Ensemble Particle Swarm Model Selection

Hugo Jair Escalante, *Member IEEE*, Manuel Montes, *Member IEEE*, and Enrique Sucar, *Senior Member IEEE*

***Abstract*— This paper elaborates on the benefits of using particle swarm model selection (PSMS) for building effective ensemble classification models. PSMS searches in a toolbox for the best combination of methods for preprocessing, feature selection and classification for generic binary classification tasks. Throughout the search process PSMS evaluates a wide variety of models, from which a single solution (i.e. the best classification model) is selected. Satisfactory results have been reported with the latter formulation in several domains. However, many models that are potentially useful for classification are disregarded for the final model. In this paper we propose to re-use such candidate models for building effective ensemble classifiers. We explore three simple formulations for building ensembles from intermediate PSMS solutions that do not require of further computation than that of the traditional PSMS implementation. We report experimental results on benchmark data as well as on a data set from object recognition. Our results show that better models can be obtained with the ensemble version of PSMS, motivating further research on the combination of candidate PSMS models. Additionally, we analyze the diversity of the classification models, which is known to be an important factor for the construction of ensembles.**

I. Introduction

Ensemble classifiers are predictive models build upon the combination of multiple classification methods. These models are very popular because of their ability for improving the performance and stability of individual models. From previous work on multiple classifier systems (e.g. [1], [2], [3]), we know that the success of committee methods mainly depends on two factors, namely, the performance and diversity of individual models. Hence, by generating accurate enough individual models and trying to make them as diverse as possible we can obtain better predictions than when using any of the individual models. The selection of accurate classification models is known as the model selection problem [4]. Therefore, a natural approach for building ensembles is to rely on model selection techniques for obtaining effective individual models, while ensuring that the diversity among the individual models is acceptable.

The complications of the above formulation is that traditional model selection schemes are able to optimize the performance of individual models only and do not consider the diversity among sets of methods; thus additional strategies must be adopted (e.g. subsampling). Also, creating an ensemble with $N-$members, requires the application of the model selection strategy $N-$times, which can be computationally expensive. Therefore this formulation seems to be very limited. Nevertheless, there are a sort of model selection strategies (population based methods) that despite they select individual models, they are able to generate diverse and accurate individual models through the model selection process. We argue that we can take advantage of such sort of model selection methods for constructing effective ensemble classifiers without increasing the computational cost of the optimization procedure.

Particle swarm model selection (PSMS) is a recently proposed technique for the selection of effective (individual) classification models for generic domains [5]. Given a binary classification problem PSMS searches for the best combination of methods for preprocessing, feature selection and classification from a predefined set of methods that are available in a machine learning toolbox; PSMS also optimizes the parameters of the considered methods according to the available data. Since the search space that PSMS explores is composed of many heterogenous models, PSMS evaluates throughout its search process a broad diversity of methods from which a single solution (i.e. the best classification model) is selected. Satisfactory results have been reported with PSMS on a variety of domains [6], [5], [7]. However, many of the evaluated models that are potentially useful for the classification problem are disregarded for the final model. Our hypothesis in this work is that we can take advantage of the variety of models evaluated by PSMS for building ensemble classifiers that can outperform the (single) best solution as selected with traditional PSMS.

In this paper we study the suitability of PSMS for building ensemble classifiers. We explore three strategies for generating ensembles from PSMS's partial solutions (i.e. candidate classification models). Under these formulations no extra computation is required, although the classification accuracy can be significantly improved. We conduct experiments on benchmark (machine learning) data as well as on an object recognition data set and we evaluate both accuracy and diversity of individual models. Experimental results show that PSMS can be very helpful for building ensembles, as the ensemble version improves the performance of traditional PSMS. The ensemble version of PSMS is also able to improve the stability of predictions from models selected with PSMS. An interesting finding is that models as evaluated by PSMS can provide of highly diverse models, thus motivating the development of more elaborated strategies for building ensembles.

A. Related work

The underlying idea of ensemble methods is that by considering multiple views of the same problem we can obtain more accurate and more robust predictions. The latter fact is justified by theoretic and empirical studies that have shown that, under certain conditions, the combination of

The authors are with the Department of Computer Science, at the National Institute of Astrophysics, Optics and Electronics (INAOE), Tonantzintla, Puebla, 72840, Mexico (email: hugojair@inaoep.mx).

978-1-4244-8126-2/10/$26.00 ©2010 IEEE

multiple individual models is beneficial in terms of accuracy and stability of predictors [1]. However, despite that the ensemble learning paradigm has been studied for more than two decades, there are still open some issues that deserve further study. One of such issues is for example on how to select the set of classifiers for creating an ensemble.

Previous studies suggest that the effectiveness of ensembles depends on the accuracy and diversity of individual models [2], [1], [3], [8]. Accordingly, successful ensemble methods attempt to guarantee (at least) *default accuracy* and high diversity by adopting diverse strategies; For example, learning weights for weak learners [9]; randomizing the sets of features and instances that are considered for each classifier [10], [11]; partitioning the input space into clusters and learning different classifiers for the different clusters [12]; determining the most appropriate classifier (from a predefined set) for each test instance according to distance measures [13] or using different learning algorithms for each individual model [14], [3], [15], [16], [17]. The latter strategy, often called heterogeneous ensembles, is mostly related to our work.

Heterogenous ensembles are based on the assumption that since different learning algorithms have different biases, their decision functions will be different, which may lead to obtain high diversity among the individual models. However, a problem with this sort of ensembles is that it is not clear how to select what learning algorithms are to be considered for an ensemble. Some researchers have adopted diverse search strategies for the selection of a set of models so that the performance of the ensemble under a certain fusion strategy is optimized [14], [3], [15], [16], [17]. They consider a pool of classification algorithms and by using combinatorial optimization techniques they attempt to select the combination of methods that maximizes the performance of the ensemble [18], [19], [15]. Some researchers also attempt to optimize the weighs by which each of the considered methods contributes to the ensemble [17], [16]. However, under the latter approach the parameters of the learning algorithms are fixed and hence the classifiers are not really optimized for the individual problems; additionally, the same data preprocessing methods and the same feature selection techniques are used for all of the models that are considered in the ensemble. Thus reducing the potential diversity of the members of the ensemble.

