

SEG-SSC: A Framework based on Synthetic Examples Generation for Self-Labeled Semi-Supervised Classification

Isaac Triguero, Salvador García, and Francisco Herrera

Abstract—Self-labeled techniques are semi-supervised classification methods that address the shortage of labeled examples via a self-learning process based on supervised models. They progressively classify unlabeled data and use them to modify the hypothesis learned from labeled samples. Most relevant proposals are currently inspired by boosting schemes to iteratively enlarge the labeled set. Despite their effectiveness, these methods are constrained by the number of labeled examples and their distribution, which in many cases is sparse and scattered. The aim of this work is to design a framework, named SEG-SSC, to improve the classification performance of any given self-labeled method by using synthetic labeled data. These are generated via an oversampling technique and a positioning adjustment model that use both labeled and unlabeled examples as reference. Next, these examples are incorporated in the main stages of the self-labeling process. The principal aspects of the proposed framework are: (a) introducing diversity to the multiple classifiers used by using more (new) labeled data, (b) fulfilling labeled data distribution with the aid of unlabeled data, and (c) being applicable to any kind of self-labeled method. In our empirical studies, we have applied this scheme to four recent self-labeled methods, testing their capabilities with a large number of data sets. We show that this framework significantly improves the classification capabilities of self-labeled techniques.

Index Terms—Self-Labeled methods, co-training, synthetic examples, semi-supervised classification.

I. INTRODUCTION

HAVING a multitude of unlabeled data and few labeled ones occurs quite often in many practical applications such as medical diagnosis, spam filtering, bioinformatics, etc. In this scenario, learning appropriate hypotheses with traditional supervised classification methods [1] is not straightforward because they only can exploit labeled data. Nevertheless, Semi-Supervised Classification (SSC) [2], [3], [4] approaches also utilize unlabeled data to improve the predictive performance, modifying the learned hypothesis obtained from labeled examples alone.

With SSC we may pursue two different objectives: transductive and inductive classification [5]. The former is devoted to predicting the correct labels of a set of unlabeled examples that is also used during the training phase. The latter refers

to the problem of predicting unseen data by learning from labeled and unlabeled data as training examples. In this work, we will analyze both settings.

Existing SSC algorithms are usually classified depending on the conjectures they make about the relation of labeled and unlabeled data distributions. Broadly speaking, they are based on the manifold and/or cluster assumption. The manifold assumption is satisfied if data lie approximately on a manifold of lower dimensionality than the input space [6]. The cluster assumption states that similar examples should have the same label. Graph-based models [7] are the most common approaches to implementing the manifold assumption [8]. As regards examples of models based on the cluster assumption, we can find generative models [9] or semi-supervised support vector machines [10]. Recent studies have addressed multiple assumptions in one model [11], [5], [12].

Self-labeled techniques are SSC methods that do not make any specific suppositions about the input data [13]. These models use unlabeled data within a supervised framework via a self-training process. First attempts correspond to the self-training algorithm [14] that iteratively enlarges the labeled training set by adding the most confident predictions of the supervised classifier used. The standard co-training [15] methodology splits the feature space into two different conditionally independent views. Then, it trains one classifier in each specific view, and the classifiers teach each other the most confidently predicted examples. Advanced approaches do not require explicit feature splits or the iterative mutual-teaching procedure imposed by co-training, as they are commonly based on disagreement-based classifiers [16], [17], [18]. These models have been successfully applied to many real applications such as image classification [19], shadow detection [20], computer-aided diagnosis [21], etc.

Self-labeled techniques are limited by the number of labeled points and their distribution to identifying reliable unlabeled examples. This problem is more pronounced when the labeled ratio is greatly reduced and labeled examples do not minimally represent the domain. Moreover, most of the advanced models use some diversity mechanisms, such as bootstrapping [22], to provide differences between the hypotheses learned with the multiple classifiers. However, these mechanisms may provide a similar performance to classical self-training or co-training approaches if the number of labeled data is insufficient to achieve different learned hypotheses.

The aim of this work is to alleviate these weaknesses by using new synthetic labeled examples to introduce diversity

This work was supported by the Research Projects TIN2011-28488, P10-TIC-6858 and P11-TIC-7765.

I. Triguero and F. Herrera are with the Department of Computer Science and Artificial Intelligence of the University of Granada, CITIC-UGR, Granada, Spain, 18071. E-mails: {triguero, herrera}@decsai.ugr.es

Salvador García is with the Department of Computer Science of the University of Jaén Jaén, Spain, 23071. E-mail: sglopez@ujaen.es

to multiple classifier approaches and fulfill the labeled data distribution. A complete motivation for the use of synthetic labeled examples is discussed in Section III-A.

We propose a framework applicable to any self-labeled method that incorporates synthetic examples in the self-learning process. We will denote this framework ‘‘Synthetic Examples Generation for Self-labeled Semi-supervised Classification’’ (SEG-SSC). It is composed of two main parts: generation and incorporation.

- The generation process consists of an oversampling technique and a later adjustment of the positioning of the examples. It is initially inspired by the SMOTE algorithm [23] to generate new synthetic examples, for all the classes, based on both the small labeled set and the unlabeled data. Then, this process is refined using a positioning adjustment of prototypes model [24] based on a differential evolution algorithm [25].
- New labeled points are then included in two of the main steps of a self-labeling method: the initialization phase and the update of the labeled training set, so that it introduces new examples in a progressive manner during the self-labeling process.

An extensive experimental analysis is carried out to check the performance of the proposed framework. We apply the SEG-SSC scheme to four recent self-labeled techniques that have different characteristics, comparing the performance obtained with the original proposals. We conduct experiments over 55 standard classification data sets extracted from the KEEL and UCI repositories [26], [27] and 11 high dimensional data sets from the book by Chapelle et al. [2]. The results will be contrasted with nonparametric statistical tests [28], [29].

The remainder of this paper is organized as follows. Section II defines the SSC problem and sums up the classical and current self-labeled approaches. Then, Section III presents the proposed framework, explaining its motivation and the details of its implementation. Section IV describes the experimental setup and discusses the results obtained. Finally, Section V summarizes the conclusions drawn in this work.

II. SELF-LABELED SEMI-SUPERVISED CLASSIFICATION

This section provides the definition of the SSC problem (Section II-A) and briefly describes the most relevant self-labeled approaches proposed in the literature (Section II-B).

A. Semi-supervised classification

A formal description of the SSC problem is as follows: Let \mathbf{x}_p be an example where $\mathbf{x}_p = (\mathbf{x}_{p1}, \mathbf{x}_{p2}, \dots, \mathbf{x}_{pD}, \omega)$, with \mathbf{x}_p belonging to a class ω and a D -dimensional space in which \mathbf{x}_{pi} is the value of the i -th feature of the p -th sample. Then, let us assume that there is a labeled set L which consists of n instances \mathbf{x}_p with ω known and an unlabeled set U which consists of m instances \mathbf{x}_q with ω unknown, let $m > n$. The $L \cup U$ set forms the training set TR . Moreover, there is a test set TS composed of t unseen instances \mathbf{x}_r with ω unknown, which has not been used at the training stage.

The aim of SSC is to obtain a robust learned hypothesis using TR instead of L alone. It can be applied in two slightly

different settings. On the one hand, transductive learning is devoted to classify all the m instances \mathbf{x}_q of U with their correct class. The class assignment should represent the distribution of the classes efficiently, based on the input distribution of L and U . On the other hand, the inductive learning phase consists of correctly classifying the instances of TS based on the previously learned hypothesis.

B. Self-labeled techniques: previous work

Self-labeled techniques form an important family of methods in SSC [3]. They are not intrinsically geared to learning in the presence of both labeled and unlabeled data, but they use unlabeled points within a supervised learning paradigm. These techniques aim to obtain one (or several) enlarged labeled set/s, based on the most reliable predictions. Thus, these models do not make any specific assumptions about the input data, but the models accept that their own predictions tend to be correct. Some authors state that self-labeling is likely to be the case when the classes form well-separated clusters [3] (cluster assumption).

