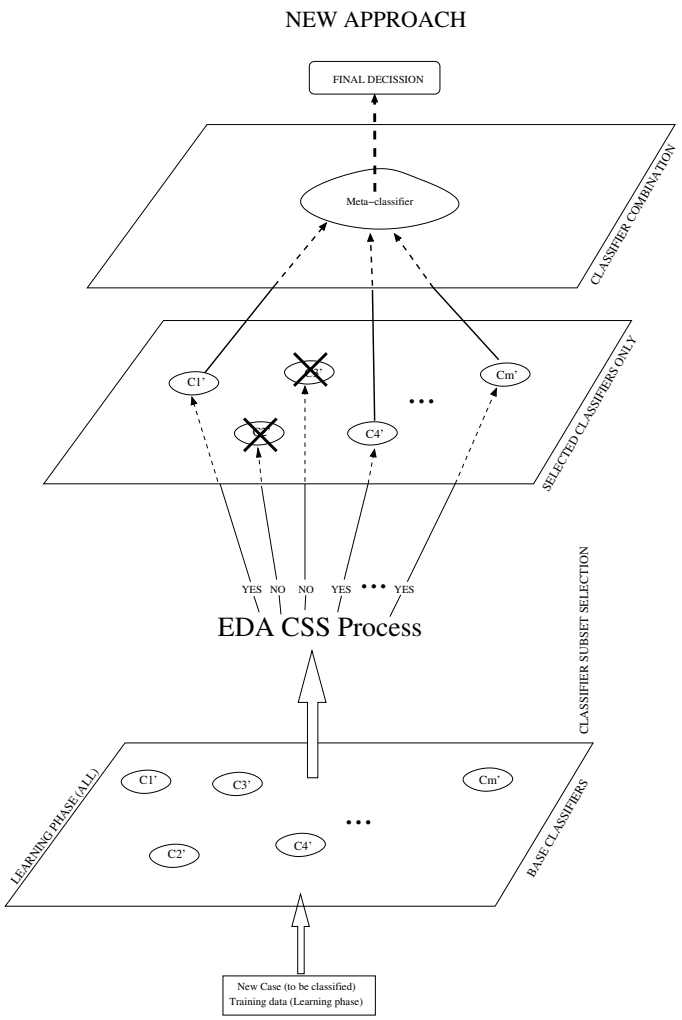

Figure 2: Classifier Subset Selection Stacked Generalization.

be the maximization of the classification accuracy in a multi-classifier; CSS task can be thus exposed as a search problem, each state in the search space identifying a subset of possible base classifiers selected.

The method used to select the classifiers could be any one, but in this type of scenarios evolutionary approaches are often introduced with promising results. Today, some of the best known evolutionary algorithms for FSS, are based on Estimation of Distribution Algorithms (EDAs) [19]. EDA combines statistical learning with population-based search in order to automatically identify and exploit certain structural properties of optimization problems. Inza et al. [48] proposed an approach that used an EDA called Estimation of Bayesian Network Algorithm (EBNA) [49] for a FSS problem. Viewing that in [50] EBNA shows better behaviour than genetic and sequential search algorithms for FSS problems (and hence for CSS in this approach), we decide to use EBNA. Moreover EBNA has been selected as the model in the recent work that analyses the behaviour of the EDAs [51].

EBNA is an EDA that learns a Bayesian Network and it follows the typical EDA structure. It starts with a population of candidate solutions to the problem, starting with a population generated with uniform distribution over all admissible solutions. The population is then scored using a fitness function. This fitness function gives a numerical ranking for each string, with the higher the number the better the string. From this ranked population, a subset of the most promising solutions are selected by the selection operator. The characteristic of EBNA is that it uses a Bayesian Network to deal with the probability distribution of the selected solutions. Once the model is constructed, new solutions are generated by sampling the distribution encoded by this model. These new solutions are then incorporated

back into the old population. The process is repeated until some termination criteria is met (usually when a solution of sufficient quality is reached or when the number of iterations reaches some threshold). Figure 3 shows the pseudocode of the algorithm.

EBNA
$D_{t=0} \leftarrow$ Generate N individuals (the initial population) randomly.
Repeat for $l = 1, 2, \ldots$ until a stop criterion is met.
$V_i \leftarrow$ Evaluate the Value V_i for each of the N individuals.
$D^s_{t-1} \leftarrow$ Select $S \leq N$ individuals from D_{t-1} according to a selection method.
$p_l(\mathbf{x}) = p(\mathbf{x}|D^s_{t-1}) \leftarrow$ Estimate the joint probability distribution of an individual being among the selected individuals with the trained Bayesian Network.
$D_l \leftarrow$ Sample N individuals (the new population) from $p_l(\mathbf{x})$.

Figure 3: Main scheme of the EDA approach.

In our approach, an individual in the EDA algorithm will be defined as a n-tuple of binary 0, 1 values –the so called Binary Encoding–, and each position in the tuple refers to a concrete base classifier, and the value indicates whether this classifier is used (1 value) or not (0 value). An example with 10 classifiers (the value used in this paper) can be seen in Figure 4. In this example, Cl1, Cl4 and Cl7 are the selected classifiers, and the remaining seven are not used.

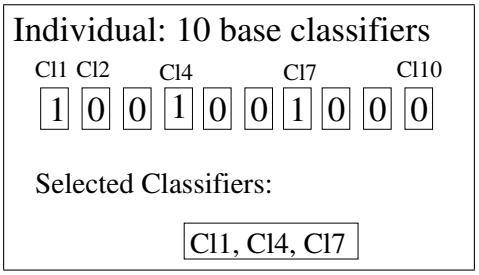

Figure 4: The combinations of base classifiers as EDA individuals

Once an individual has been sampled, it has to be evaluated. The aim is to consider the predictive power of each subset of base classifiers. For this, a multi-classifier is built for each individual using the corresponding subset of classifiers and the obtained validated accuracy is used as fitness function. Thus, when looking for the individual that maximizes the fitness function, the EDA algorithm is also searching the optimal subset of base classifiers.

For individual selection we decide to use the range-based selection, selecting the best S individuals from the N individuals of the population. Although S can be any value, we decide to set it to N/2 since is the most common value in EDA literature.

4. Experimental results

In this section we show the set-up of the experimental framework and the obtained results.

4.1. Data Sets

In order to evaluate the performance of the proposed approach, 20 databases have been selected from the UCI repository [20]. In Table 1 the characteristics of these databases are shown. The number of cases ranges from 101 to 1,728, the number of attributes from 4 to 35 and the number of classes from 2 to 19, so a wide variety of problems is represented.

4.2. Base Classifiers

To carry out the experiments, we have used 10 well-known ML supervised classification algorithms from a software package for Machine Learning called WEKA [52].

Among the classifiers that WEKA offers, we have selected the following:

Table 1: Characteristics of the databases

Database	#Cases	#Classes	#Atributes
Balance-scale	625	3	4
Breast-cancer	286	2	9
Car	1728	4	6
Cmc	1473	3	9
Colic	368	2	22
Diabetes	768	2	8
Ecoli	336	8	7
Glass	214	7	9
Hepatitis	155	2	19
Iris	150	3	4
Limph	148	4	18
Liver-disorders	345	2	6
Solar-flare-1	323	6	12
Solar-flare-2	1066	6	12
Soybean	683	19	35
Vehicle	846	4	18
Vote	435	2	16
Vowel	990	11	13
Wine	178	3	13
Zoo	101	7	17

1R:. is a one level decision tree which tests just one attribute [53]. The chosen attribute is that which produces minimum error.