In this paper we explore the use of a full model selection strategy for building ensemble classifiers. Full model selection techniques aim at selecting the best combination of methods for preprocessing, feature selection and classification starting from a training data set [6], [5], [20]. The main benefits of such methods is that the work job the data analyst is simplified and very effective classification models can be selected without spending time on the design and development of specific models for different data sets. Therefore, our hypothesis is that by adopting full model selection techniques we can obtain effective classification models that also can offer a high degree of diversity because of the heterogeneity of models.

To the best of our knowledge there are only two full model selection strategies that have developed so far [5], [20]. On the one hand, Gorissen et al. have used genetic algorithms for model type selection [20], on the other hand, Escalante et al. have used particle swarm optimization for full model selection (i.e., PSMS) [5]. Despite both techniques have reported satisfactory results in this work we considered PSMS because it has been applied to high dimensional data sets without any problem, whereas the genetic algorithm approach have been applied on low dimensional data only. One should note that Gorissen et al. have already considered the use of ensembles to cross-over models of different types [20]; thus, they can select an ensemble as the final model. We believe that the performance of the final model can be further improved in both Gorrisen et al's and Escalante et al.'s approaches by merging the output of some models evaluated through the search. Thus in future work we will explore the suitability of Gorissen et al.'s method, under our approach, for building ensembles.

The rest of this paper is organized as follows. The next section describes PSMS. Section III elaborates on the suitability of PSMS for building ensembles and introduces the strategies we propose for building ensembles from PSMS solutions. Section IV reports experimental results of our methods. Section V summarizes our findings and outlines future work directions.

II. Particle swarm model selection

This section describes PSMS, a generic technique for the selection of individual classification models. In a nutshell PSMS can be considered a black-box tool that receives as input a training data set for binary classification and returns as output a full classification model. Given a machine learning toolbox, PSMS selects the best combination of methods for preprocessing, feature selection and classification; additionally, PSMS optimizes parameters of the selected methods. PSMS explores the classifiers space by means of particle swarm optimization (PSO), which attempts to select the model that minimizes the classification error using training data.

PSO is a bio-inspired search technique that has proved to be very effective in several domains [21]. The algorithm mimics the behavior of biological societies that share goals and present local and social behavior. Solutions are called particles, at each iteration t, each particle i has a position in the search space $\mathbf{x}_i^t =< x_{i,1}^t, \ldots, x_{i,d}^t >$, and a velocity $\mathbf{v}_i^t =< v_{i,1}^t, \ldots, v_{i,d}^t >$ value, with d the dimensionality of the problem. The PSO algorithm that we consider is described in Algorithm 1.

At the beginning a population of $m-$particles (i.e. the swarm) is randomly initialized; next an iterative process starts where particles update their positions in the search space as follows:

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^{t+1} \quad (1)$$

$$\mathbf{v}_i^{t+1} = w \times v_i^t + c_1 \times r_1 \times (\mathbf{p}_i - \mathbf{x}_i^t) + c_2 \times r_2 \times (\mathbf{g}^t - \mathbf{x}_i^t) \quad (2)$$

Algorithm 1 Particle swarm optimization.

Require:
– $\mathbf{c}_1, \mathbf{c}_2$: weights for local and global information;
– $\mathbf{m}$: number of particles in the swarm;
– $\mathbf{I}_{max}$: number of iterations;
– $\mathbf{W}$: inertia weight
Initialize swarm ($\mathbf{S} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$)
Compute fitness function $f(\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\})$
Identify global best ($\mathbf{p}_g^t$) solution
Identify personal best solutions ($\mathbf{p}_{1,\ldots,m} = \mathbf{x}_{1,\ldots,m}$)
$t = 1$
while $t < I_{max}$ **do**
  for all $\mathbf{x}_i \in \mathbf{S}$ **do**
    Calculate velocity $\mathbf{v}_i$ for $\mathbf{x}_i$ (Equation (2))
    Update position of $\mathbf{x}_i$ (Equation (1))
    Compute $f(\mathbf{x}_i)$
    Update $\mathbf{p}_i$ (*if needed*)
  end for
  Update $\mathbf{p}_g^t$ (*if needed*)
  Decrease $\mathbf{W}$
  t^{++}
end while
return $\mathbf{p}_g^t$

where $\mathbf{p}_i$ is the best position obtained by $\mathbf{x}_i$ (personal best), $\mathbf{p}_g^t$ is the best particle in the swarm up to iteration t (global best), c_1 and c_2 constants weighting the contribution of local and global solutions, whereas r_1, r_2 random numbers, W is the so called inertia term, which weights the contribution of the previous velocity into the new one, see [21] for details. The goodness of particles is evaluated with a fitness function ($f(\mathbf{x}_i)$) that is specific for the task at hand. PSO stops when a fixed number of iterations (I_{max}) is performed.

In PSMS the particles are full models (i.e., combinations of preprocessing, feature selection and classification methods), codified as numerical vectors. The optimization problem consists of minimizing an estimate of the classification errors that models would obtain on unseen data (i.e. maximizing the generalization performance). In particular, we considered the balanced error rate (BER) as fitness function; $BER = \frac{E_+ + E_-}{2}$, where E_+ and E_- are the error rates in the positive and negative classes, respectively. As the test data are unseen during training, the error of solutions (i.e., full models) is estimated with $k-$fold cross validation (CV) on the training set. Thus, the PSO algorithm is used to search for the model that minimizes the CV-BER; the single model that achieves the lowest CV-BER is returned as output.