The major benefits of this family of methods are: simplicity and being a wrapper methodology. The former is related to the facility of implementation and applicability. The latter means that any kind of classifier can be used regardless of its complexity, which is very important depending on the problem tackled. As caveats, the addition of wrongly labeled examples during the self-labeling process can lead to an even worse performance. Several mechanisms have been proposed to reduce this problem [30].

A preeminent work with this philosophy is the self-training paradigm designed by Yarowsky [14]. In self-training, a supervised classifier is initially trained with the L set. Then it is retrained with its own most confident predictions, enlarging its labeled training set. Thus, it is defined as a wrapper method for SSC. This idea was later extended by Blum and Mitchell [15] with the method known as co-training. This consists of two classifiers that are trained on two sufficient and redundant sets of attributes. This requirement implies that each subset of features should be able to perfectly define the frontiers between classes. Then, the method follows a mutual teaching procedure that works as follows: each classifier labels the most confidently predicted examples from its point of view and they are added to the L set of the other classifier. It is also known that usefulness is constrained by the imposed requirement [31], which is not satisfied in many real applications. Nevertheless, this method has become an example for recent models thanks to the idea of using the agreement (or disagreement) of multiple classifiers and the mutual teaching approach. A good study of when co-training works can be found in [32].

Due to the success of co-training and its relatively limited application, many works have proposed the improvement of standard co-training by eliminating the established conditions. In [33], the authors proposed a multi-learning approach, so that two different supervised learning algorithms were used without splitting the feature space. They showed that this mechanism divides the instance space into a set of equivalence classes. Later, the same authors proposed a faster and more precise

alternative, named Democratic co-learning (Democratic-Co) [34], which is also based on multi-learning. As an alternative, which requires neither sufficient and redundant views nor several supervised learning algorithms, Zhou and Li [35] presented the Tri-Training algorithm, which attempts to determine the most reliable unlabeled data as the agreement of three classifiers (same learning algorithm). Then, they proposed the Co-Forest algorithm [21] as a similar approach that uses Random Forest [36]. A further similar approach is Co-Bagging [37], [38] where confidence is estimated from the local accuracy of committee members. Other recent self-labeled approaches are [39], [40], [41], [42], [43].

In summary, all of these recent schemes work on the hypothesis that several weak classifiers, learned with a small number of instances, can produce better generalizations than only one weak classifier. These methods are also known as disagreement-based models that are motivated, in part, by the empirical success of ensemble learning. The term disagreement-based was recently coined by Zhou and Li [17].

III. SYNTHETIC EXAMPLES GENERATION FOR SELF-LABELED METHODS.

In this section we present the SEG-SSC framework. Firstly, Section III-A enumerates the arguments that justify our proposal. Secondly, Section III-B explains how to generate useful synthetic examples in a semi-supervised scenario. Finally, Section III-C describes the SEG-SSC framework, emphasizing when synthetic data should be used.

A. Motivation: Why add synthetic examples?

The most important weakness of self-labeling models can occur when erroneous labeled examples are added to the labeled training set. This will incorrectly modify the learned model, which may lead to the addition of wrong examples in successive iterations. Why does this situation occur?

- There may be outliers in the original unlabeled set. This problem can be avoided if they are detected and not included in the labeled training set. For this problem, there are several solutions in the literature such as edition schemes [30], [44], [45] or some other mechanisms [33]. Recent models, such as Tri-Training [35] or Co-Forest [21], establish some criteria to compensate for the negative influence of noise by augmenting the labeled training set with sufficient new labeled data.
- Independently of the number of unlabeled examples, they can be limited by the distribution of labeled input data. If the available labeled instances do not represent a reliable domain of the problem, it may complicate the estimation of confidence predictions because the supervised classifiers used do not have enough information to establish coherent hypotheses. Furthermore, it is even more difficult if these labeled points are very close to the decision boundaries. Figure 1 shows an example with the appendicitis problem [27]. This picture presents a two-dimensional projection (obtained with PCA [46]) of the problem and a partition with 10 % of labeled examples. As we can observe, not only is the problem not well

represented by labeled points, it also shows that some of the nearest unlabeled points to the two labeled examples of class 1 (blue circles) belong to class 0 (red crosses). This fact can affect confidence of a self-labeled method estimated with the base classifier.

- A greatly reduced labeled ratio may produce a lack of diversity among self-labeling methods with more than one classifier. As we have established above, multiple classifier approaches work as a combination of several weak classifiers. However, if there are only a few labeled data it is very difficult to obtain different hypotheses, and therefore, the classifiers are identical. For example, the Tri-Training algorithm is based on a bootstrapping approach [22]. This re-sampling technique creates new labeled sets for each classifier by modifying the original L . In general, this operation yields different labeled sets to the original, but it is not significant in the case of small labeled data sets and the existence of outliers in the sample. As a consequence, it could lead to biased examples which will not accurately represent the domain of the problem. Although multi-learning approaches attempt to achieve diversity by using different kinds of learning models, a reduced number of instances usually damages their performance because the models are too weak.

The first limitation has already been addressed in the literature with different mechanisms [47]. However, the last two issues are currently open problems.

In order to ease both situations, mainly induced by the shortage of labeled points, we introduce new labeled data into the self-labeling process. To do this, we rely on the success of oversampling approaches in imbalanced domains [48], [49], [50], [51], but with the difference that we deal with all the classes of the problem.

Nevertheless, the use of synthetic data for self-labeling methods is not straightforward and must be carefully performed. The aim of using an oversampling method is to effectively reinforce the decision regions between classes. To do so, we will be aided by the distribution of unlabeled data in conjunction with the labeled ones, because if we focus only on labeled examples, it may lead to generate noisy instances when the second issue explained above happens. The effectiveness of this idea will be empirically checked in Section IV.

B. Generation of synthetic examples

To generate new labeled data in an SSC context we perform certain operations on the available data, so that we use both labeled and unlabeled sets. Algorithm 1 outlines the pseudo-code of the oversampling technique proposed. This method is initially based on the SMOTE algorithm proposed in [23] which was designed for imbalanced domains [52] and is limited to oversampling the minority class. In our proposal, we use the underlying idea of SMOTE as an initialization procedure, to generate new examples of all the classes. Furthermore, the resulting synthetic set of prototypes is then readjusted with a positioning adjustment of prototypes scheme [24]. Therefore, this mechanism is divided into two phases: initialization and adjustment of prototypes.