KNN:. is a case-based, Nearest-Neighbor classifier [54]. To classify a new test sample, a simple distance measure is used to find the training instance closest to the given test instance, and then it predicts the same class as this nearest training instance.

RIPPER:. (Repeated Incremental Pruning to Produce Error Reduction) [55] is a rule-base learner, an optimized version of IREP, that forms rules through a process of repeated growing (to fit training data) and pruning (to avoid overfitting). RIPPER handles multiple classes by ordering them from least to most prevalent, and then treating each in order as a distinct two-class problem.

Naive Bayes (NB):. The Naive-Bayes rule [56] uses the Bayes theorem to predict the class for each case, assuming that the predictive genes are independent given the category. To classify a new sample characterized by d genes $\mathbf{X} = (X_1, X_2, ..., X_d)$, the NB classifier applies the following rule:

$$c_{NB} = \arg\max_{c_j \in C} p(c_j) \prod_{i=1}^{d} p(x_i|c_j)$$

where c_{NB} denotes the class label predicted by the Naive-Bayes classifier and the possible classes of the problem are grouped in $C = \{c_1, \ldots, c_l\}$.

C4.5:. The C4.5 [57] represents a classification model by a decision tree. The tree is constructed in a top-down way, dividing the training set and beginning with the selection of the best variable in the root of the tree.

K:*. is an instance-based algorithm that uses an entropy-based distance function [58].

Bayesian Networks (BN):. A Bayesian network [59], belief network or directed acyclic graphical model is a probabilistic graphical model that represents a set of random variables and their conditional independencies via a directed acyclic graph (DAG).

Naive Bayes Tree (NBT):. uses a decision tree with naive Bayes classifiers at the leaves [60].

Random Forest (RF):. constructs a combination of many unpruned decision trees [61]. The output class is the mode of the classes output by individual trees.

Support Vector Machines (SVM):. are a set of related supervised learning methods used for classification and regression [62]. Viewing input data as two sets of vectors in an n-dimensional space, an SVM will construct a separating hyperplane in that space, one which maximizes the margin between the two data sets.

As can be seen, we have selected classifiers with different approaches to learning, and widely used in different classification tasks. The goal is to combine them in a multi-classifier to maximize the benefits of each modality by intelligently fusing their information, and by overcoming the limitations of each modality alone. As we treat the classifiers as black boxes, we have used the default parameters of the classifiers.

4.3. Method

The experimental phase is divided in four steps that are applied to each of the 20 databases:

1. Single classifiers: build 10 classifiers, applying the 10 base Machine Learning algorithms to the training data set, and get validated classification accuracies.
2. Standard multi-classifiers: build 5 classifiers, applying Bagging (REPTree and C4.5), Boosting (DecitionStump and C4.5) and StackingC Machine Learning algorithms to the training data set, and get validated classification accuracies.
3. Stacked Generalization: applying classic Stacked Generalization algorithm (with the ten base classifiers at level-0), build 10 classifiers, one for each base classifier at level-1, and get validated classification accuracies.
4. Classifier Subset Selection for Stacked Generalization: using our new approach, build 10 classifiers, one for each base classifier at level-1, and each of them with only a subset of classifiers, selected by EDA algorithm.

4.4. Experimental setup

In all the experiments 10-fold cross-validation [63] has been used to get a validated classification accuracy (well classified rate), and this accuracy has been the criterion to define the fitness of an individual, inside the evolutionary algorithm.

For Classifier Subset Selection, the EDA algorithm used has been EBNA (Estimation of Bayesian Network Algorithm) [49], with Algorithm B [64] for structural learning of the Bayesian Network. Population size N has been set to 50 individuals, representing 50 combination of classifiers, the number S of selected individuals at each generation is 20 (40% of the population size), and the number of generations of new individuals was set to 4.

4.5. Obtained Results

4.5.1. Single classifiers results

All base classifiers are evaluated independently for each dataset; it is worth mentioning that the obtained results are supposed to be outperformed by multi-classifiers.

Table 2 shows the results obtained by each single classifier on each dataset. As can be seen, the type of classifier that obtains the best accuracy for each classification problem varies considerably. As a matter of fact, all of the base classifiers, except 1R, obtain the best validated result for at least one dataset.

4.5.2. Standard multi-classifiers results

In order to obtain a more honest comparison with the proposed approach, some state-of-the-art multi-classifier results are also shown; we have used Bagging, Boosting and StackingC

Table 3 shows the result obtained by those standard multi-classifiers. The best results are obtained by different approaches for different datasets as well.

4.5.3. Stacked Generalization results

Obtained results for Stacked Generalization classifier ensembles are shown in Table 4, using the 10 base paradigms as meta-classifier (level-1) in the SG structure. All paradigms used as meta-classifier, except C4.5, obtain the best result for at least one of the classification problems. It should be noticed that the meta-classifier with the best result differs in general from the best single classifier obtained for the same dataset; only in five of the 20 datasets (diabetes, lymph,solar-flare-2, soybean and zoo) is there a coincidence.

A graphical image of the results obtained by the SG paradigm is shown in Figure 5; it can be seen that, for each dataset, the result varies significantly for each meta-classifier. This fact could indicate that the selection of an appropriate meta-classifier is very important in the SG classifier design.

Table 2: Single classifiers' results

Dataset	1R	KNN	RIPPER	NB	C4.5	K*	BN	NBT	RF	SVM
Balance-scale	56.32	86.56	80.80	**90.40**	76.64	88.48	72.32	76.64	80.48	87.68
Breast-cancer	65.73	72.38	70.98	71.68	**75.53**	73.43	72.03	70.98	69.23	69.58
Car	70.02	93.52	86.46	85.53	92.36	87.56	85.71	**94.21**	92.65	93.75
Cmc	48.13	44.33	**52.41**	50.78	52.14	50.24	51.05	51.73	50.85	48.20
Colic	81.52	81.25	84.24	77.99	85.33	76.63	81.25	82.06	**86.14**	82.61
Diabetes	72.79	70.18	76.04	76.30	73.83	69.14	74.35	74.35	73.83	**77.34**
Ecoli	64.88	80.36	81.25	**85.42**	84.23	80.95	81.25	82.14	83.63	84.23
Glass	58.41	70.56	68.69	48.60	66.82	**75.23**	70.56	70.56	72.90	56.08
Hepatitis	83.23	80.64	78.06	84.52	83.87	81.94	83.23	80.00	82.58	**85.16**
Iris	93.33	95.33	94.00	**96.00**	**96.00**	94.67	92.67	94.00	95.33	**96.00**
Liver-disorders	59.42	62.90	64.64	55.36	68.70	66.96	56.23	66.09	**68.99**	58.26
Lymph	75.00	82.43	77.70	83.11	77.03	85.14	85.81	80.41	81.08	**86.49**
Solar-flare-1	54.80	68.11	**72.45**	65.64	72.14	69.35	66.87	70.59	68.42	70.28
Solar-flare-2	61.45	72.89	70.45	74.39	74.48	74.67	74.48	74.58	72.98	**75.23**
Soybean	39.97	91.22	91.95	92.97	91.51	87.99	93.27	91.51	91.66	**93.85**
Vehicle	51.42	69.86	68.56	44.80	72.46	71.39	60.05	72.93	**77.07**	74.35
Vote	95.63	92.41	95.40	90.11	**96.32**	93.33	90.11	95.63	95.86	96.09
Vowel	31.82	**99.29**	69.70	63.74	81.52	98.99	60.81	93.53	96.06	71.41
Wine	77.53	94.94	91.57	96.63	93.82	**98.88**	**98.88**	96.63	97.19	98.31
Zoo	42.57	**96.04**	86.14	95.05	92.08	**96.04**	94.06	94.06	89.11	**96.04**