PSMS has been used with the CLOP toolbox[1], the methods available in such toolbox are shown in Table I. Therefore, PSMS solutions are combinations of such techniques with different parameters settings; a sample (decodified) solution in PSMS is as follows:

[standardize($c = 1$),s2n($f = 8$),neural($u = 10$,$s = 0.5$,$iter = 10$)]

Under the above model the data is first standardized, next the *s2n* feature selection method is used for selecting at most $f = 8$ features, then a *neural* classifier with specific parameters is used for classification.

[1]http://clopinet.com/CLOP

TABLE I
CLASSIFICATION (C), FEATURE SELECTION (F) AND PREPROCESSING (P) METHODS CONSIDERED IN OUR EXPERIMENTS; WE SHOW THE OBJECT NAME AND THE NUMBER OF PARAMETERS FOR EACH METHOD.

Object name	Type	# pars.	Description
zarbi	C	0	Linear classifier
naive	C	0	Naïve Bayes
logitboost	C	3	Boosting with trees
neural	C	4	Neural network
svc	C	4	SVM classifier
kridge	C	4	Kernel ridge regression
rf	C	3	Random forest
lssvm	C	5	Kernel ridge regression
Ftest	F	4	F-test criterion
Ttest	F	4	T-test criterion
aucfs	F	4	AUC criterion
odds-ratio	F	4	Odds ratio criterion
relief	F	3	Relief ranking criterion
Pearson	F	4	Pearson correlation coefficient
ZFilter	F	2	Statistical filter
s2n	F	2	Signal-to-noise ratio
$pc - extract$	F	1	Principal components analysis
svcrfe	F	1	SVC- recursive feature elimination
normalize	P	1	Data normalization
standardize	P	1	Data standardization
$shift - scale$	P	1	Data scaling

PSMS has reported satisfactory results on diverse binary classification problems without requiring significant supervision [5], [7], [4]. The main benefits of PSMS is that (1) very effective models can be obtained, (2) no knowledge is required on machine learning nor on the application domain and (3) it can be applied to any binary classification problem. A disadvantage of PSMS is that it can be computationally expensive as many models must be trained and evaluated. Nevertheless, subsampling heuristics have been proposed for speeding up PSMS [5]. In this paper we take advantage of the large number of models that are evaluated through the search for the improving performance of models selected with PSMS.

III. Ensemble particle swarm model selection

PSO is a population-based heuristic technique in which solutions are not eliminated/created. Instead the initial population of solutions is maintained. At each iteration t the positions of the $m-$particles are updated by taking into account local ($\mathbf{p}_i$ in Equation (1)) and global information ($\mathbf{p}_g^t$ in Equation(2)); at the end of the search process (i.e. when $t = I_{max}$) the solution that obtains the best fitness value (i.e. $\mathbf{p}_g^{I_{max}}$) is returned as the selected model.

One should note, however, that after I_{max} iterations a total of $M = [(I_{max} + 1) \times m]$ solutions are evaluated by PSMS; from which a single solution is returned as the selected model. What is more, at the end of the search the optimized swarm provides us with $m-$candidate solutions, all of which are potentially useful, but again a single one is selected. Despite this approach lies at the core of PSO (i.e. the leader particle is the best solution), for PSMS the best single model can be one that has over-fitted the model selection criterion. Thus, even when the selected solution minimizes the CV-BER its performance on test data may be rather poor. Therefore, alternative strategies for selecting a single model from PSMS's candidate solutions must be

adopted. Note that the latter is a difficult task as the only indicator of classifier effectiveness is the CV-BER estimate.

An alternative solution, that we explore herein, is to combine the outputs of several of the models that are evaluated through the search process instead of selecting a single one. The motivation of this idea is the well known fact that, under certain conditions, the combination of multiple classifiers can result in better and more stable predictions. The main conditions that are required for the success of ensemble methods are related to the accuracy and diversity of the individual models that form the ensemble [2], [1], [3], [8]. Accordingly, our proposal is to identify, from the M solutions tried by PSMS, those models that seem to be accurate (according to the CV-BER criteria) and diverse (of heterogeneous nature) at the same time, and next using such models for building ensemble classifiers.

In traditional ensemble methods diversity can be achieved by pattern subsampling, feature subset selection, input space partitioning or by considering different learning algorithms for the ensemble members. Thus we must adopt at least one of the latter strategies for guaranteeing diversity in PSMS's partial solutions. Nevertheless, in PSMS the models that are considered through the search are based on different learning algorithms and different methods for data preprocessing and feature selection; additionally, the parameters of the different models may vary considerably. In consequence models considered by PSMS are very heterogeneous, our hypothesis is that such heterogeneity may be related to diversity (i.e. the ability of ensemble members of making uncorrelated errors) among models. In Section IV we evaluate our hypothesis experimentally, by measuring the diversity of models selected with PSMS.

Note that heterogeneity in models is guaranteed, to some extend, at the beginning of the search because the initial population in PSMS is generated under a uniform distribution over the models and parameters. As the search process goes on, particles will try to adapt their behavior according to both the current global and local minima. Hence at the end of the search it may be possible that most of the particles converge to similar classification models. For avoiding the latter issue we can modify PSMS's parameters so that we can control the impact that global solutions have into the generation of new solutions; this way, the heterogeneity of solutions can be maintained at the end of the search. For example, by setting c_2 (the weight for the global best solution) to a small value will produce that new updates of particles positions depend mostly on their own previous best solutions; setting W (the inertia weight) to zero will produce that previous velocities will have no influence on the generation of new solutions; also, running PSMS for a small number of iterations (I_{max}) will prevent PSMS of performing an intensive search that may lead to keeping models diverse.