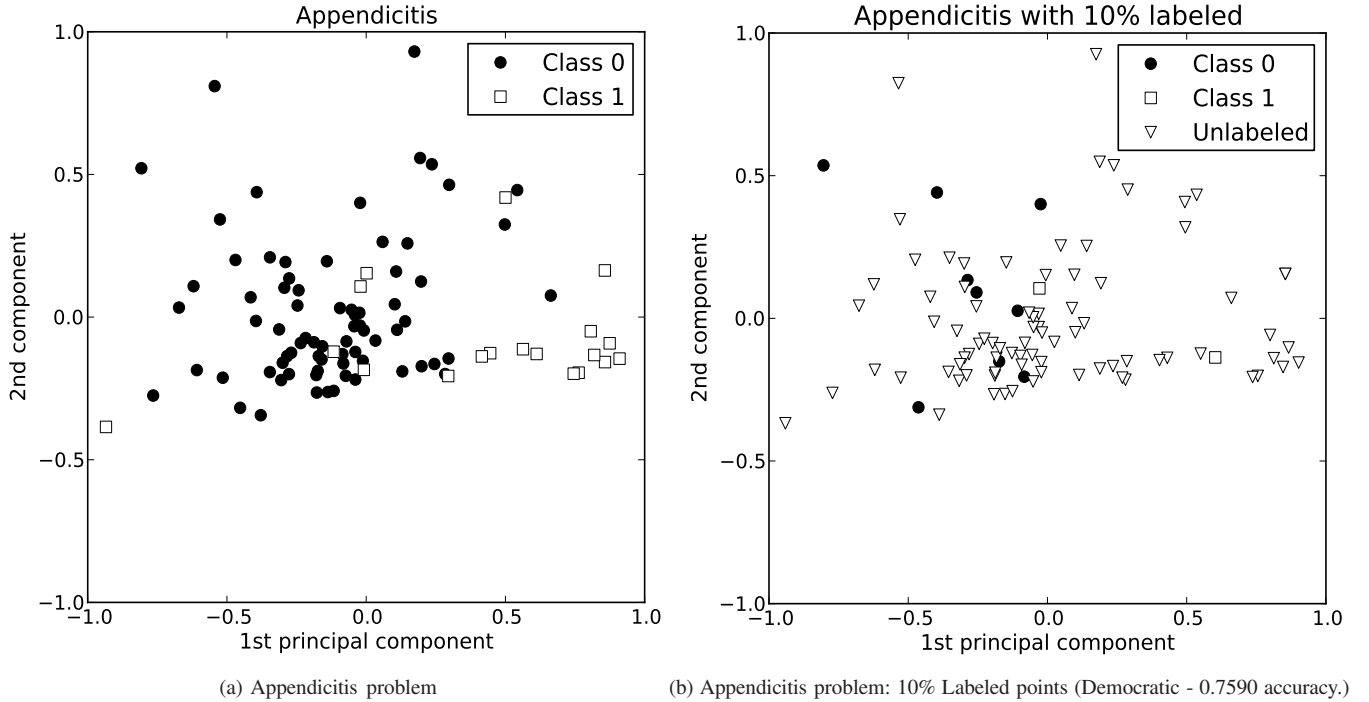


Fig. 1. Two-dimensional projections of Appendicitis. Red circles, class 0. Blue squares, class 1. White triangles, unlabeled.

```

1: Input: Labeled set  $L$ , Unlabeled set  $U$ , Oversampling factor  $f$ ,
   Number of Neighbors  $k$ .
2: Output:  $OverSampled$  set.
3:  $OverSampled = \emptyset$ 
4:  $TR = L \cup U$ 
5:  $ratio = \frac{f \cdot \#TR}{\#L}$ 
6: Randomize  $TR$ 
7: for  $i = 1$  to  $NumberOfClasses$  do
8:    $PerClass[i] = \text{getFromClass}(L, i)$ 
9:   for  $j = 1$  to  $\#PerClass[i]$  do
10:     $Generated = 0$ 
11:    repeat
12:      $neighbors[1..k] = \text{Compute } k \text{ nearest neighbors}$ 
     for  $PerClass[i][j]$  in  $TR$ 
13:      $nn = \text{Random number between } 1 \text{ and } k$ 
14:      $Sample = PerClass[j]$ 
15:      $Nearest = TR[neighbors[nn]]$ 
16:     for  $m = 1$  to  $NumberOfAttributes$  do
17:        $dif = Nearest[m] - Sample[m]$ 
18:        $gap = \text{Random number between } 0 \text{ and } 1$ .
19:        $Synthetic[m] = Sample[m] + gap * dif$ 
20:     end for
21:      $OverSampled = OverSampled \cup Synthetic$ 
22:      $Generated ++$ 
23:    until  $Generated < ratio$ 
24:   end for
25: end for
26:  $OverSampled = DE\_adjustment(OverSampled, L)$ 
27: return  $OverSampled$ 

```

Algorithm 1: Generation of synthetic examples

Initialization: We start from the L and U sets as well as an user-defined oversampling factor f and a number k of nearest neighbors. We will generate a set of synthetic prototypes $OverSampled$ that is initialized as empty (Instruction 3).

The ratio of synthetic examples to be generated is computed according to f and the proportion of labeled examples in the

training set TR (See instructions 4 and 5). Furthermore, to prevent the influence of the order of labeled and unlabeled instances when computing distances, the TR set is randomized (Instruction 6).

Next, the algorithm enters a loop (Instructions 7-25) to proportionally oversample each class, using its own labeled samples as the base. Thus, we extract from L a set of examples $PerClass$ that belong to the current class (Instruction 8). Each one will serve as the base prototype and will be oversampled as many times as the previous computed ratio indicates (Instructions 11-23).

New synthetic examples are located along the line segments joining any of the k nearest neighbors (randomly chosen). To face the SSC scenario, the nearest neighbors are not only being looked for in the L set, but are searched for in the TR set (Instruction 12). In this way, we try to avoid the negative effects of the second weakness of self-labeled techniques explained before. Following the idea of SMOTE, synthetic examples are initially generated as the difference between an existing sample and one of its nearest neighbors (Instruction 17). Then, this difference is scaled by a random number in the range $[0,1]$, and is added to the base example (Instruction 18 and 19). It is noteworthy that the class value of the generated example is the same as the considered base sample. The generated prototypes are iteratively stored in $OverSampled$ until the stopping condition is satisfied.

Adjustment of prototypes: Can we use this process to improve the distribution of labeled input data? The answer depends on the specific problem and partition used. Although the generation algorithm provides more labeled examples that may be very useful in many domains, they are not totally confident. It may suffer from the same problem as the self-

labeling approaches and their confidence predictions. It is well-known that SMOTE can generate noisy data [53] which are usually eliminated with edition schemes. Because we are not interested in removing synthetic data, we will apply an evolutionary adjustment process to the *OverSampled* set (Instruction 26) based on the differential evolution algorithm used in [54].

Differential evolution [25] follows the general procedure of an evolutionary algorithm [55]. It starts with a set of candidate solutions, the so-called individuals, which evolve during a determined number of generations through different operators: mutation, crossover and selection; aiming to minimize/maximize a fitness function. For our purposes, this algorithm is adapted in the following way:

- Each individual encodes a single prototype. The process consists of the optimization of the location of all the individuals of the population.
- Mutation and crossover operators guide the optimization of the positioning of the prototypes. These operators only produce modifications to the attributes of the prototypes of the *OverSampled* set, keeping the class value unchangeable throughout the evolutionary cycle. We will focus on the *DE/CurrentToRand/1* strategy to generate new prototypes [56].
- Then, we obtain a new set of synthetic prototypes that should be evaluated to decide whether it is better or not than the current set. To make this decision, we use the most reliable data we have, that is, the labeled data L . The generated data should be able to correctly classify L . To check this, the nearest neighbor rule is used as the base classifier to obtain the corresponding fitness value. We try to maximize this value.

The stopping criteria is achieved when the generated data perfectly classify L , or a given number of iterations have been performed. More details in Section III.B of reference [54].

It is worth mentioning that this optimization process is only applied to cases in which the former oversampling approach generates synthetic data that is not able to classify L . We thereby endow our model with greater robustness. Figure 2 shows an example of a resulting set of over-sampled prototypes in the appendicitis problem. We can observe that in comparison with Figure 1, the available labeled data points better represent the domain of the problem.

C. Self-labeling with synthetic data

In this subsection, we describe the SEG-SSC framework in depth. With the generation method presented, we obtain new labeled data that can be directly used to improve the generalization capabilities of self-labeled approaches. Nevertheless, the aim of this framework is to be as flexible as possible, so that it can be applied to different self-labeled algorithms. Although each method proceeds in a different way, they either share some operations or are very similar. Therefore, we explain how to incorporate synthetic examples in the self-learning process in order to address the limitations on the distribution of labeled data and the lack of diversity in multiple classifier methods.