Table 3: Bagging, Boosting and StackingC results

Dataset	AdaBoost M1 (DS)	Bagging (REPTree)	AdaBoost M1 (C4.5)	Bagging (C4.5)	StackingC
Balance-scale	72.32	82.72	78.88	82.24	**88.96**
Breast-cancer	70.28	68.88	69.58	**73.43**	73.08
Car	70.02	91.67	**96.12**	93.52	95.54
Cmc	42.70	**54.24**	50.78	54.11	53.77
Colic	81.25	84.78	83.42	**85.60**	84.51
Diabetes	74.35	74.48	72.40	74.09	**76.17**
Ecoli	64.58	83.33	81.25	84.82	**86.01**
Glass	44.86	71.03	74.30	71.03	**74.77**
Hepatitis	82.58	83.23	**85.81**	83.23	83.87
Iris	**95.33**	94.67	93.33	**95.33**	93.33
Liver-disorders	66.09	71.30	71.59	**72.75**	70.72
Lymph	74.32	79.05	81.08	79.05	**85.14**
Solar-flare-1	46.75	**73.38**	71.21	72.14	72.75
Solar-flare-2	53.47	74.86	73.17	73.83	**75.42**
Soybean	27.97	87.12	92.83	93.27	**93.70**
Vehicle	39.95	72.34	76.24	76.60	**76.83**
Vote	95.40	95.86	95.86	**96.32**	96.09
Vowel	17.37	86.87	93.33	90.40	**99.09**
Wine	91.57	94.94	96.63	94.94	**97.75**
Zoo	60.40	42.57	**95.05**	93.07	**95.05**

4.5.4. Classifier Subset Selection for Stacked Generalization

Finally CSS has been applied to each of the datasets and meta-classifiers; obtained results indicate the appropriateness of the proposed approach.

It should be noticed that, although the learning time is time consuming (until the EDA search converges), the classification time is very short, and it can be done in real time. In this way, and for any dataset used, once the classifier is constructed it can be used at a very high frequency, as it is composed of fast classifiers.

The results obtained by the proposed approach are shown in Table 5; in this case, 1R is the only meta-classifier for which a best result is not obtained in any dataset. Best SG meta-classifier is also different in 10 of the 20 classification problems, which indicates different structures between the standard SG and the CSS approaches. Regarding the best results obtained, RF appears as meta-classifier in 10 datasets, while KNN, NB and SVM obtain as meta the best results in 5 classification problems.

A graphical image of the results obtained by the CSS paradigm is shown in Figure 6. Comparing them with the results of standard SG (Figure 5), it can be seen that, together with the improvement of accuracy, the variance of results for each database is reduced. In Table 6 means and standard deviations of accuracy values are listed for different types of classifiers and for each database. The best means and lowest standard deviation have been marked. As it can be seen. the proposed approach obtains the best mean in all the databases and the lowest standard deviation in most of

Table 4: Stacked Generalization results

Dataset	1R	KNN	RIPPER	NB	C4.5	K*	BN	NBT	RF	SVM
Balance-scale	93.12	90.56	92.96	92.48	93.28	89.92	91.68	93.92	**94.72**	92.80
Breast-cancer	69.23	65.03	69.23	69.23	71.68	65.03	69.93	72.03	70.28	**72.73**
Car	93.92	96.70	99.07	95.72	98.90	97.22	96.70	98.21	**99.31**	98.03
Cmc	45.21	45.48	50.85	53.90	47.86	47.73	53.56	52.75	52.48	**54.11**
Colic	**85.60**	78.26	84.51	85.05	82.34	79.89	**85.60**	82.88	84.51	84.24
Diabetes	71.48	70.18	74.87	76.56	76.17	71.22	74.74	75.00	74.22	**77.08**
Ecoli	72.32	80.95	83.33	74.70	82.74	80.66	83.63	82.44	**84.23**	53.87
Glass	56.08	66.82	74.30	62.62	71.96	66.82	73.36	65.42	**75.70**	66.82
Hepatitis	81.29	80.64	81.94	**85.16**	83.23	81.29	83.23	84.52	83.87	84.52
Iris	92.00	93.33	92.67	95.33	94.67	**96.00**	95.33	94.67	94.67	93.33
Liver-disorders	62.90	59.42	66.96	**71.30**	64.06	58.84	70.72	68.12	66.67	70.44
Lymph	81.08	79.05	**84.46**	81.76	83.78	81.76	83.11	79.73	81.76	**84.46**
Solar-flare-1	61.30	65.94	66.25	58.51	68.73	69.97	65.64	64.40	67.80	**70.90**
Solar-flare-2	64.54	70.73	72.05	71.95	71.11	70.36	71.58	70.36	71.39	**74.86**
Soybean	45.68	93.12	90.34	92.24	90.78	6.04	**93.85**	91.95	93.27	**93.85**
Vehicle	54.26	73.88	77.19	75.06	76.60	75.77	78.01	74.94	77.42	**78.49**
Vote	95.86	95.40	95.40	96.32	96.09	95.86	96.09	95.40	**97.01**	96.09
Vowel	34.75	99.09	97.37	98.18	98.38	96.97	98.08	94.95	98.89	**99.29**
Wine	92.70	98.88	98.88	97.19	98.88	98.31	98.88	**99.44**	98.88	97.75
Zoo	73.27	**97.03**	95.05	96.04	92.08	**97.03**	93.07	89.11	93.07	94.06

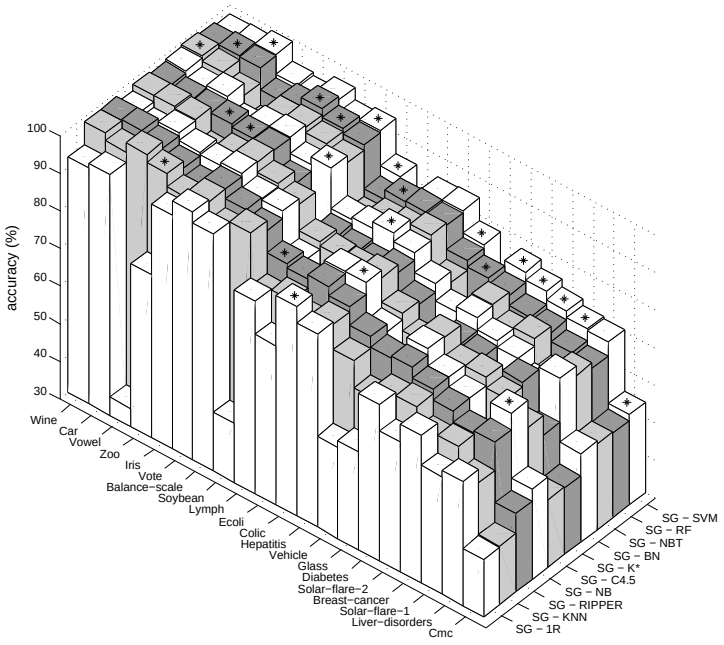

Figure 5: Stacked Generalization results obtained for each database and meta-classifier

the databases. We consider that this fact shows the effectiveness of our method and that it is not related with the type of classifier that is used as meta-classifier.