Regarding the accuracy of the models, as stated above the only indicators of classification performance are the fitness function values obtained by the models (i.e. CV-BER); thus, we resort to this estimate for selecting members of ensemble methods. In the rest of this section we describe the three strategies we have adopted for the selection of ensemble members. These strategies were defined because we think that under the below conditions one may expect that the models are both highly accurate and diverse.

A. Best-set ensembles

We consider the set of global best solutions obtained every $h-$iterations of PSMS. That is, the set $E_1 = \{\mathbf{p}_g^h, \mathbf{p}_g^{2\times h} \ldots, \mathbf{p}_g^{I_{max}}\}$. Note that models considered under this strategy may be very effective (if the CV-BER criterion is not over-fitted); however, the diversity of methods selected under this technique may be limited because noting prevents the models be similar from each other (e.g. in case the local minima $\mathbf{p}_g^{I_{max}}$ is found at the very first iterations). A total of $\left(\frac{I_{max}}{h}\right)+1$ models are evaluated. We call this configuration the **EPSMS-BS**.

B. Swarm ensemble

This is the most natural way of creating an ensemble in PSMS. Basically it consists of combining the solutions in the swarm at the end of the search process; that is, $E_3 = \{\mathbf{x}_1^{I_{max}}, \ldots, \mathbf{x}_m^{I_{max}}\}$. Under this setting the optimized swarm can provide of highly accurate models as these solutions are the ones with better performance. Although, it is possible that the diversity of these solutions may be small as most solutions will converge to a single model. Nevertheless, by adjusting the PSMS's parameters we can ensure some degree of heterogeneity among the models, see Section III. A total of $m-$models are considered in this way. We call this configuration **EPSMS-SE**.

C. Best-per-iteration ensemble

We consider the collection of models that obtained the best fitness value per each iteration; independently of whether or not they outperformed the global best solution. That is, we consider the set $E_2 = \{\mathbf{x}_{max}^1, \ldots, \mathbf{x}_{max}^{I_{max}}\}$; where $\mathbf{x}_{max}^t$ is the best particle (with respect to the rest of particles at iteration t) at iteration t. Under this formulation models are potentially accurate and diverse, as they have obtained the lowest fitness value in a certain iteration and it is difficult for the same model to obtain the same fitness score. A total of $I_{max}+1$ models are evaluated under this approach. We call this setting **EPSMS-BI**.

We have defined the set of candidate solutions from PSMS that will be considered for building ensembles. Note that no one of the above sets requires of further computation than that of traditional PSMS as all of these solutions are evaluated anyways. In the rest of this section we describe the way the selected models are combined.

D. Fusion strategy

For combining the individual models we consider a simple (unweighted) averaging strategy: when a new pattern $\mathbf{p}^T$ needs to be classified all of the individual models (which have been previously trained using training data) are used to classify the instance. Each individual model k express its

TABLE II

BENCHMARK DATA SETS USED IN OUR EXPERIMENTS.

ID	Data set	Training	Testing	Features
1	Breast cancer	200	77	9
2	Diabetes	468	300	8
3	Flare solar	666	400	9
4	German	700	300	20
5	Heart	170	100	13
6	Image	1300	1010	20
7	Splice	1000	2175	60
8	Thyroid	140	75	5
9	Titanic	150	2051	3

confidence on the class of the pattern $f_k(\mathbf{p}^T) \in [-1, 1]$, then we use average of confidence values as the confidence of the ensemble:

$$g(E_x) = \frac{1}{L}\sum_{k=1}^{L} f_k(\mathbf{p}^T) \tag{3}$$

where L is the number of members in the ensemble, $x \in \{1, 2, 3\}$ indicates the ensemble strategy, $f_k(\mathbf{p}^T)$ is the confidence that the k^{th} classifier has on the class of the pattern $\mathbf{p}^T$. Finally, we assign to test pattern the class corresponding to the sign of $g(E_x)$.

Because the considered classifiers are potentially heterogeneous their outputs are normalized before the fusion so that they lie in a comparable scale. We considered the following normalization for a classifier k

$$f_k(\mathbf{p}^T) = \frac{f_k(\mathbf{p}^T) - min(f_k(.))}{max(f_k(.)) - min(f_k(.))} \tag{4}$$

where $f_k(\mathbf{p}^T)$ is the output of classifier k for input $\mathbf{p}^T$, $min(f_k(.))$ and $max(f_k(.))$ are the minimum and maximum values, respectively, assigned by the k^{th} classifier to an instance in the test set.

IV. EXPERIMENTS AND RESULTS

In this section we report experimental results on both benchmark data and an object recognition data set. The goals of the experiments are evaluating the gain we can have by adopting the ensemble strategy instead of selecting a single model and assessing the diversity of models evaluated by PSMS.

A. Data sets and evaluation methodology

We consider the benchmark data sets described in Table II, which have been used in other studies [22], [23], [5]. All of these data sets are associated with binary classification problems.

Additionally we considered an object recognition data set[2] where the task is to classify regions according to semantic concepts [24]. The data was kindly provided by G. Papadopoulous and was obtained as follows. A set of images were segmented using an automatic technique, then regions were manually labeled with one of 10 concepts (see column 1 in Table III). Visual features were extracted from each region; thus the pairs of visual features and label associated with each region are the instances of the classification problem. In particular: 3 sets of visual features were extracted: wavelet features, SIFT features and MPEG-7 features, resulting in 768 features; in this work we combined these features and applied a feature selection method for keeping the 50 most important features, in this way we reduced the dimensionality of the data set before applying PSMS.

[2]http://mklab.iti.gr/project/scef

TABLE III

CHARACTERISTICS OF THE OBJECT RECOGNITION DATA SET THAT WE CONSIDERED.