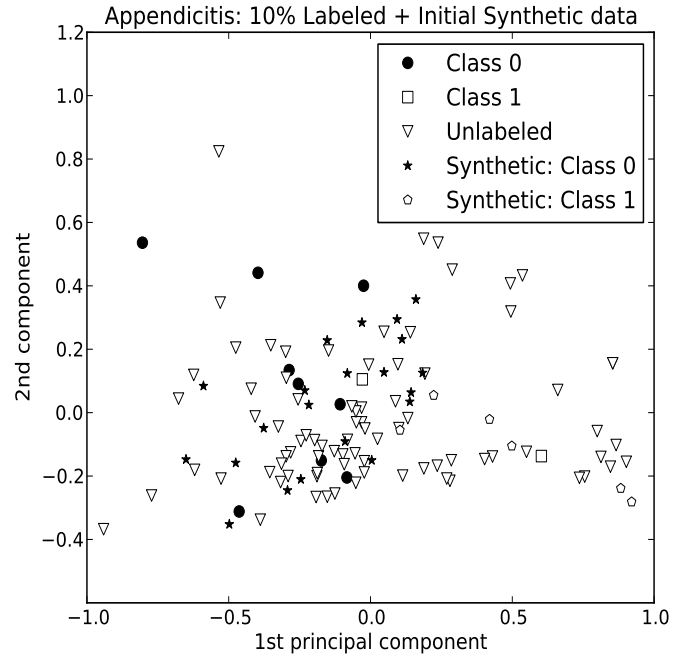


Fig. 2. Example of data generation in the Appendicitis problem. Two-dimensional projections of Appendicitis. Red circles, class 0. Blue squares, class 1. White triangles, unlabeled. Red stars, synthetic class 0. Blue pentagons, synthetic class 1. (SEG-SSC+Democratic - 0.8072 accuracy.)

In general, self-labeled methods use a set of N classifiers C_i , where $i \in [1, N]$, to predict the class of unlabeled instances. Each C_i has an associated labeled set L_i that is iteratively enlarged. In what follows, we describe the three main operations that support our proposal. For clarity, Figure 3 depicts a flowchart of the proposed scheme, outlining its more general operations and way of working.

- **Initialization of classifiers:** In current approaches, L_i is initially formed from the available data in L . Depending on the particular method, they may use the same labeled data for each L_i or apply a bootstrapping to introduce diversity. As we showed before, both alternatives can lead to a lack of diversity when more than one classifier is used. To solve this, we promote the generation of different synthetic examples for each classifier C_i . In this way, the generation mechanism is applied a total of N times. Because L data are the most confident examples, we ensure that they belong to each L_i in conjunction with synthetic examples. Note that the generation method has some randomness, so different executions generate distinct synthetic points. This ensures the diversity between L_i sets.
- **Self-labeling stage:** After the initialization phase, each classifier is trained with its respective L_i . Then, the learned hypotheses are used to classify unlabeled points, determining the most reliable examples. There are several ways to perform this operation. Single classifier approaches extract their confidence from the base classifier and multiple classifiers calculate confidence predictions in terms of the agreement or combination of hypotheses. Independently of the procedure followed, each classifier

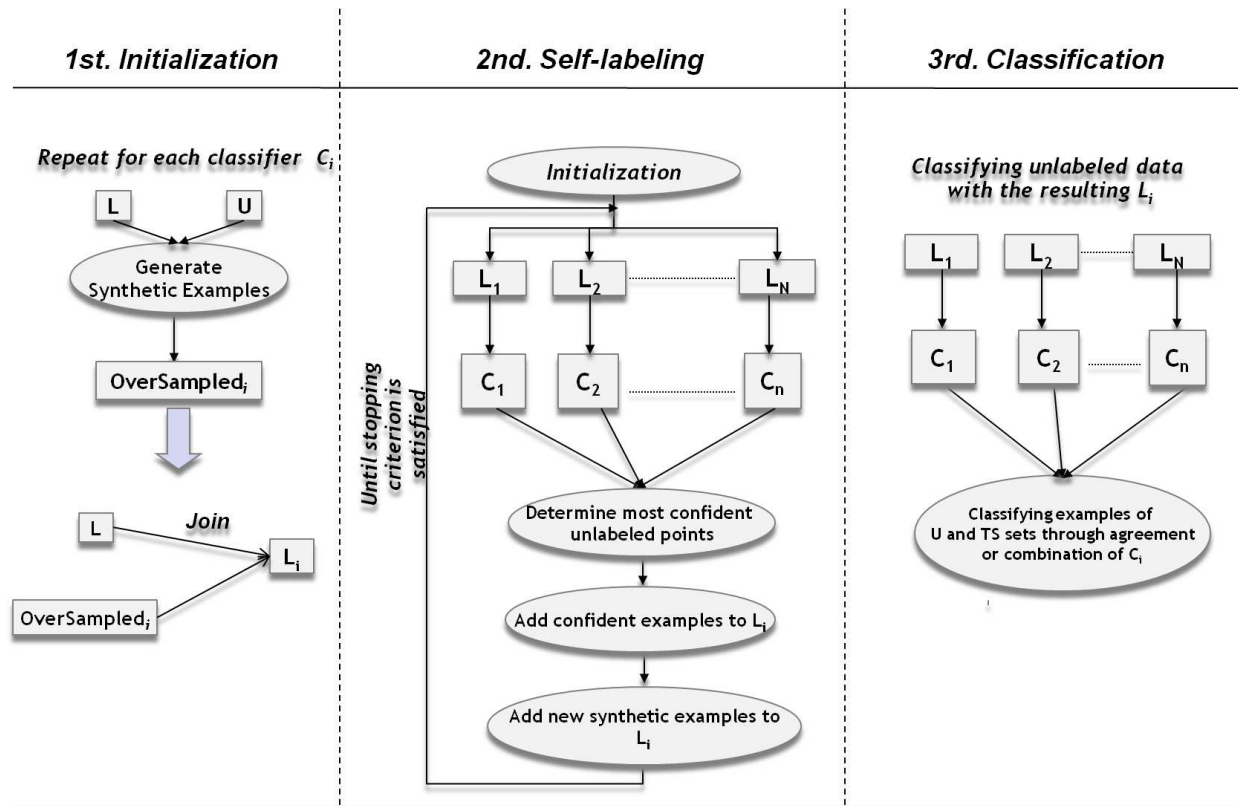


Fig. 3. SEG-SSC flowchart

TABLE I
MAIN CHARACTERISTICS OF SELECTED SELF-LABELED METHODS

Algorithm	Initialization	Classifiers	Teaching scheme	Confidence rule
Democratic-Co	Simple	Different learning algorithms	Self-teaching	Weighted majority
Tri-Training	Bootstrapping	Same learning algorithms	Mutual-teaching	Majority
Co-Forest	Bootstrapping	Same learning algorithms	Self-teaching	Majority
Co-Bagging	Simple	Same learning algorithms	Self-teaching	Majority

obtains a set L'_i that will be used to enlarge L_i . At this point, there are two possibilities: self or mutual teaching. The former uses its own predictions to augment its L_i . With a mutual teaching approach, a classifier C_j teaches its confidence predictions to the rest of the classifiers, that is, it increases $L_i, \forall i \neq j$. When all the L_i are increased, a new oversampling stage is performed for each L_i , using its prototypes and the remaining unlabeled examples. The resulting L_i sets are ready to be used in the next iteration.

- **Final classification:** The stopping criteria depends on the specific self-labeled method used, which is usually defined by a given number of iterations or by the condition of the learned hypotheses of the classifiers used, which does not change. When it is satisfied, not all the unlabeled instances have had to be added to one of the L_i sets. For this reason, the resulting L_i sets have to be used to classify the remaining instances of U and the TS set.

As such, this scheme is applicable to any self-labeling method and should provide better generalization capabilities to all of them. To test the proposed framework, we have applied these ideas to four self-labeling approaches: Democratic-Co

[34], Tri-Training [35], Co-Forest [21] and Co-Bagging [37], [38]. These models have different characteristics, such as distinct mechanisms to determine confident examples (agreement or combination), teaching schemes, uses of different learning algorithms or having a different initialization scheme. Table I summarizes the main properties of these models. We modify these models by adding synthetic examples, as explained above, to have an idea of how flexible our framework is. The modified versions of these algorithms will be denoted: SEG-SSC+Democratic-Co, SEG-SSC+Tri-Training, SEG-SSC+Co-Forest and SEG-SS+Co-Bagging.

As an additional comment of the proposed model, we can note that the generation of synthetic data is based on meta-heuristics that may lack of solid theoretical insights. In the specialized literature, this kind of techniques, such as [23], [50], does not provide any theoretical analyses because of the stochastic nature of the models. However, their applicability and effectiveness has been proved in many real world applications [52], [53]. This fact motivates the large experimental study that we will perform in the following section to support the usefulness and soundness of our model.