In Figures 7 and 8 comparisons of CSS, SG and single classifiers are shown for each database. These figures show the best results obtained for each base classifier (single) and using this classifier type as meta (SG, CSS).

To emphasize the differences between paradigms, Figure 9 shows a comparison of best results (maximum accuracy) obtained for each database. As can be seen, CSS paradigm outperforms the others in all used datasets except breast-cancer. It is worth noting that the aim of this paper is to present a new competitive approach; it is not the authors' intention to show a MSC that always sets the best results, although this has been obtained with the selected

Table 5: Results obtained by the proposed approach

Dataset	1R	KNN	RIPPER	NB	C4.5	K*	BN	NBT	RF	SVM
Balance-scale	93.12	95.04	95.20	93.92	94.72	95.20	93.44	95.68	**96.48**	94.24
Breast-cancer	74.13	74.48	75.17	**75.52**	75.17	74.83	73.43	**75.52**	**75.52**	74.48
Car	93.92	99.13	99.48	98.15	99.48	99.42	98.78	99.42	**99.77**	98.67
Cmc	48.47	52.00	53.90	**55.67**	53.23	53.16	55.53	55.13	52.68	55.33
Colic	86.69	85.33	**87.77**	86.69	**87.77**	87.23	86.96	87.23	85.60	86.14
Diabetes	77.34	77.60	78.13	78.00	78.52	**78.65**	76.82	78.00	78.13	77.47
Ecoli	75.89	86.01	86.61	84.23	86.91	86.61	85.42	84.82	87.80	**88.09**
Glass	64.49	75.23	78.04	73.83	77.57	78.97	78.50	72.43	**79.91**	79.44
Hepatitis	85.81	85.16	**87.74**	86.45	**87.74**	85.81	85.81	**87.74**	**87.74**	85.16
Iris	96.00	**97.33**	96.67	96.67	96.67	**97.33**	96.67	96.67	96.67	96.67
Liver-disorders	71.01	68.70	72.46	**73.04**	71.88	71.30	71.88	71.59	70.44	72.75
Lymph	83.78	86.49	86.49	87.16	86.49	85.81	**88.51**	85.14	87.16	87.84
Solar-flare-1	63.78	**75.23**	73.06	70.59	73.99	74.30	73.06	73.38	73.99	73.99
Solar-flare-2	66.04	75.05	73.64	73.45	75.23	75.33	72.89	74.39	75.33	**76.36**
Soybean	50.22	94.14	93.70	93.56	93.41	93.41	92.83	91.65	**94.44**	**94.44**
Vehicle	63.00	77.54	78.72	77.31	78.49	78.84	78.84	77.19	**80.38**	79.79
Vote	96.78	96.78	96.78	97.01	96.32	97.01	**97.24**	**97.24**	**97.24**	96.55
Vowel	37.98	**99.50**	99.29	99.09	99.29	99.19	98.28	98.79	**99.50**	99.39
Wine	92.70	**100.00**	**100.00**	**100.00**	**100.00**	99.44	**100.00**	**100.00**	98.88	**100.00**
Zoo	75.25	**98.02**	96.04	**98.02**	97.03	**98.02**	95.05	95.05	**98.02**	**98.02**

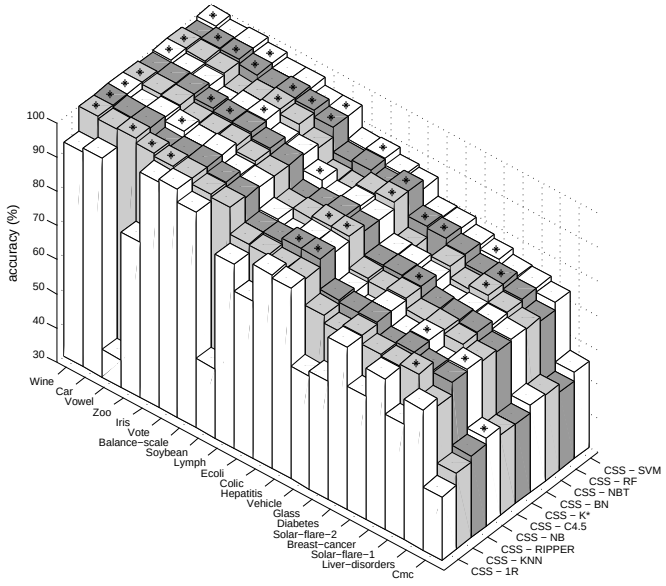

Figure 6: Results obtained by the proposed approach for each database and meta-classifier

datasets.

Finally, to have a more accurate idea of the kind of classifiers subsets selected by CSS method, Table 7 shows the configurations that give the best results for each dataset.

4.5.5. Statistical tests

According to [65], we have used the Iman-Davenport test to detect statistical differences among the different strategies. This test rejects the null hypothesis of equivalence between algorithms, since p-value (2.2e-16) is lower than our α-value (0.1). Thus, we have applied Shaffer post-hoc test in order to find out which algorithms are distinctive among them. Table 8 shows the statistical differences that our approach has obtained with the rest of methods. The table shows that when RF or SVM are used as meta-classifier in CSS (CSS-RF and CSS-SVM) best results are