Class	Training	Testing	Imbalance ratio
building	280	450	11.77 - 13.64
foliage	506	681	21.27 - 20.63
mountain	203	349	08.53 - 10.57
person	224	219	09.41 - 06.63
road	89	127	03.74 - 03.84
sailing-boat	39	70	01.64 - 02.12
sand	208	273	08.74 - 08.27
sea	325	338	13.66 - 10.24
sky	461	664	19.28 - 20.12
snow	43	129	01.80 - 03.90
Total	2378	3300	(Avg.) 10 - 10

As the data set is associated to a multiclass classification problem we adopted the one-vs-all strategy for building up multiclass classifiers from multiple binary models [25]. Under this approach we create K binary classifiers (with K the number of classes), the k^{th} classifier is able to distinguish examples from class k (positive examples) from the rest $j : j \neq k$ (negative examples). When a new instance needs to be classified the K classifiers are tested and the classifier with the highest confidence decides the class of the instance. It is rather clear that the individual classifiers face a highly imbalanced problem, column 3 in Table III shows the imbalance ratio for each of the classes for the training and test sets. For this data set we use PSMS for selecting classifiers (and ensemble classifiers) for each class; hence we report both the per-class performance and the multiclass accuracy.

We evaluate accuracy by using the area under the ROC curve performance (**AUC**) [26]; we use AUC as leading measure because it is unsensitive to the selection of a classification threshold in the output of classifiers, since we are averaging the outputs of heterogeneous classifiers setting an appropriate decision threshold may be a difficult task. Additionally, for illustrative purposes we report the maximum possible accuracy (**M-ACC**) that can be obtained with the members of the ensemble; that is, the accuracy we would get if we select the correct output for each test instance from the predictions of individual models, provided the correct label is predicted by a member of the ensemble.

We evaluate diversity using one of the widely used measures for assessing non-pairwise diversity. Namely the coincident failure diversity measure (**CFD**):

$$CFD = \begin{cases} \frac{1}{1-p_0}\sum_{r=1}^{L}\frac{L-r}{L-1}p_r & \text{if } p_0 < 1 \\ 0 & \text{if } p_0 = 1 \end{cases} \tag{5}$$

where p_r the probability the r models fail on a randomly

chosen data, see [2], [3] for details. In this work we estimate **CFD** using test data, thus we define p_r as the average, over the test data, of errors made by r models. These measure evaluate how complimentary the ensemble members are, the higher the value of **CFD** the more diverse they are.

For each of the above described data sets we adopted the following methodology. We ran PSMS under a fixed parameter setting storing the solutions described in Sections III-A, III-C and III-B. At the end of search we train the model selected with PSMS (i.e. $\mathbf{p}_g^{I_{max}}$) using the entire training set, the trained model is used to predict the outputs for instances in the test set, then we evaluate the classification performance of the model (we call this model **PSMS-BEST**). Until this stage no extra computation is required than traditional PSMS. Next, the individual models considered for the different ensemble strategies are trained using the entire training data, the individual models are tested on test data and their output are combined as described in Section III-D. Then, the performance and diversity of the ensemble methods is evaluated.

Note that no extra computation is required for selecting ensemble members with PSMS than that required for the execution of straight PSMS. Thus, the complexity of the ensemble selection approach is that of the PSMS technique. As described in [5], the complexity of PSMS depends on the methods considered, the number of patterns and the dimensionality of the data. Thus, for some data sets the application of PSMS can be very expensive. Nevertheless, we can rely on heuristics that can ameliorate the complexity of PSMS. For example, in [5] a subsampling strategy is considered for applying PSMS to data sets with hundreds of thousands of patterns and dozens of thousands of features.

B. Influence of PSMS's parameters

In preliminary experiments we varied the PSMS parameters that can influence the diversity of solutions, namely c_2, W and I_{max}, see Section III (these results are not shown here because of space constraints). From our experiments we found that no significative difference in ensemble accuracy can be obtained by adjusting such parameters, although, as expected, such parameters have an impact on the diversity of models (i.e. on **CFD**). The difference in accuracy between ensembles and the single best solution selected with PSMS was similar to that reported in the next section, thus we postpone our discussion in such topic for the next section.

We found that when c_2 approaches to zero and when we use a a constant $W = 0$ the diversity of models is significantly increased; such increase, however, did not result in a significant improvement in performance. This can be due to the fact that the mechanisms we adopted for increasing diversity affected the performance of individual models. We also found that I_{max} do not have a significant impact in the performance of PSMS ensembles; by running PSMS for $I_{max} = 5$ and $I_{max} = 10$ we obtained the best results, running PSMS for more iterations reduced the diversity of models, whereas increased the individual performance of models. Summarizing, c_2, W and I_{max} have an impact in the diversity and accuracy of ensembles, although, even when the default parameter settings are used (see [5]), the ensemble versions of PSMS (i.e. EPSMS) outperform the best single model.

TABLE IV

AVERAGE AND STANDARD DEVIATION OF AUC OVER 10 TRIALS FOR EACH DATA SET IN TABLE II AND FOR EACH METHOD.

