Journal of Intelligent & Fuzzy Systems 1–13
IOS Press

CST-Voting: A semi-supervised ensemble method for classification problems

G. Kostopoulos [a,*], I.E. Livieris [b], S. Kotsiantis [a], V. Tampakas [b]

[a] *Educational Software Development Laboratory (ESDLab), Department of Mathematics, University of Patras, GR 265-00, Greece.*

[b] *Department of Computer & Informatics Engineering (DISK Lab), Technological Educational Institution of Western Greece, Greece, GR 263-34.*

Abstract. Semi-supervised learning is an emerging subfield of machine learning, with a view to building efficient classifiers exploiting a limited pool of labeled data together with a large pool of unlabeled ones. Most of the studies regarding semi-supervised learning deal with classification problems, whose goal is to learn a function that maps an unlabeled instance into a finite number of classes. In this paper, a new semi-supervised classification algorithm, which is based on a voting methodology, is proposed. The term attributed to this ensemble method is called CST-Voting. Ensemble methods have been effectively applied in various scientific fields and often perform better than the individual classifiers from which they are originated. The efficiency of the proposed algorithm is compared to three familiar semi-supervised learning methods on a plethora of standard benchmark datasets using three representative supervised classifiers as base learners. Experimental results demonstrate the predominance of the proposed method, outperforming classical semi-supervised classification algorithms as illustrated from the accuracy measurements and confirmed by the Friedman Aligned Ranks nonparametric test.

Keywords: Semi-supervised learning, classification, voting, ensemble methods, accuracy

1. Introduction

Learning from examples is a method that has been extensively analyzed in machine learning, a fast growing subfield of computer science. Over recent years, different machine learning approaches have been applied in several scientific fields, such as supervised, unsupervised and semi-supervised learning. Supervised learning deals with problems in which the output classes of the instances in the training set are known, while in unsupervised learning there is no knowledge regarding the output classes [34].

Semi-supervised learning (SSL) is a combination of supervised and unsupervised learning aiming to obtain better results from each one of these methods exploiting a small pool of l labeled examples $L_d = \{(x_1,y_1),(x_2,y_2),\ldots,(x_l,y_l)\}$ with $x_i \in \mathbb{R}^n$, $i = 1,2,\ldots,l$ together with a large pool of k unlabeled examples $U_d = \{x_1,x_2,\ldots,x_k\}$ with $x_i \in \mathbb{R}^n$, $i = 1,2,\ldots,k$. Since it is difficult and expensive to obtain a fully labeled dataset in many real world problems, SSL has turned to a powerful and effective machine learning tool for learning from both labeled and unlabeled data [42].

Depending on the nature of the output class, SSL is divided into two main tasks: semi-supervised classification (SSC) for discrete output class and semi-supervised regression (SSR) for real-valued. Most of the studies about SSL deal with classification problems, with a view to predicting a label from a finite set of class labels. The most frequently studied classification problem is considered to be the binary classification, where the output variable $y \in \{0,1\}$, while in multi-class classification $y \in \{0,1,2,\ldots,n\}$. SSC can be either inductive or transductive [42]. Inductive SSC aims to learn a classifier for future unknown data, while transductive SSC classifies instances from the unlabeled dataset.

*Corresponding author. E-mail: kostg@sch.gr.

Several ensemble methods have also emerged recently, combining different classifiers for the improvement of the classification accuracy [21]. Ensemble learning or committee-based learning or learning with multiple classifier systems, concerns the formulation and training of a set of classifiers [40] for classifying new instances, usually through an iterative voting procedure, that is based on classifiers' individual predictions. There is a growing apprehension that ensemble methods outperform each one of the single classifiers within the ensemble, especially when these component classifiers are as accurate and diverse as possible [9]. Accordingly, ensemble learning has become an increasingly widespread methodology for building powerful and accurate predictive models.

In view of the above, SSL methods and ensemble methods constitute two significant machine learning paradigms. The former attempts to achieve strong generalization by exploiting the hidden information on unlabeled data while the latter attempts to achieve strong generalization by using multiple classifiers [40]. Although both methodologies have been applied effectively in a variety of scientific fields during the past decade, they were almost developed separately. Zhou [40] showed that SSL and ensemble learning may be beneficial to each other, since unlabeled data are in abundance and can be exploited to the full by a combination of diverse classifiers.