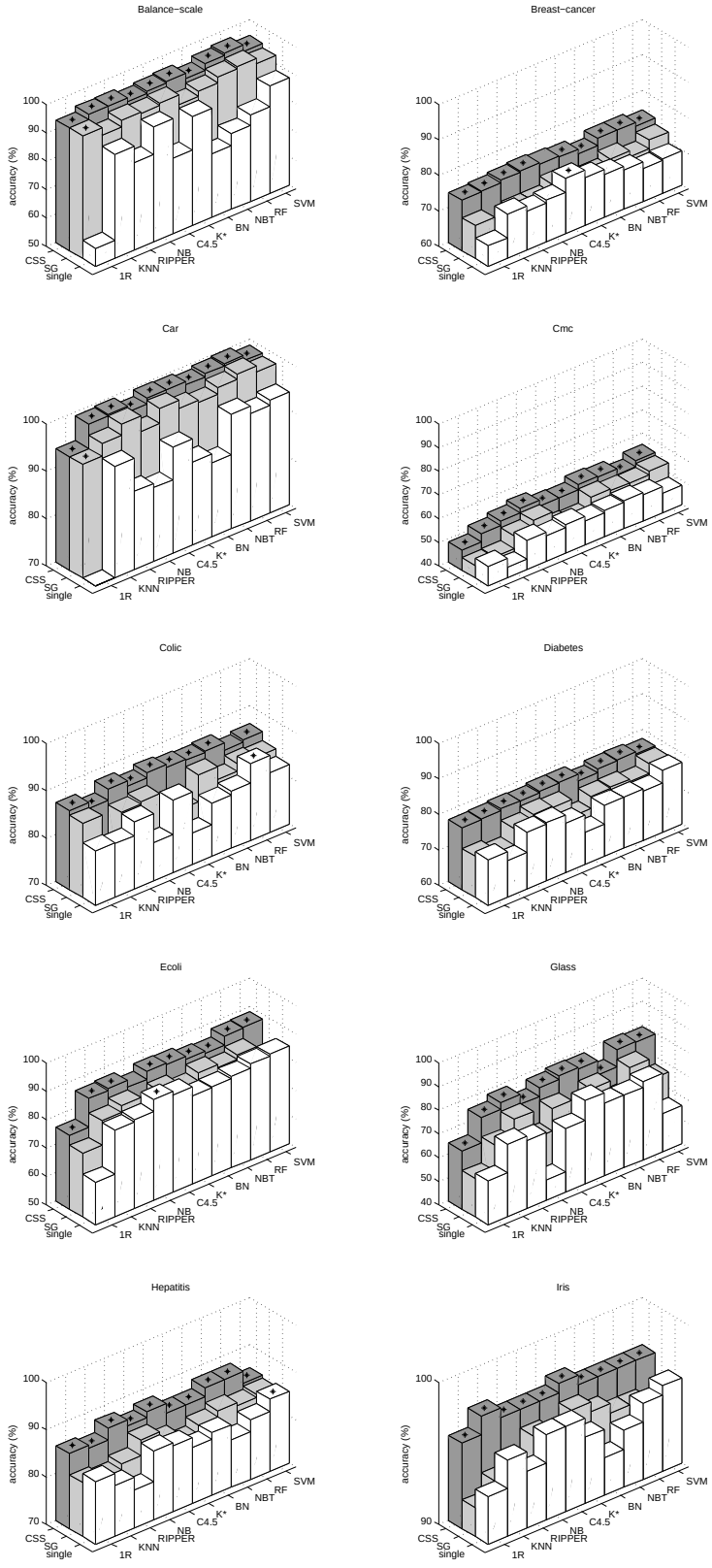

Figure 7: Comparison of CSS, SG and single classifiers for each database with different meta-classifiers.

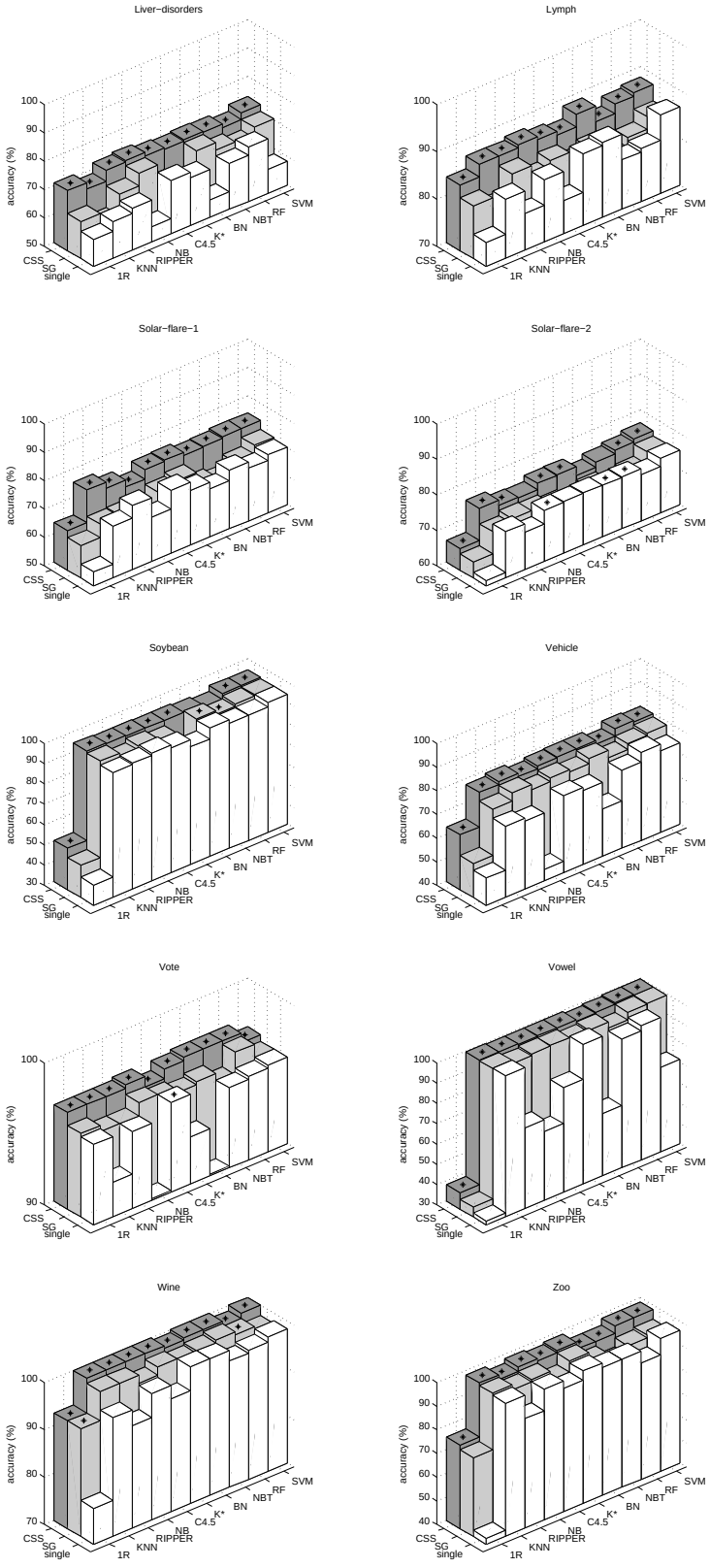

Figure 8: Comparison of CSS, SG and single classifiers for each database with different meta-classifiers

Table 6: Mean and Standard Deviation results obtained for each of the classifier sets used: single, ensemble, Stacked Generalization and CSS

Classifier Set	Single		Ensemble		Stacking		CSS	
Dataset	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev
Balance-scale	79.63	10.13	79.04	4.79	92.54	1.46	**94.70**	**1.04**
Breast-cancer	71.15	2.63	70.54	2.01	69.44	2.63	**74.83**	**0.70**
Car	88.18	7.30	87.83	12.01	97.38	**1.69**	**98.62**	1.72
Cmc	49.99	2.47	50.46	5.41	50.39	3.51	**53.51**	**2.20**
Colic	81.90	2.98	83.76	1.90	83.29	2.48	**86.74**	**0.84**
Diabetes	73.82	2.59	73.83	0.97	74.15	2.39	**77.86**	**0.56**
Ecoli	80.83	5.86	78.50	9.39	77.89	9.32	**85.24**	**3.50**
Glass	65.84	8.58	65.30	13.72	67.99	6.01	**75.84**	**4.70**
Hepatitis	82.32	2.20	83.71	1.43	82.97	1.59	**86.52**	**1.12**
Iris	94.73	1.19	94.67	0.94	94.20	1.30	**96.73**	**0.38**
Liver-disorders	62.75	5.11	70.43	2.97	65.94	4.52	**71.51**	**1.27**
Lymph	81.42	3.92	78.38	2.87	82.09	1.86	**86.49**	**1.35**
Solar-flare-1	67.86	5.07	65.87	12.78	65.94	3.82	**72.54**	**3.31**
Solar-flare-2	72.56	4.15	68.83	10.26	70.89	**2.58**	**73.77**	2.91
Soybean	86.59	16.46	75.29	31.68	79.11	29.60	**89.18**	**13.71**
Vehicle	66.29	10.69	66.28	17.66	74.16	7.15	**77.01**	**5.03**
Vote	94.09	2.44	95.86	0.38	95.95	0.50	**96.90**	**0.31**
Vowel	76.69	21.60	71.99	36.51	91.60	20.01	**93.03**	**19.35**
Wine	94.44	6.38	94.52	2.12	97.98	**1.97**	**99.10**	2.28
Zoo	88.12	16.34	72.77	25.65	91.98	7.01	**94.85**	**7.00**