IV. EXPERIMENTAL SETUP AND ANALYSIS OF RESULTS

This section presents all of the issues related to the experimental framework used in this work and the analysis of results. Section IV-A describes the main properties of the data sets used and the parameters of the selected algorithms. Section IV-B presents and analyzes the results obtained with

TABLE II
SUMMARY DESCRIPTION OF STANDARD CLASSIFICATION DATA SETS

Data set	#Ex.	#D.	# ω .	Data set	#Ex.	#D.	# ω .
abalone	4174	8	28	movement_libras	360	90	15
appendicitis	106	7	2	mushroom	8124	22	2
australian	690	14	2	nursery	12 690	8	5
autos	205	25	6	pageblocks	5472	10	5
banana	5300	2	2	penbased	10 992	16	10
breast	286	9	2	phoneme	5404	5	2
bupa	345	6	2	pima	768	8	2
chess	3196	36	2	ring	7400	20	2
cleveland	297	13	5	saheart	462	9	2
coil2000	9822	85	2	satimage	6435	36	7
contraceptive	1473	9	3	segment	2310	19	7
crx	125	15	2	sonar	208	60	2
dermatology	366	33	6	spambase	4597	55	2
ecoli	336	7	8	spectheart	267	44	2
flare-solar	1066	9	2	splice	3190	60	3
german	1000	20	2	tae	151	5	3
glass	214	9	7	texture	5500	40	11
haberman	306	3	2	tic-tac-toe	958	9	2
heart	270	13	2	thyroid	7200	21	3
hepatitis	155	19	2	titanic	2201	3	2
housevotes	435	16	2	twonorm	7400	20	2
iris	150	4	3	vehicle	846	18	4
led7digit	500	7	10	vowel	990	13	11
lymphography	148	18	4	wine	178	13	3
magic	19 020	10	2	wisconsin	683	9	2
mammographic	961	5	2	yeast	1484	8	10
marketing	8993	13	9	zoo	101	17	7
monks	432	6	2				

TABLE III
SUMMARY DESCRIPTION OF HIGH DIMENSIONAL DATA SETS

Data set	#Ex.	#D.	# ω .	Reference
bci	400	117	2	
coil	1500	241	6	
coil2	1500	241	2	
digit1	1500	241	2	
g241c	1500	241	2	
g241n	1500	241	2	
sectr	83 679	315	2	
text	1500	11 960	2	
usps	1500	241	2	[2]
bbc	2225	9636	5	
bbcspport	737	4613	5	[57]

standard classification data sets. Finally, Section IV-C studies the behavior of the proposed framework when dealing with high dimensional problems.

A. Data sets and parameters

The experimentation is based on 55 standard classification data sets taken from the UCI repository [27] and the KEEL-dataset repository¹[26] and 11 high dimensional problems extracted from the book by Chapelle et al. [2] and the BBC News web page [57]. Tables II and III summarize the properties of the selected data sets. They show, for each data set, the number of examples (#Ex.), the number of attributes (#D.), and the number of classes (# ω). The standard classification data sets considered contain between 100 and 19,000 instances, the number of attributes ranges from 2 to 90 and the number of classes varies between 2 and 28. However, the 11 high dimensional data sets contain between 400 and 83,679 instances and the number of features oscillates from 117 to 11,960.

¹<http://sci2s.ugr.es/keel/datasets>

TABLE IV
PARAMETER SPECIFICATION FOR THE BASE LEARNERS AND THE SELF-LABELLED METHODS USED IN THE EXPERIMENTATION

Algorithm	Parameters
KNN C4.5	Number of Neighbors = 3, Euclidean Distance Confidence level: $c = 0.25$ Minimum number of item-sets per leaf: $i = 2$ Prune after the tree building
Democratic-Co Tri-Training Co-Forest Co-Bagging	Classifiers = 3NN, C4.5, NB No parameters specified Number of RandomForest Classifiers = 6, Threshold = 0.75 $MAX_ITER = 40$, Committee members = 3 Ensemble Learning = Bagging, Pool U = 100
SEG-SSC Differential evolution parameters	Oversampling factor=0.25, Number of Neighbors = 5 Iterations = 100, iterSFGSS = 8, iterSFHC = 20 FI=0.1, Fu=0.9

All the data sets have been partitioned using the 10 fold cross-validation procedure, that is, the data set has been split into 10 folds, each one containing 10% of the examples of the data set. For each fold, an algorithm is trained with the examples contained in the rest of the folds (training partition) and then tested with the current fold. Note that test partitions are kept aside to assess the performance of the learned hypothesis.

Each training partition is then divided into two parts: labeled and unlabeled examples. Using the recommendation established in [41], in the division process we do not maintain the class proportion in the labeled and unlabeled sets since the main aim of SSC is to exploit unlabeled data for better classification results. Hence, we use a random selection of examples that will be marked as labeled instances, and the class label of the rest of the instances will be removed. We ensure that every class has at least one representative instance. In standard classification data sets we have taken a labeled ratio of 10%. For high dimensional data sets, we will use two splits for training partitions with 10 and 100 labeled examples, respectively. In both cases, the remaining instances are marked as unlabeled points.

Regarding the parameters of the algorithms, the selected values are fixed for all problems, and they have been chosen according to the recommendation of the corresponding authors of each algorithm. From our point of view, the approaches analyzed should be as general and as flexible as possible. It is known that a good choice of parameters boosts their better performance over different data sources, but their way of working should offer good enough results in spite of the fact that the parameters are not optimized for a specific data set. This is the main purpose of this experimental setup, to show how the proposed framework can improve the efficacy of self-labeled techniques. Table IV specifies the configuration parameters of all the methods. Because these algorithms carry out some random operations during the labeling process, they have been run three times per partition.

In this table, we also present the parameters involved in our framework: the oversampling factor, the number of neighbors and the parameters needed for the differential evolution optimization. They can also be adjusted for each problem, however, with the same aim of being as flexible as possible. We have fixed these values empirically in previous experiments. The parameters used for the differential evolution optimization

are the same as those established in [54], except for the number of iterations that have been reduced. We decrease this value because, under this framework, the reference set used by differential evolution contains a smaller number of instances than in the case of supervised learning.

The Co-Forest and Democratic-Co algorithms were designed and tested with determined base classifiers. In this study, these algorithms maintain their classifiers. However, the interchange of the base classifiers is allowed in the Tri-Training and Co-Bagging approaches. In these cases, we will test two base classifiers, the K-Nearest Neighbor [58] and the C4.5 algorithms [59]. A brief description of these base classifiers and their associated confidence prediction computation are given as follows:

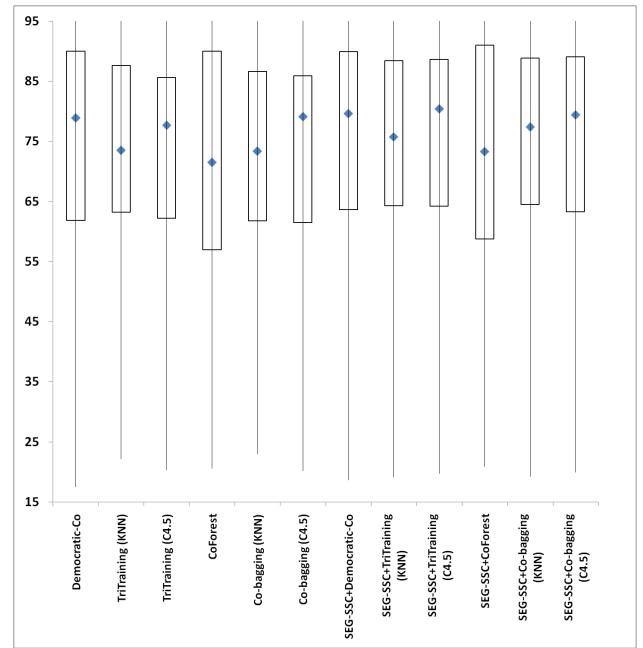
- *K-Nearest Neighbor (KNN)*: This is an instance-based learning algorithm that belongs to the lazy learning family of methods [60]. As such, it does not build a model during the learning process and is based on dissimilarities among a set of instances. For those self-labeled methods that need to estimate confidence predictions from this classifier, they can approximate it in terms of distance from the currently labeled set.
- *C4.5*: This is a decision tree algorithm [59] that induces classification rules for a given training set. The decision tree is built with a top-down scheme, using the normalized information gain (difference in entropy) that is obtained from choosing an attribute for splitting the data. The attribute with the highest normalized information gain is the one used to make the decision. Confidence predictions can be obtained from the accuracy of the leaf that makes the prediction. The accuracy of a leaf is the percentage of correctly classified train examples from the total number of covered train instances.