Following up the results of this study, in the present paper an attempt is made to put forward an ensemble of SSC algorithms. Therefore, a new semi-supervised ensemble algorithm, called CST-Voting, is proposed. CST-Voting combines three representative SSL algorithms, and in particular Co-training [4], Self-training [37] and Tri-training [41], and is based on a voting methodology. The efficiency of the proposed algorithm is evaluated with a number of benchmark datasets in terms of classification accuracy using three familiar supervised classifiers as base learners, while several experiments are carried out showing its efficacy.

The rest of this paper is organized as follows: Section 2 refers to familiar SSL algorithms. The proposed ensemble method is described in detail in Section 3. The experiments' procedure and their results are presented and analyzed in Section 4, while comparing the proposed method to its component SSL algorithms. Finally, Section 5 concludes considering some further research topics for future work.

2. Semi-supervised techniques

In recent years, a number of familiar SSC algorithms have been implemented with remarkable results in many application areas. Self-training, Co-training and Tri-training constitute representative SSL methods trying to effectively exploit the unlabeled data as far as possible, since the utilization of unlabeled data is essential for their efficiency [14].

Self-training or self-teaching is considered to be a simple and effective SSL method. The self-training idea first appeared in Yarowsky's study concerning the implementation of an unsupervised algorithm for word sense disambiguation [37]. In accordance to Ng and Cardie [27], self-training is a "*single-view weakly supervised algorithm*" which is based on its own predictions on unlabeled data to teach itself. At first, a classifier is trained from a small amount of labeled data constituting the training set, which is then used for classifying a predefined number of unlabeled data [42]. The most confident predictions, i.e. the unlabeled examples that have a high probability of been assigned with the correct label, are added to the labeled training set and the classifier is retrained. The previous learning process is repeated until all unlabeled data are finally labeled. Self-training is considered to be an iterative bootstrapping method since it is based on its own predictions to teach itself. However, false predictions of the classifier on the initial steps frequently result to a large number of erroneous predictions [37]. Therefore, several techniques and filters have been applied to reduce the impact of misclassified instances during the initial stages of the learning process [42]. In [26], a soft-labeling approach is adopted to minimize the expected loss of a classifier, while in [10], a new SSC method is proposed to overcome false labeling during the learning process and improve the classification performance. According to this method, unlabeled data are only used to generate new synthetic data extending the amount of labeled data and the final classifier is learned in the extended feature space. Very recently, a new method called SSL Sparse Representation based Classification was introduced addressing the problem of a small number of noisy labeled examples [12].

Co-training [4] is a widely used method in SSL setting that was originally proposed by Blum and Mitchell and is principally based on the assumption that each example in the dataset can be partitioned into two distinct views. Moreover, each view is assumed to be sufficient to make correct classifications and the two views are considered to be conditionally

independent given the class label. Two learning algorithms are trained separately for each view using the initial labeled dataset and the most confident predictions of each algorithm on unlabeled data are used to augment the training set of the other algorithm through an iterative learning process until some stopping criterion is met, i.e. for a predefined number of iterations. In essence, Co-training is a "*two-view weakly supervised algorithm*" [27] since it uses the Self-training approach on each view. The efficacy of the method is influenced mainly by the appropriate selection of the two algorithms, as well as the existence of two different views and the above-mentioned assumptions. Nigam and Gani [28] showed that Co-training outperforms other SSL algorithms when there is a natural existence of two distinct and independent views. However, the existence of two independent views on a dataset can hardly be met in practice. Several variants of Co-training algorithm have been developed to overcome this hurdle, such as Democratic Co-training [39] and Tri-training [41]. A very recent approach employs an improved variant of Co-training algorithm for software defect prediction using random under-sampling technique [24].

Tri-Training is an improved single-view extension of the Co-training algorithm exploiting unlabeled data without relying on the existence of two views of instances [41]. Tri-training is a bagging ensemble of three classifiers [15], since they are initially constructed by Bagging [5] and trained on data subsets generated through bootstrap sampling from the original labeled training set. If two of the classifiers agree on the labeling of an unlabeled example, then this example is used to teach the third classifier. In contrast to other variants of the Co-learning approach, Tri-training does not require different supervised algorithms, leading to greater applicability and implementation of the algorithm in many real world datasets.

Several notable ensemble methods have also applied in the SSL setting mainly based on the Co-training paradigm, such Co-Forest [22] and CoBC [15]. In Co-Forest, an initial ensemble of random trees is trained on bootstrap sub-samples generated from the original labeled dataset, while in CoBC, a committee of diverse classifiers is used instead of redundant and independent views. In [36], a prediction model based on semi-supervised twin support vector machine is introduced for medical prognosis of patients. In a very recent study [38], a progressive SSL ensemble approach is introduced that handles high dimensional datasets through random subspaces, as well as datasets with a limited amount of labeled instances enlarging the training set by a progressive generation process and a self sample selection process. Following up the results of these studies, a new semi-supervised ensemble method is proposed and described in the next section.