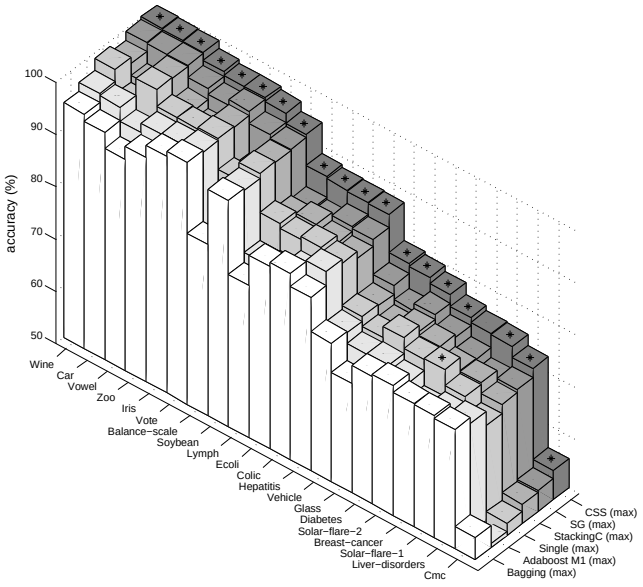

Figure 9: Propossed approach versus best results obtained by the remaining paradigms

obtained, closely followed by CSS-K* and CSS-C4.5. In the case of CSS-RF and CSS-SVM, they significantly improve most of the cases and draw with only 3 of them: StackingC, SG-RF and SG-SVM. On the other hand, CSS-1R obtains the worst results because it draws with all the cases. We also want to emphasize that our algorithm never gets worse results.

In Table 8 it can be seen that CSS does not significantly improve StackingC, SG-RF and SG-SVM with any of the meta-classifiers. Because of that we have compared these 3 methods with our approach applying Wilcoxon signed-rank test by pairwise. Table 9 shows the p-values obtained, where "+" symbol implies that the CSS is statistically better than the confronting one, whereas "=" means that there are not significant differences between the compared algorithms and "-" means that the CSS is statistically worse. It can be seen that, with the exception of 1R, the rest of

Table 7: Configuration of the best multi-classifiers obtained for each dataset

Dataset	Meta	Base classifiers
Balance-scale	RF	1R, RIPPER, NB, K*, BN, SVM
Breast-cancer	NB	1R, KNN, RIPPER, NB, K*, BN, NBT, RF, SVM
	NBT	1R, KNN, RIPPER, NB, C4.5, BN, NBT, RF
	RF	1R, KNN, RIPPER, NB, BN, RF
Car	RF	1R, KNN, NB, C4.5, K*
Cmc	NB	1R, RIPPER, C4.5, K*, NBT, SVM
Colic	RIPPER	KNN, K*, BN, NBT, RF
	C4.5	NBT, RF, SVM
Diabetes	K*	RF, SVM
Ecoli	SVM	1R, NB, K*,RF
Glass	RF	1R, KNN, C4.5, K*
Hepatitis	RIPPER	KNN, RIPPER, NB, SVM
	C4.5	RIPPER, NB, BN, NBT,SVM
	NBT	1R, KNN, RIPPER, NB, RF, SVM
	RF	1R, NB, BN, RF, SVM
Iris	KNN	1R, KNN, NB, C4.5, RF, SVM
	K*	KNN, NB, C4.5, K*, BN, SVM
Liver-disorders	NB	KNN, RIPPER,C4.5,RF, SVM
Lymph	BN	KNN, NB, C4.5, K*, BN, NBT, RF, SVM
Solar-flare-1	KNN	1R,SVM
Solar-flare-2	SVM	KNN, NB, NBT, RF, SVM
Soybean	RF	1R, RIPPER, C4.5, K*, BN, NBT, SVM
	SVM	NB, RF, SVM
Vehicle	RF	1R, NB, C4.5, K*, BN, NBT, RF
Vote	BN	C4.5, K*, NBT, RF, SVM
	NBT	KNN, RIPPER,C4.5, RF
	RF	1R, KNN, NB, C4.5, NBT, RF, SVM
Vowel	KNN	KNN, NB, C4.5, K*, SVM
	RF	KNN, NB, C4.5, K*, RF
Wine	KNN	RIPPER, NB, BN, NBT, RF
	NB	1R, RIPPER, NB, K*, NBT, RF, SVM
	C4.5	RIPPER, NB, C4.5, RF, SVM
	BN	1R, C4.5, K*, BN,RF, SVM
	NBT	1R, KNN, NB, C4.5, K*, RF, SVM
	RF	C4.5, K*, BN, RF
	SVM	1R, KNN, RIPPER, C4.5, BN, SVM
Zoo	KNN	1R, KNN, RIPPER, NB, C4.5, RF, SVM
	NB	KNN, K*, BN, NBT, SVM
	K*	KNN, RIPPER, NB, C4.5, BN, NBT, SVM
	RF	NB, C4.5
	SVM	RIPPER, NB, BN, NBT, RF

meta-classifiers significantly improve the rest of algorithms. Viewing the results of both statistical tests, considerably demonstrates the strength of our new method.

5. Conclusions and future work

In this paper an evolutionary computation based classifier subset selection process is presented to construct a Multiple-Classifier System. To this end, ten base classifiers have been selected to construct an MCS. Stacked Generalization is the model used, although other possibilities can be considered as well: voting, hierarchical classifiers, etc.

Estimation of Distribution Algorithm is the evolutionary computation algorithm selected to perform the Classifier Subset Selection, but other approaches could be used: Genetic algorithms, Ant, Colony, etc.

The obtained experimental results are very good, and a better or equal result has been obtained by using the proposed approach, compared to other state-of-the-art paradigms. It is not the aim of the authors to present an MCS better then existing ones, but a competitive one. The results obtained in these datasets indicate the validity of the approach, but certainly there would be some other datasets in which the results are worse than those obtained by other paradigms.

As future work, the performance of the presented approach to a real problem is going to be investigated, and compared to other models. More base classifiers can be included as well to improve the MCS accuracy.