B. Experiments on standard classification data sets

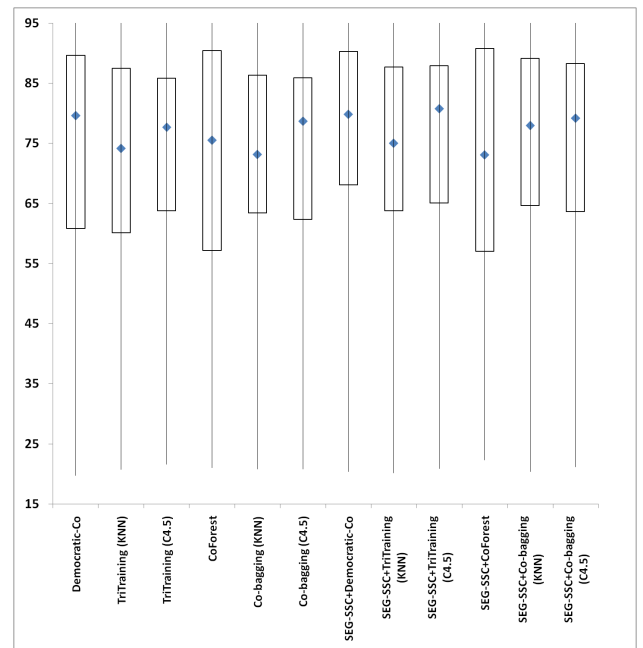
In this subsection we compare the modified versions of the selected self-labeled methods (within SEG-SSC) with the original ones, focusing on the results obtained on the 55 standard classification data sets and a labeled ratio of 10%. We analyze the transductive and inductive accuracy capabilities of these methods. Both results are presented in Tables V and VI, respectively. In these tables, we have specified the base classifier between brackets for Tri-Training and Co-Bagging algorithms. The best result of each row has been highlighted in bold face.

Aside from these tables, Figure 4 depicts two box plot representations of the results obtained in transductive and inductive settings, respectively. With these box plots we show a graphical comparison of the performance of the algorithms, indicating their most important characteristics such as the median, extreme values and spread of values about the median in the form of quartiles (Q1 and Q3).

Observing these tables and the figure we can appreciate differences between each of the original proposals and the improvement achieved by the addition of synthetic examples. Nevertheless, the use of hypothesis testing methods is mandatory in order to contrast the results of a new proposal with



(a) Transductive accuracy



(b) Inductive accuracy

Fig. 4. Box plot of transductive and inductive accuracy rates. The boxes contain 50% of the data (Q1 to Q3), blue points are the median values and the lines extend to the most extreme values

several comparison methods. The aim of these techniques is to identify the most relevant differences found between methods, which is highly recommended in the data mining field [29]. To do this, we focus on the Wilcoxon signed-ranks test [61] because it establishes a pairwise comparison between methods. In this way, we can see if there are significant differences between the original and modified versions. More information about this test and other statistical procedures can be found at <http://sci2s.ugr.es/sicidm/>.

Table VII collects the results of the application of the

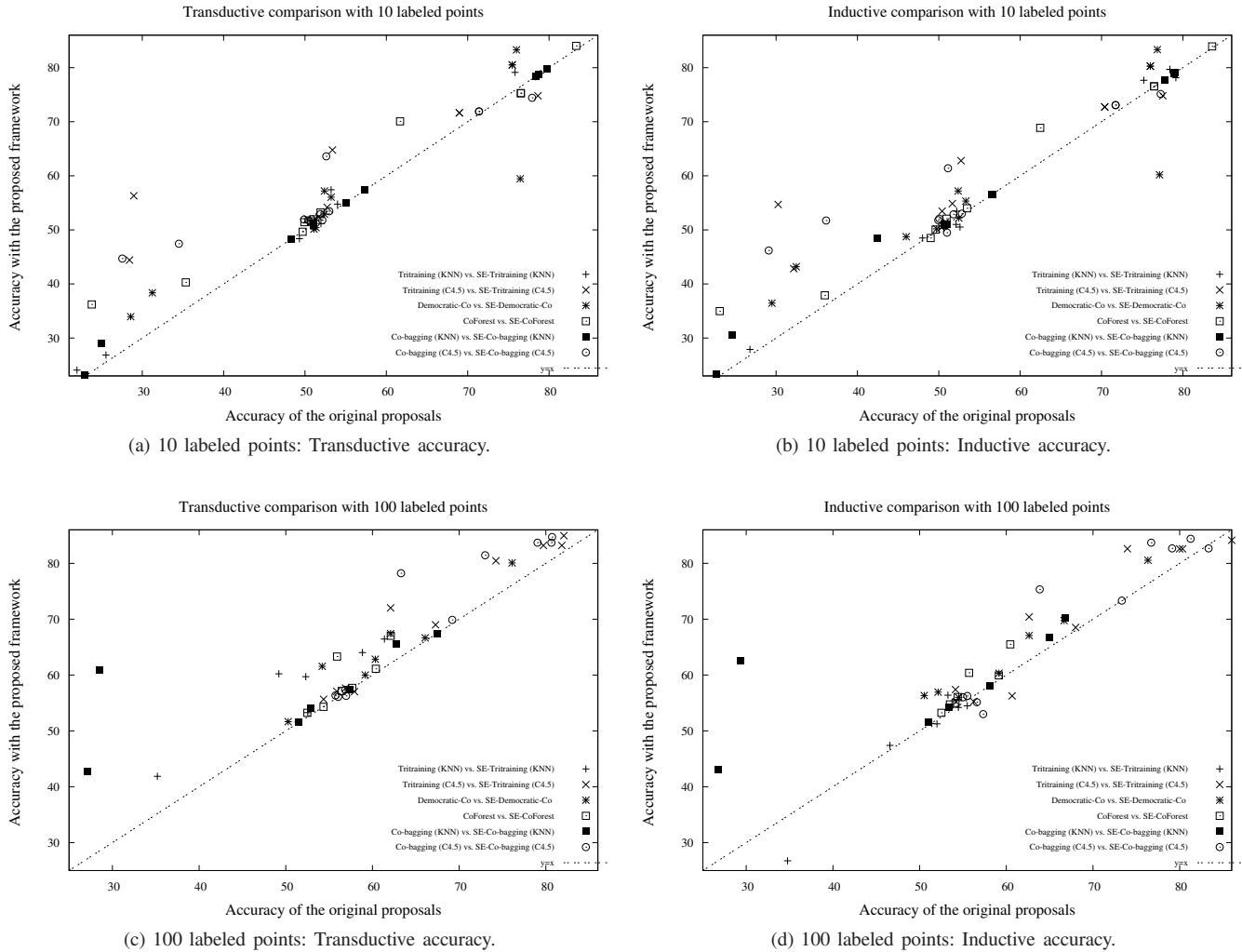


Fig. 5. High dimensional data sets: Transductive and inductive accuracy results

11 data sets considered, including transductive and inductive phases for both 10 and 100 splits.

TABLE VIII
HIGH DIMENSIONAL DATA SETS: AVERAGE RESULTS OBTAINED IN TRANSDUCTIVE (TRS) AND INDUCTIVE (TST) PHASES.