ID	PSMS-BEST	EPSMS-BS	EPSMS-SE	EPSMS-BI
1	$72.03^{+}_{-}2.24$	$73.40^{+}_{-}0.78$	$74.05^{+}_{-}0.91$	$\mathbf{74.35^{+}_{-}0.49}$
2	$82.11^{+}_{-}1.29$	$82.60^{+}_{-}1.52$	$74.07^{+}_{-}13.70$	$\mathbf{83.42^{+}_{-}0.46}$
3	$68.81^{+}_{-}4.31$	$69.38^{+}_{-}4.53$	$70.13^{+}_{-}7.48$	$\mathbf{72.16^{+}_{-}1.42}$
4	$73.92^{+}_{-}1.23$	$73.84^{+}_{-}1.53$	$74.70^{+}_{-}0.72$	$\mathbf{74.77^{+}_{-}0.69}$
5	$85.55^{+}_{-}5.48$	$87.40^{+}_{-}2.01$	$87.07^{+}_{-}0.75$	$\mathbf{88.36^{+}_{-}0.88}$
6	$97.21^{+}_{-}3.15$	$98.85^{+}_{-}1.45$	$95.27^{+}_{-}3.04$	$\mathbf{99.58^{+}_{-}0.33}$
7	$97.26^{+}_{-}0.55$	$98.02^{+}_{-}0.64$	$96.99^{+}_{-}1.21$	$\mathbf{98.84^{+}_{-}0.26}$
8	$96.00^{+}_{-}4.75$	$98.18^{+}_{-}0.94$	$97.29^{+}_{-}1.54$	$\mathbf{99.22^{+}_{-}0.45}$
9	$73.24^{+}_{-}1.16$	$73.50^{+}_{-}0.95$	$\mathbf{75.37^{+}_{-}1.05}$	$74.40^{+}_{-}0.91$
Avg.	$82.90^{+}_{-}2.68$	$83.91^{+}_{-}1.59$	$82.77^{+}_{-}3.38$	$\mathbf{85.01^{+}_{-}0.65}$

C. Diversity and accuracy of Ensemble PSMS

We now analyze the performance and diversity of ensembles by using the following parameters setting: $c_2 = 0.1$, $W = 0$ and $I_{max} = 10$; the rest of parameters were fixed as described in [5]. For this experiment we considered the benchmark data sets, 10 trials for each data set were performed. Table IV shows the average and standard deviation of the **AUC** obtained with each configuration of PSMS.

From Table IV we can see that for most data sets the ensemble versions of PSMS (columns 3-5) outperformed the best single model (column 2). The best method was **EPSMS-BI** which outperformed the other techniques in 8 out of the 9 data sets considered; for the titanic data set **EPSMS-SE** obtained better performance, although still **EPSMS-BI** outperformed the other two methods in this data set.

We performed a Wilcoxon signed-rank test for the comparison of the different methods [27]. In the following we will refer to this statistical test with 95% of confidence when mentioning statistical significance. The per-data set differences between **PSMS-BEST** and **EPSMS-BI** are statistically significant for all data sets. Whereas the differences between **PSMS-BEST** and **EPSMS-SE** are significative for 5 out of the 9 data sets and the differences between **EPSMS-BS** and **PSMS-BEST** are significant for 7 out of the 9 data sets. These results show that the ensemble versions of PSMS can improve the performance of the single best model. In particular building ensembles with the best model per iteration (i.e. **EPSMS-BI**) results in better performance.

From Table IV we can observe an interesting fact: the standard deviation in the **AUC** performance obtained with **EPSMS-BI** is smaller than that obtained with any other technique. This result suggest that the ensembles built with **EPSMS-BI** across the different trials of each data set obtained similar performances, whereas for the rest of the methods the standard deviation is very high, which means that such models provide highly unreliable predictions. Therefore, **EPSMS-BI** can also provide more stable predictions than the **PSMS-BEST** and than the other ensemble methods.

TABLE V

AVERAGE AND STANDARD DEVIATION OF CFD OVER 10 TRIALS FOR EACH DATA SET IN TABLE II AND FOR EACH VARIANT OF EPSMS.

ID	EPSMS-BS	EPSMS-SE	EPSMS-BI
1	$0.2055^{+}_{-}0.1498$	$\mathbf{0.5422^{+}_{-}0.0550}$	$0.5017^{+}_{-}0.1149$
2	$0.3547^{+}_{-}0.1711$	$\mathbf{0.6241^{+}_{-}0.0169}$	$0.5081^{+}_{-}0.0728$
3	$0.1295^{+}_{-}0.1704$	$\mathbf{0.4208^{+}_{-}0.1357}$	$0.4012^{+}_{-}0.1071$
4	$0.3019^{+}_{-}0.1732$	$\mathbf{0.5159^{+}_{-}0.0596}$	$0.4296^{+}_{-}0.0490$
5	$0.2733^{+}_{-}0.1714$	$\mathbf{0.5993^{+}_{-}0.0925}$	$0.5647^{+}_{-}0.0655$
6	$0.7801^{+}_{-}0.0818$	$0.7555^{+}_{-}0.0524$	$\mathbf{0.8427^{+}_{-}0.0408}$
7	$0.5427^{+}_{-}0.3230$	$0.7807^{+}_{-}0.0585$	$\mathbf{0.8050^{+}_{-}0.0294}$
8	$0.6933^{+}_{-}0.1558$	$0.8173^{+}_{-}0.0626$	$\mathbf{0.8514^{+}_{-}0.0403}$
9	$0.7473^{+}_{-}0.0089$	$0.7473^{+}_{-}0.0089$	$0.7473^{+}_{-}0.0089$
Avg.	$0.4476^{+}_{-}0.1562$	$\mathbf{0.6448^{+}_{-}0.0603}$	$0.6280^{+}_{-}0.0588$

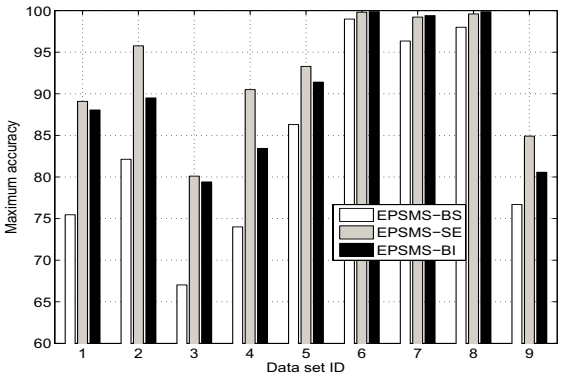

Fig. 1. Average of maximum accuracy (**M-ACC**) that can be obtained with each EPSMS strategy.