Table 8: Significant differences obtained with Shaffer post-hoc test

	Individual			Ensemble			Stacked Generalization		
CSS	win	draw	loose	win	draw	loose	win	draw	loose
RF	ALL	-	-	BAG-C4.5 BAG-REP BOS-DS BOS-C4.5	STC	-	1R KNN RIP NB C4.5 K* BN NBT	RF SVM	-
SVM	ALL	-	-	BAG-C4.5 BAG-REP BOS-DS BOS-C4.5	STC	-	1R KNN RIP NB C4.5 K* BN NBT	RF SVM	-
K*	ALL	-	-	BAG-C4.5 BAG-REP BOS-DS BOS-C4.5	STC	-	1R KNN RIP NB C4.5 K* NBT	BN RF SVM	-
C4.5	ALL	-	-	BAG-C4.5 BAG-REP BOS-DS BOS-C4.5	STC	-	1R KNN RIP NB C4.5 K* NBT	BN RF SVM	-
RIP	ALL	-	-	BAG-REP BOS-DS BOS-C4.5	STC BAG-C4.5	-	1R KNN RIP NB C4.5 K* NBT	BN RF SVM	-
KNN	1R KNN RIP NB C4.5 K* BN NBT RF	SVM	-	BAG-REP BOS-DS BOS-C4.5	STC BAG-C4.5	-	1R KNN RIP C4.5 K* NBT	NB BN RF SVM	-
NB	1R KNN RIP NB C4.5 K* BN NBT RF	SVM	-	BAG-REP BOS-DS BOS-C4.5	STC BAG-C4.5	-	1R KNN RIP C4.5 K* NBT	NB BN RF SVM	-
NBT	1R KNN RIP NB K* BN NBT RF	C4.5 SVM	-	BAG-REP BOS-DS BOS-C4.5	STC BAG-C4.5	-	1R KNN RIP C4.5 K* NBT	NB BN RF SVM	-
BN	1R KNN RIP NB K* BN NBT RF	C4.5 SVM	-	BAG-REP BOS-DS BOS-C4.5	STC BAG-C4.5	-	1R KNN K* NBT	RIP NB C4.5 BN RF SVM	-
1R	-	ALL	-	-	ALL	-	-	ALL	-

Table 9: The p-values obtained with Wilcoxon test

CSS	SG-RF	SG-SVM	STC
1R	-(0.01923)	=(0.10540)	-(0.02148)
KNN	+(0.00032)	+(0.00365)	+(0.00584)
RIPPER	+(0.00001)	+(0.00058)	+(0.00058)
NB	+(0.00401)	+(0.00199)	+(0.00745)
C4.5	+(0.00027)	+(0.00068)	+(0.00008)
K*	+(3.8E-006)	+(0.00010)	+(0.00008)
BN	+(0.00032)	+(0.00315)	+(0.00745)
NBT	+(0.00233)	+(0.00121)	+(0.02944)
RF	+(0.00014)	+(0.00050)	+(0.00005)
SVM	+(0.00004)	+(1.9E-006)	+(1.9E-006)

Based on this work, another approach is planned that takes into account the diversity among the different base classifiers for each classification problem, and selects the classifier subset which increases, in a validation phase, the obtained accuracy considering the different classifications given to each case.

Acknowledgment

The work described in this paper was partially conducted within the Basque Government Research Team grant and the University of the Basque Country UPV/EHU and under grant UFI11/45 (BAILab).

References

References

[1] T. Dietterich, Machine Learning Research: Four Current Directions, Artificial Intelligence Magazine 18 (4) (1997) 97–136.

[2] L. I. Kuncheva, An experimental study on diversity for bagging and boosting with linear classifiers, Information Fusion 3 (4) (2002) 245 – 258.

[3] L. I. Kuncheva, "fuzzy" versus "nonfuzzy" in combining classifiers designed by boosting, IEEE T. Fuzzy Systems 11 (6) (2003) 729–741.

[4] L. I. Kuncheva, J. J. Rodríguez, Classifier ensembles with a random linear oracle, IEEE Trans. Knowl. Data Eng. 19 (4) (2007) 500–508.

[5] L. K. Hansen, P. Salamon, Neural network ensembles, IEEE Trans. Pattern Anal. Mach. Intell. 12 (10) (1990) 993–1001.

[6] J. M. Martínez-Otzeta, B. Sierra, E. Lazkano, A. Astigarraga, Classifier hierarchy learning by means of genetic algorithms, Pattern Recognition Letters 27 (16) (2006) 1998–2004.

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

[8] T. G. Dietterich, G. Bakiri, Solving multiclass learning problems via error-correcting output codes, Journal of Artificial Intelligence Research 2.

[9] J. Fürnkranz, Round robin classification, The Journal of Machine Learning Research 2 (2002) 721–747.

[10] Y. Freund, R. E. Schapire, A short introduction to boosting, Journal of Japanese Society for Artificial Intelligence 14 (5) (1999) 771–780.

[11] L. Breiman, Bagging predictors, Machine Learning 24 (2) (1996) 123–140.

[12] D. Wolpert, Stacked generalization, Neural Networks 5 (1992) 241–259.

[13] B. Sierra, N. Serrano, P. Larrañaga, E. J. Plasencia, I. Jiménez, P. Revuelta, M. L. Mora, Using Bayesian networks in the construction of a bi-level multi-classifier. A case study using intensive care unit patients data, Artificial Intelligence in Medicine 22 (3) (2001) 233–248. doi:http://dx.doi.org/10.1016/S0933-3657(00)00111-1.
URL http://www.sciencedirect.com/science/article/pii/S0933365700001111

[14] J. Kittler, M. Hatef, R. P. Duin, J. Matas, On combining classifiers, Pattern Analysis and Machine Intelligence, IEEE Transactions on 20 (3) (1998) 226–239.

[15] G. Fumera, F. Roli, Performance analysis and comparison of linear combiners for classifier fusion, in: Structural, Syntactic, and Statistical Pattern Recognition, Springer, 2002, pp. 424–432.

[16] L. I. Kuncheva, A theoretical study on six classifier fusion strategies, Pattern Analysis and Machine Intelligence, IEEE Transactions on 24 (2) (2002) 281–286.

[17] N. García-Pedrajas, D. Ortiz-Boyer, An empirical study of binary classifier fusion methods for multiclass classification, Information Fusion 12 (2) (2011) 111–130.

[18] A. Bella, C. Ferri, J. Hernández-Orallo, M. J. Ramírez-Quintana, On the effect of calibration in classifier combination, Applied Intelligence (2013) 1–20.

[19] P. Larrañaga, J. Lozano, Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation, Kluwer Academic Press, 2001.

[20] K. Bache, M. Lichman, UCI machine learning repository (2013).
URL http://archive.ics.uci.edu/ml

[21] L. I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, John Wiley and Sons, Inc., 2004.

[22] L. Oliveira, U. Nunes, P. Peixoto, On exploration of classifier ensemble synergism in pedestrian detection, IEEE Transactions on Intelligent Transportation Systems 11 (1) (2010) 16–27.

[23] U. Maulik, D. Chakraborty, A robust multiple classifier system for pixel classification of remote sensing images, Fundam. Inf. 101 (4) (2010) 286–304.

[24] F. Keyvanfard, M. Shoorehdeli, M. Teshnehlab, K. Nie, M.-Y. Su, Specificity enhancement in classification of breast mri lesion based on multi-classifier, Neural Computing and Applications 22 (1) (2013) 35–45.

[25] J. Du, J. Guo, S. Wang, X. Zhang, Multi-classifier combination for translation error detection, in: Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data, Vol. 8202 of Lecture Notes in Computer Science, 2013, pp. 291–302.

[26] L. B. Batista, S. Ratte, A multi-classifier system for sentiment analysis and opinion mining, in: Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012), IEEE Computer Society, 2012, pp. 96–100.