#L	Phase	Democratic-Co	Tri-Training (KNN)	Tri-Training (C4.5)	Co-Forest	Co-Bagging (KNN)	Co-Bagging (C4.5)
10	TRS	56.6987	56.1988	53.1233	55.5031	56.8976	53.8810
	TST	56.5331	55.9070	53.3841	55.6181	56.3126	53.8852
100	TRS	70.8563	66.5833	68.7295	68.0144	65.4552	68.7148
	TST	70.1286	65.4899	69.1031	67.1261	66.1272	69.7143

#L	Phase	SEG-SSC+ Democratic-Co	SEG-SSC+ Tri-Training (KNN)	SEG-SSC+ Tri-Training (C4.5)	SEG-SSC+ Co-Forest	SEG-SSC+ Co-Bagging (KNN)	SEG-SSC+ Co-Bagging (C4.5)
10	TRS	58.4330	57.1121	58.7192	58.0919	57.2811	57.8010
	TST	58.8520	56.0891	58.3297	57.6709	57.4786	58.1651
100	TRS	73.4449	70.1754	71.7133	70.1887	70.4770	72.4768
	TST	73.5080	65.1796	71.1483	69.2605	71.0570	71.9971

Given Figure 5 and Table VIII, we can make the following comments:

- In all the plots of Figure 5, most of the points are above the $y = x$ line, which means that, with the proposed framework, the self-labeled techniques perform better than the original algorithms. Differentiating between 10

and 100 available labeled points, we can see that when we have 100 labeled examples, there are more points above this line in both the transductive and inductive phases. We do not discern great differences between the performance obtained in both learning phases which shows that the hypotheses learned with the available labeled and unlabeled data were appropriate.

- Table VIII shows that, on average, the proposed scheme obtains a better performance level than the original ones in most cases, independently of the learning phase and the number of labeled data considered. Attending to the difference between transductive and inductive results, we observe that, in general, SEG-SSC increments both proportionally. Nevertheless, there are significant differences between the results obtained with 10 and 100 labeled points.
- With these results in mind, we can see the good synergy between synthetic examples and self-labeled techniques in these domains, but, what are the main differences with the results obtained in the previous subsection? We observe great differences between those algorithms that

use KNN as a base classifier and those that use C4.5. With standard classification data sets, we ascertained that C4.5 was the best base classifier for Tri-Training and performs similarly to KNN for Co-Bagging. These statements are maintained in these domains, where C4.5 performs better. In this study, SEG-SSC+Democratic may be highlighted as the best performing model, obtaining the highest transductive and inductive accuracy results with 10 and 100 labeled examples.

V. CONCLUDING REMARKS

In this paper we have developed a novel framework called SEG-SSC to improve the performance of any self-labeled semi-supervised classification method. It is focused on the idea of using synthetic examples in order to diminish the drawbacks occasioned by the absence of labeled examples, which deteriorates the efficiency of this family of methods.

The proposed self-labeled scheme with synthetic examples has been incorporated in four well-known self-labeled techniques that have been modified by introducing the necessary elements to follow the designed framework. These models are able to overcome the original self-labeled methods due to the fact that the addition of new labeled data implies a better diversity of multiple classifier approaches and fulfills the distribution of labeled data.

The wide experimental study carried out has allowed us to investigate the behavior of the proposed scheme with a high number of data sets with a varied number of instances and features. The results have been statistically compared, supporting the assertion that our proposal is a suitable tool for enhancing self-labeled methods.

Among the used data sets, we have tackled problems related to diverse applications with a high practical interest. For instance, our model can be used to address practical problems such as computer-aided diagnosis, image-classification, spam filtering, etc [21], [47].

There are many possible variations of our proposed semi-supervised scheme that could be interesting to explore as future work. In our opinion, the use of oversampling techniques with self-labeled techniques is not only a new way to improve the capabilities of this family of techniques, but could also be useful for most of the existing semi-supervised learning algorithms.

REFERENCES

- [1] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011.
- [2] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning*, 1st ed. The MIT Press, 2006.
- [3] X. Zhu and A. B. Goldberg, *Introduction to Semi-Supervised Learning*, 1st ed. Morgan and Claypool, 2009.
- [4] F. Schwenker and E. Trentin, "Pattern classification and clustering: A review of partially supervised learning approaches," *Pattern Recognition Letters*, vol. 37, pp. 4 – 14, 2014.
- [5] K. Chen and S. Wang, "Semi-supervised learning via regularized boosting working on multiple semi-supervised assumptions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 129–143, 2011.
- [6] G. Wang, F. Wang, T. Chen, D.-Y. Yeung, and F. Lochovsky, "Solution path for manifold regularized semisupervised classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 2, pp. 308–319, 2012.
- [7] A. Blum and S. Chawla, "Learning from labeled and unlabeled data using graph mincuts," in *Proceedings of the Eighteenth International Conference on Machine Learning*, 2001, pp. 19–26.
- [8] J. Wang, T. Jebara, and S.-F. Chang, "Semi-supervised learning using greedy max-cut," *Journal of Machine Learning Research*, vol. 14, no. 1, pp. 771–800, 2013.
- [9] A. Fujino, N. Ueda, and K. Saito, "Semisupervised learning for a hybrid generative/discriminative classifier based on the maximum entropy principle," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 3, pp. 424–437, 2008.
- [10] T. Joachims, "Transductive inference for text classification using support vector machines," in *Proc. 16th International Conference on Machine Learning*. Morgan Kaufmann, 1999, pp. 200–209.
- [11] P. Kumar Mallapragada, R. Jin, A. Jain, and Y. Liu, "Semiboost: Boosting for semi-supervised learning," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 11, pp. 2000–2014, 2009.
- [12] Q. Wang, P. Yuen, and G. Feng, "Semi-supervised metric learning via topology preserving multiple semi-supervised assumptions," *Pattern Recognition*, vol. 46, no. 9, pp. 2576–2587, 2013.
- [13] I. Triguero, S. García, and F. Herrera, "Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study," *Knowledge and Information Systems*, pp. 1–40, 2014, in press, doi: 10.1007/s10115-013-0706-y.
- [14] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, 1995, pp. 189–196.
- [15] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with Co-Training," in *Proceedings of the Annual ACM Conference on Computational Learning Theory*, 1998, pp. 92–100.
- [16] K. Bennett, A. Demiriz, and R. Maclin, "Exploiting unlabeled data in ensemble methods," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002, pp. 289–296.
- [17] Z.-H. Zhou and M. Li, "Semi-supervised learning by disagreement," *Knowl. Inf. Syst.*, vol. 24, no. 3, pp. 415–439, 2010.
- [18] G. Jin and R. Raich, "Hinge loss bound approach for surrogate supervision multi-view learning," *Pattern Recognition Letters*, vol. 37, pp. 143 – 150, 2014.
- [19] U. Maulik and D. Chakraborty, "A self-trained ensemble with semisupervised svm: An application to pixel classification of remote sensing imagery," *Pattern Recognition*, vol. 44, no. 3, pp. 615 – 623, 2011.
- [20] A. Joshi and N. Papanikolopoulos, "Learning to detect moving shadows in dynamic environments," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 11, pp. 2055–2063, nov. 2008.
- [21] M. Li and Z. H. Zhou, "Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 37, no. 6, pp. 1088–1098, 2007.
- [22] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, pp. 123–140, August 1996.
- [23] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.
- [24] I. Triguero, S. García, and F. Herrera, "Differential evolution for optimizing the positioning of prototypes in nearest neighbor classification," *Pattern Recognition*, vol. 44, no. 4, pp. 901–916, 2011.
- [25] K. V. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution A Practical Approach to Global Optimization*, ser. Natural Computing Series, G. Rozenberg, T. Bäck, A. E. Eiben, J. N. Kok, and H. P. Späink, Eds., 2005.
- [26] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *Journal of Multiple-Valued Logic and Soft Computing*, vol. 17, no. 2-3, pp. 255–277, 2011.
- [27] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: <http://archive.ics.uci.edu/ml>
- [28] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.
- [29] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced non-parametric tests for multiple comparisons in the design of experiments