Table V show the average and standard deviation of **CFD** for the EPSMS methods. We can see that the **EPSMS-BS** method obtain the less diverse models, which is due to the fact that several of the individual models considered in this technique can be repeated, thus decreasing diversity; note that diversity across trials is highly unstable for this method too, as evidenced by the standard deviation results. On the other hand, **EPSMS-BI** and **EPSMS-SE** showed similar diversity and similar stability across different trials, which makes them particularly helpful for building ensembles. Since diversity is comparable for the latter techniques, the better accuracy of **EPSMS-BI** (see Table IV) is due to the fact that better individual models are obtained with such strategy.

Figure 1 shows the (average of) maximum accuracy that can be obtained by combining the outputs of individual models under the different strategies. We can see that with **EPSMS-SE** we could obtain the maximum accuracy among EPSMS techniques. Then, why **EPSMS-BI** outperforms **EPSMS-SE** in ensemble performance? This can be due to the fact that **EPSMS-SE** can contain a few very effective models that make the **M-ACC** to be very high, but most of the individual models present rather limited performance, possibly due to over-fitting of the optimization criterion; therefore, when their outputs are merged, their combined performance is not satisfactory. On the other hand, most models that compose **EPSMS-BI** obtain acceptable performance and thus its combination its beneficial.

TABLE VI

AVERAGE AND STANDARD DEVIATION OF AUC ACROSS THE CATEGORIES OF THE OBJECT RECOGNITION DATA SET AND MULTICLASS ACCURACY OBTAINED WITH EACH CONFIGURATION OF EPSMS (ACC).

Measure	PSMS-BEST	EPSMS-BS	EPSMS-SE	EPSMS-BI
Avg. AUC	$0.9153^{+}_{-}0.068$	$0.9327^{+}_{-}0.056$	$0.9279^{+}_{-}0.074$	$0.9405^{+}_{-}0.053$
ACC	69.58 %	76.59 %	79.13 %	**81.49** %

Summarizing, in this section we have shown empirical evidence about the benefits of building ensembles with PSMS's partial solutions. The three strategies we proposed resulted very effective in terms of accuracy and diversity. However, **EPSMS-BI** obtained the best performance and high diversity. Furthermore, predictions obtained with **EPSMS-BI** resulted more stables than that of any other method we tried. Therefore, we recommend the use of **EPSMS-BI** instead of raw PSMS (i.e. **PSMS-BEST**) or the other two strategies.

D. Object recognition data set

In this section we evaluate the performance of EPSMS techniques into the object recognition data set. For this experiment we used the same parameters as above. Table VI shows the results we obtained in this experiment. The average accuracy over the classes (Avg. AUC) is slightly superior for the EPSMS methods; again **EPSMS-BI** obtained the best result. This time the standard deviation obtained by the different methods is comparable among all techniques; however, note that this result only reveals that the accuracy across the different classes was similar.

The best multiclass accuracy was obtained with **EPSMS-BI** as well. To the best of our knowledge this the best performance reported so far for this data set: Papadopoulos et al. reported a classification performance of 62.45% [24], whereas Escalante et al. reported 81.41% [28]. The difference with the latter work is insignificant, however, one should note that in that previous work the same classification model has been used for every class in the collection; thus when adopting the one-vs-all approach the outputs of such classifiers are directly comparable. In our approach, the models for each class are different, thus their outputs lie in different scales and the application of the one-vs-all approach result in a loss of performance. Therefore, we believe that by applying more elaborated strategies than one-vs-all the multiclass performance will increase significantly.

Figure 2 shows the per-label gain obtained by each of the three EPSMS techniques with respect to **PSMS-BEST**. We can see that the **EPSMS-BS** and **EPSMS-BI** techniques outperformed the single best solution for all classes, whereas **EPSMS-ES** obtained lowest performance in two classes. Again, giving evidence of the instability of **EPSMS-ES**.

Summarizing, our results on the object recognition data set show that EPSMS can also be helpful for obtaining classifiers in multiclass classification problems. Again, **EPSMS-BI** obtained the best performance. The results obtained with our proposals in this data set are superior to any other published work that have used the same collection. Even

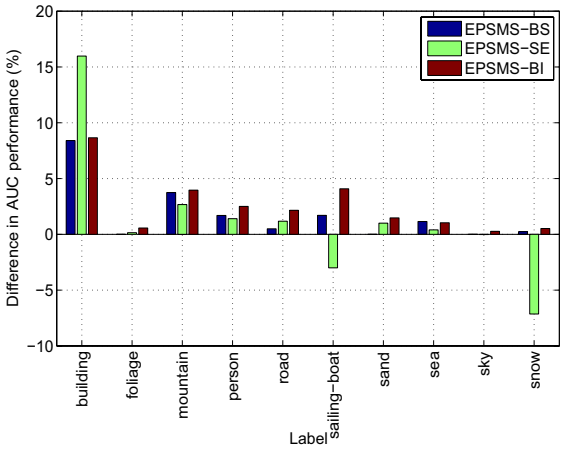

Fig. 2. Per-label performance gain of the EPSMS techniques over the **PSMS-BEST** classifier.

when the outputs of the models for the different classes are not comparable at all.

V. Conclusions

We have described three strategies for building ensemble classifiers from PSMS's partial solutions. PSMS is a generic technique for the selection of highly effective classifiers. In this work we improved the performance of models selected with PSMS by adopting an ensemble approach where the outputs of some models that are evaluated by PSMS are combined. The selection of the ensemble members do not require of further computation than that of traditional PSMS.

We reported experimental results in both benchmark data and an object recognition data set. We found that ensemble versions of PSMS outperform the single best solution as selected with traditional PSMS. In particular, the strategy **EPSMS-BI** obtained the best results. The diversity of ensembles obtained from PSMS solutions is acceptable and hence motivates further research on the use of PSMS for building ensembles. Our results in the object recognition data set show that EPSMS can also be helpful for multiclass classification problems. Future work includes the study of better strategies to identify ensemble members from PSMS solutions and the comparison of **EPSMS-BI** to other strategies for building heterogeneous ensembles (e.g. [16], [15]). Finally, we would like to point out that PSMS is distributed with the CLOP toolbox, in the latest version of PSMS[3] you find the code of the methods we introduced in this paper.