[27] F. Keyvanfard, M. A. Shoorehdeli, M. Teshnehlab, K. Nie, M.-Y. Su, Specificity enhancement in classification of breast mri lesion based on multi-classifier, Neural Computing and Applications (2012) 1–11.

[28] M. Ferrara, A. Franco, D. Maio, A multi-classifier approach to face image segmentation for travel documents, Expert Systems with Applications 39 (9) (2012) 8452–8466.

[29] Y. Haibo, Z. Hongling, W. Zongmin, Remote sensing classification based on hybrid multi-classifier combination algorithm, in: Audio Language and Image Processing (ICALIP), 2010 International Conference on, IEEE, 2010, pp. 1688–1692.

[30] F. Roli, G. Giacinto, Design of multiple classifier systems (2002).

[31] F. Glover, G. Kochenberger, Handbook of Metaheuristics, Kluwer, 2003.

[32] K. M. Ting, I. H. Witten, Issues in stacked generalization, J. Artif. Int. Res. 10 (1) (1999) 271–289.
URL http://dl.acm.org/citation.cfm?id=1622859.1622868

[33] A. Seewald, How to make stacking better and faster while also taking care of an unknown weakness, in: C. Sammut, A. Hoffmann (Eds.), Nineteenth International Conference on Machine Learning, Morgan Kaufmann Publishers, 2002, pp. 554–561.

[34] A. Ekbal, S. Saha, Stacked ensemble coupled with feature selection for biomedical entity extraction, Knowledge-Based Systems.

[35] A. Ibarguren, I. Maurtua, B. Sierra, Layered architecture for real time sign recognition: Hand gesture and movement, Engineering Applications of Artificial Intelligence 23 (7) (2010) 1216–1228.

[36] C. Qian, Y. Yu, Z.-H. Zhou, An analysis on recombination in multi-objective evolutionary optimization, Artif. Intell. 204 (2013) 99–119.

[37] A. Rahman, B. Verma, Ensemble classifier generation using non-uniform layered clustering and genetic algorithm, Knowledge-Based Systems 43 (2013) 30–42.

[38] D. Impedovo, G. Pirlo, D. Barbuzzi, Multi-classifier system configuration using genetic algorithms (2012) 560–564.
[39] Y.-S. Ding, T.-L. Zhang, Using chou's pseudo amino acid composition to predict subcellular localization of apoptosis proteins: An approach with immune genetic algorithm-based ensemble classifier, Pattern Recognition Letters 29 (13) (2008) 1887 – 1892.
[40] Z.-H. Zhou, J. Wu, W. Tang, Ensembling neural networks: many could be better than all, Artificial intelligence 137 (1) (2002) 239–263.
[41] M.-J. Kim, D.-K. Kang, Classifiers selection in ensembles using genetic algorithms for bankruptcy prediction, Expert Systems with Applications 39 (10) (2012) 9308 – 9314.
[42] Y. Chen, M. L. Wong, Optimizing stacking ensemble by an ant colony optimization approach, in: Proceedings of the 13th annual conference companion on Genetic and evolutionary computation, GECCO '11, ACM, New York, NY, USA, 2011, pp. 7–8. doi:10.1145/2001858.2001863.
URL http://doi.acm.org/10.1145/2001858.2001863
[43] Y. Chen, M.-L. Wong, Applying ant colony optimization in configuring stacking ensemble, in: Soft Computing and Intelligent Systems (SCIS) and 13th International Symposium on Advanced Intelligent Systems (ISIS), 2012 Joint 6th International Conference on, 2012, pp. 2111–2116. doi:10.1109/SCIS-ISIS.2012.6505018.
[44] P. Shunmugapriya, S. Kanmani, Optimization of stacking ensemble configurations through artificial bee colony algorithm, Swarm and Evolutionary Computation (0) (2013) –.
[45] A. Ledezma, R. Aler, A. SanchÃs, D. Borrajo, Ga-stacking: Evolutionary stacked generalization., Intell. Data Anal. 14 (1) (2010) 89–119.
[46] B. Sierra, N. Serrano, P. Larrañaga, E. J. Plasencia, I. Inza, J. J. Jiménez, P. Revuelta, M. L. Mora, Using bayesian networks in the construction of a bi-level multi-classifier. A case study using intensive care unit patients data., Artificial Intelligence in Medicine 22 (3) (2001) 233–248.
[47] D. W. Aha, R. L. Bankert, Feature selection for case-based classification of cloud types: An empirical comparison, in: Proceedings of the AAAI'94 Workshop on Case-Based Reasoning, 1994, pp. 106–112.
[48] I. Inza, P. Larrañaga, R. Etxeberria, B. Sierra, Feature Subset Selection by Bayesian network-based optimization, Artificial Intelligence 123 (1-2) (2000) 157–184.
[49] R. Etxeberria, P. Larranaga, Global optimization using bayesian networks, Proceedings of the Second Symposium on Artificial Intelligence (CIMAF-99) (1999) 332–339.
[50] I. Inza, P. Larrañaga, B. Sierra, Feature subset selection by bayesian networks: a comparison with genetic and sequential algorithms, International Journal of Approximate Reasoning 27 (2) (2001) 143–164.
[51] C. Echegoyen, A. Mendiburu, R. Santana, J. A. Lozano, Toward understanding edas based on bayesian networks through a quantitative analysis, Evolutionary Computation, IEEE Transactions on 16 (2) (2012) 173–189.
[52] M. Hall, E. Frank, G. Holmes, B. P. P., Reutemann, I. Witten, The weka data mining software: an update, SIGKDD Explor. Newsl. 11 (1) (2009) 10–18.
[53] R. Holte, Very simple classification rules perform well on most commonly used datasets, Machine Learning 11 (1993) 63–91.
[54] D. Aha, D. Kibler, M. Albert, Instance-based learning algorithms, Machine Learning 6 (1991) 37–66.
[55] W. Cohen, Fast effective rule induction, in: 12th International Conference on Machine Learning, Morgan Kaufmann, 1995, pp. 115–123.
[56] B. Cestnik, Estimating probabilities: a crucial task in machine learning, in: Proceedings of the European Conference on Artificial Intelligence, 1990, pp. 147–149.
[57] J. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, 1993.
[58] J. Cleary, L. Trigg, K*: An instance-based learner using an entropic distance measure, in: 12th International Conference on Machine Learning, 1995, pp. 108–114.
[59] B. Sierra, E. Lazkano, E. Jauregi, I. Irigoien, Histogram distance-based bayesian network structure learning: A supervised classification specific approach, Decision Support Systems 48 (1) (2009) 180–190.
[60] R. Kohavi, Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid, in: Second International Conference on Knoledge Discovery and Data Mining, 1996, pp. 202–207.
[61] L. Breiman, Random forests, Machine Learning 45 (1) (2001) 5–32.
[62] D. Meyer, F. Leisch, K. Hortnik, The support vector machine under test, Neurocomputing 55 (1) (2003) 169–186.
[63] M. Stone, Cross-validation choice and assessment of statistical procedures, Journal Royal of Statistical Society 36 (1974) 111–147.
[64] W. Buntine, Theory refinement on bayesian networks, Morgan Kaufmann, 1991, pp. 52–60.
[65] S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power, Information Sciences 180 (10) (2010) 2044–2064.