- in computational intelligence and data mining: Experimental analysis of power," *Information Sciences*, vol. 180, pp. 2044–2064, 2010.
- [30] M. Li and Z. H. Zhou, "SETRED: self-training with editing," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3518 LNAI, 2005, pp. 611–621.
- [31] S. Dasgupta, M. L. Littman, and D. A. McAllester, "PAC generalization bounds for co-training," in *Advances in Neural Information Processing Systems 14, Neural Information Processing Systems: Natural and Synthetic*, 2001, pp. 375–382.
- [32] J. Du, C. X. Ling, and Z. H. Zhou, "When does co-training work in real data?" *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 5, pp. 788–799, 2010.
- [33] S. Goldman and Y. Zhou, "Enhancing supervised learning with unlabeled data," in *In proceedings of the 17th International Conference on Machine Learning*. Morgan Kaufmann, 2000, pp. 327–334.
- [34] Y. Zhou and S. Goldman, "Democratic co-learning," in *Tools with Artificial Intelligence, IEEE International Conference on*, 2004, pp. 594–202.
- [35] Z. H. Zhou and M. Li, "Tri-training: Exploiting unlabeled data using three classifiers," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, pp. 1529–1541, 2005.
- [36] L. B. Statistics and L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [37] M. Hady and F. Schwenker, "Combining committee-based semi-supervised learning and active learning," *Journal of Computer Science and Technology*, vol. 25, pp. 681–698, 2010.
- [38] M. Hady, F. Schwenker, and G. Palm, "Semi-supervised learning for tree-structured ensembles of rbf networks with co-training," *Neural Networks*, vol. 23, pp. 497–509, 2010.
- [39] Y. Yaslan and Z. Cataltepe, "Co-training with relevant random subspaces," *Neurocomput.*, vol. 73, no. 10–12, pp. 1652–1661, 2010.
- [40] T. Huang, Y. Yu, G. Guo, and K. Li, "A classification algorithm based on local cluster centers with a few labeled training examples," *Knowledge-Based Systems*, vol. 23, no. 6, pp. 563–571, 2010.
- [41] Y. Wang, X. Xu, H. Zhao, and Z. Hua, "Semi-supervised learning based on nearest neighbor rule and cut edges," *Knowledge-Based Systems*, vol. 23, no. 6, pp. 547–554, 2010.
- [42] S. Sun and Q. Zhang, "Multiple-view multiple-learner semi-supervised learning," *Neural Processing Letters*, vol. 34, no. 3, pp. 229–240, 2011.
- [43] A. Halder, S. Ghosh, and A. Ghosh, "Aggregation pheromone metaphor for semi-supervised classification," *Pattern Recognition*, vol. 46, no. 8, pp. 2239–2248, 2013.
- [44] M.-L. Zhang and Z.-H. Zhou, "CoTrade: Confident co-training with data editing," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 41, no. 6, pp. 1612–1626, 2011.
- [45] I. Triguero, J. A. Sáez, J. Luengo, S. García, and F. Herrera, "On the characterization of noise filters for self-training semi-supervised in nearest neighbor classification," *Neurocomputing*, 2013, , in press, doi: 10.1016/j.neucom.2013.05.055.
- [46] I. T. Jolliffe, *Principal Component Analysis*. Berlin; New York: Springer-Verlag, 1986.
- [47] C. Deng and M. Guo, "A new co-training-style random forest for computer aided diagnosis," *Journal of Intelligent Information Systems*, vol. 36, pp. 253–281, 2011.
- [48] Y. Sun, A. K. C. Wong, and M. S. Kamel, "Classification of imbalanced data: A review," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 04, pp. 687–719, 2009.
- [49] H. He and E. Garcia, "Learning from imbalanced data," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 21, no. 9, pp. 1263–1284, 2009.
- [50] S. García, J. Derrac, I. Triguero, C. J. Carmona, and F. Herrera, "Evolutionary-based selection of generalized instances for imbalanced classification," *Know-Based Syst.*, vol. 25, no. 1, pp. 3–12, 2012.
- [51] H. Zhang and M. Li, "Rwo-sampling: A random walk over-sampling approach to imbalanced data classification," *Information Fusion*, 2014, in press, doi: 10.1016/j.inffus.2013.12.003.
- [52] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics," *Information Sciences*, vol. 250, pp. 113 – 141, 2013.
- [53] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behaviour of several methods for balancing machine learning training data," *SIGKDD Explorations*, vol. 6, no. 1, pp. 20–29, 2004.
- [54] I. Triguero, S. García, and F. Herrera, "IPADE: Iterative prototype adjustment for nearest neighbor classification," *IEEE Transactions on Neural Networks*, vol. 21, no. 12, pp. 1984–1990, 2010.
- [55] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*. Springer-Verlag, Berlin, 2003.
- [56] S. Das and P. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.
- [57] "BBC datasets," 2014. [Online]. Available: <http://mlg.ucd.ie/datasets/bbc.html>
- [58] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.
- [59] J. R. Quinlan, *C4.5: programs for machine learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers, 1993.
- [60] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Machine Learning*, vol. 6, no. 1, pp. 37–66, 1991.
- [61] F. Wilcoxon, "Individual Comparisons by Ranking Methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.
- [62] Z. Jiang, S. Zhang, and J. Zeng, "A hybrid generative/discriminative method for semi-supervised classification," *Knowledge-Based Systems*, vol. 37, pp. 137–145, 2013.
- [63] W. Li, L. Duan, I. Tsang, and D. Xu, "Co-labeling: A new multi-view learning approach for ambiguous problems," in *Proceedings - IEEE International Conference on Data Mining, ICDM, 2012*, pp. 419–428.



Isaac Triguero received the M.Sc. and Ph.D. degree in Computer Science from the University of Granada, Granada, Spain, in 2009 and 2014, respectively.

He is currently researcher in the Department of Computer Science and Artificial Intelligence, University of Granada, Granada, Spain. His research interests include data mining, data reduction, biometrics, evolutionary algorithms and semi-supervised learning.



Salvador García received the M.Sc. and Ph.D. degrees in Computer Science from the University of Granada, Granada, Spain, in 2004 and 2008, respectively.

He is currently an Associate Professor in the Department of Computer Science, University of Jaén, Jaén, Spain. He has published more than 40 papers in international journals. As edited activities, he has co-edited two special issues in international journals on different Data Mining topics and is member of the editorial board of the Information Fusion journal.

His research interests include data mining, data reduction, data complexity, imbalanced learning, semi-supervised learning, statistical inference and evolutionary algorithms.



Francisco Herrera received his M.Sc. in Mathematics in 1988 and Ph.D. in Mathematics in 1991, both from the University of Granada, Spain.

He is currently a Professor in the Department of Computer Science and Artificial Intelligence at the University of Granada. He has published more than 240 papers in international journals. He is coauthor of the book "Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases" (World Scientific, 2001).

He currently acts as Editor in Chief of the international journals "Information Fusion" (Elsevier) and "Progress in Artificial Intelligence" (Springer). He acts as area editor of the International Journal of Computational Intelligence Systems and associated editor of the journals: IEEE Transactions on Fuzzy Systems, Information Sciences, Knowledge and Information Systems, Advances in Fuzzy Systems, and International Journal of Applied Metaheuristics Computing; and he serves as member of several journal editorial boards, among others: Fuzzy Sets and Systems, Applied Intelligence, Information Fusion, Evolutionary Intelligence, International Journal of Hybrid Intelligent Systems, Memetic Computation, and Swarm and Evolutionary Computation.

He received the following honors and awards: ECCAI Fellow 2009, 2010 Spanish National Award on Computer Science ARITMEL to the "Spanish Engineer on Computer Science", International Cajastur "Mamdani" Prize for Soft Computing (Fourth Edition, 2010), IEEE Transactions on Fuzzy System Outstanding 2008 Paper Award (bestowed in 2011), and 2011 Lotfi A. Zadeh Prize Best paper Award of the International Fuzzy Systems Association.

His current research interests include computing with words and decision making, bibliometrics, data mining, biometrics, data preparation, instance selection, fuzzy rule based systems, genetic fuzzy systems, knowledge extraction based on evolutionary algorithms, memetic algorithms and genetic algorithms.