Acknowledgments: We thank the reviewers for their comments that helped us to improve this paper. This work was partially supported by *CONACyT* under project grant No. 61335 and scholarship No. 205834.

References

[1] T. Dietterich, "Ensemble methods in machine learning," in *Proceedings of the First workshop on Multiple Classifier Systems*, vol. 1857 of *LNCS*, pp. 1–15, Springer, 2000.

[2] W. Wang, "Some fundamental issues in ensemble methods," in *IJCNN07*, (Orlando, FL, USA), pp. 2244–2251, IEEE, July 2007.

[3] S. Bian and W. Wang, "On diversity and accuracy of homogeneous and heterogeneous ensembles," *International Journal of Hybrid Intelligent Systems*, vol. 4, pp. 103–128, 2007.

[4] I. Guyon, A. Saffari, G. Dror, and G. Cawley, "Analysis of the ijcnn 2007 competition agnostic learning vs. prior knowledge," *Neural Networks*, vol. 21, no. 2–3, pp. 544–550, 2008.

[5] H. J. Escalante, M. Montes, and E. Sucar, "Particle swarm model selection," *Journal of Machine Learning Research*, vol. 10, pp. 405–440, February 2009.

[6] H. J. Escalante, M. Montes, and E. Sucar, "Psms for neural networks on the ijcnn 2007 agnostic vs prior knowledge challenge," in *IJCNN07*, (Orlando, FL, USA), pp. 1191–1197, IEEE, 2007.

[7] H. J. Escalante, M. Montes, and L. Villaseñor, "Particle swarm model selection for authorship verification," in *Proceedings of the 14th Iberoamerican Congress on Pattern Recognition*, vol. 5856 of *LNCS*, (Guadalajara, Mexico), pp. pp. 563–570, Springer, 2009.

[8] L. Kuncheva and C. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," *Machine Learning*, vol. 51, pp. 181–207, 2003.

[9] Y. Freund and R. Schapire, "Experiments with a new boosting algorithm," in *Proceedings of the 13th Conference on Machine Learning*, pp. 148–156, 1996.

[10] L. Breiman, "Bagging prediction," *Machine Learning*, vol. 14, pp. 123–140, 1996.

[11] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[12] L. Kuncheva, "Cluster and selection method for classifier combination," in *Proceedings of the 4th International Conference on Knowledge-based Intelligent Engineering Systems and Allied Technologies*, (Brighton, UK), pp. 185–188, 2000.

[13] G. Giacinto and F. Roli, "Methods for dynamic classifier selection," in *Proceedings of the 10th International Conference on Image Analysis and Processing*, (Venice, Italy), pp. 659–664, 1999.

[14] M. C. J. D. Wichard and M. Ogorzalek, "Building ensembles with heterogeneous models," in *Proceedings of the 7th Course on the International School on Neural Nets IIASS*, (Salerno, Italy), 2002.

[15] C. Park and S. Cho, "Evolutionary computation for optimal ensemble classifier in lymphoma cancer classification," in *Foundations of Intelligent Systems: ISMIS* (N. Z. et al., ed.), vol. 2871 of *LNAI*, (Maebashi City), pp. 521–530, Springer Berlin Heidelberg, 2003.

[16] M. Macas, D. R. B. Gabrys, and, and L. Lhotska, "Particle swarm optimization of multiple classifier systems," in *IWANN*, vol. 4507 of *LNCS*, pp. 333–340, Springer, 2007.

[17] L. Yang and Z. Qin, "Combining classifiers with particle swarms," in *ICNC* (L. Wang et al., ed.), vol. 3611 of *LNCS*, pp. 756–763, Springer Berlin Heidelberg, 2005.

[18] G. Giacinto and F. Roli, "An approach to the automatic design of multiple classifier systems," *Pattern Recognition Letters*, vol. 22, no. 1, pp. 25–33, 2001.

[19] D. Ruta and B. Gabrys, "Classifier selection for majority voting," *Information Fusion*, vol. 6, pp. 63–81, March 2005.

[20] D. Gorissen, T. Dhaene, and F. de Turck, "Evolutionary model type selection for global surrogate modeling," *Journal of Machine Learning Research*, vol. 10, pp. 2039–2078, 2009.

[21] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*. Wiley, 2006.

[22] G. Rätsch, T. Onoda, and K.-R. Müller, "Soft margins for adaboost," *Machine Learning*, vol. 42, no. 3, pp. 287–320, 2001.

[23] G. C. Cawley and N. L. C. Talbot, "Preventing over-fitting during model selection via bayesian regularisation of the hyper-parameters," *Journal of Machine Learning Research*, vol. 8, pp. 841–861, 2007.

[24] G. Papadopoulos, C. Saathoff, M. Grzegorzek, V. Mezaris, I. Kompatsiaris, S. Staab, and M. Strintzis., "Comparative evaluation of spatial context techniques for semantic image analysis," in *Proceedings of the 10th International Workshop on Image Analysis for Multimedia Interactive Services*, (London, UK), pp. 161–164, IEEE, 2009.

[25] R. Rifkin and A. Klautau, "In defense of one-vs-all classification," *Journal of Machine Learning Research*, vol. 5, pp. 101–141, 2004.

[26] A. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.

[27] J. Demsar, "Statistical comparisons of classfifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[28] H. J. Escalante, M. Montes, and L. E. Sucar, "An energy-based model for image annotation and retrieval," vol. Submitted, 2010.

[3] http://ccc.inaoep.mx/∼hugojair/code/psms/