3. Proposed method

In general, the generation of an ensemble of classifiers considers mainly two steps: Selection and Combination.

The selection of the appropriate component classifiers is considered to be an essential step towards obtaining highly accurate classifier systems [40]. The key points for the effectiveness of the method is the component classifiers to be as accurate and diverse as possible. A commonly used approach is to generate classifiers by applying different learning algorithms (with heterogeneous model representations) to a single dataset (see [25,32]). On this basis, the learning algorithms which constitute the proposed ensemble are: Co-training, Self-training and Tri-training. What all these methods have in common is that they are self-labeled methods operating in different ways, trying to take full advantage of the hidden information in unlabeled data. The crucial difference between them is the mechanism used to label unlabeled data. Co-training is a multi-view method, while Self-training and Tri-training are single-view methods. Moreover, Co-training and Tri-training are indeed ensemble methods, since they both make use of multiple classifiers.

The combination of the component learning algorithms takes place through several methodologies. The proposed ensemble incorporates a majority voting methodology, since it is a simple and easy to implement method for combining the individual predictions of component classifiers in an ensemble. According to this approach, the ensemble output is the one made by more than half of them.

The pseudo-code of the proposed ensemble method is shown in Algorithm 1. Initially, Self-training, Co-training and Tri-training algorithms are trained on the labeled dataset L and then applied on unlabeled dataset U. L is augmented incrementally and the process is repeated until some stopping criterion is met or U is empty. The final hypothesis of an unlabeled example of the test set, i.e. the ensemble output, is produced via majority voting.

Algorithm 1: **CST-Voting**

Input: D - Initial training dataset.
r - Ratio of labeled instances along D.
L - Set of labeled training instances.
U - Set of unlabeled training instances.
E - Ensemble of algorithms.
MaxIter - Maximum number of iterations.
ConLev - Confidence level.
C, h, h_1, h_2, h_3 - Base learners.

```
/* Co-training algorithm */
L1=L(V1), L2=L(V2), V1,V2 are two feature views of instances
Initially train h1, h2 for each view V1, V2 on L1, L2 respectively
repeat
    Compute h1, h2 predictions for all instances in U
    for each view
        choose the Most Confidence Predictions and add to the
        training set of the other
    end for
    Retrain h1, h2 for each view on new enlarged L1, L2 respec-
     tively
until U is empty

/* Self-training algorithm */
L3=L
Initially train h on L3.
for i = 1 to MaxIter do
    Apply h on U.
    Select instances with a predicted probability more than 90%
    per iteration (xMCP).
    Remove xMCP from U and add to L3.
    Retrain h on new enlarged L3.
end for

/* Tri-training algorithm */
L4=L
for i=1,2,3 do
    Train hi on L4.
end for
repeat
    for i=1,2,3 do
        for x ∈ U do
            L'={ }
            if hj(x)=hk(x) (j,k ≠ i)
                then L' = L' ∪ (x,hj(x))
            end if
            Retrain hi on L'.
        end for
    end for
until some stopping criterion is met or U is empty

Construct a set of algorithms, E(Co(C), Self(C), Tri(C))
/* Testing phase */
for each x from test set
    Apply Self-training, Co-training, Tri-training on x.
    Use majority vote to predict the label y* of x.
end for

Output: The labels of instances in the testing set
```

4. Experimental results

The experiments were based on 40 benchmark datasets from UCI Machine Learning Repository [23] and KEEL repository [2]. A brief description of datasets structure (number of instances, number of attributes and output classes) is presented in Table 2. These datasets have been partitioned using the 10-fold cross-validation procedure so that each fold had the same distribution as the entire dataset. For each dataset, 90% was used as training set, while the remaining 10% was used as testing set to evaluate the performance of the learning algorithms. The training partition of each fold was divided into labeled and unlabeled subsets according to a selected ratio value. In order to study the influence of the amount of labeled data, three different ratios were used: 10%, 20%, and 30%.

The performance of the proposed ensemble algorithm was compared to its component SSC algorithms, and in particular Self-training, Co-training and Tri-training in terms of classification accuracy. Accuracy is one of the most frequently used measures for assessing the overall effectiveness of a classification algorithm [31] and is defined as the percentage of correctly classified instances. Furthermore, the implementation code was written in Java, using Weka Machine Learning Toolkit [16]. A plethora of experiments were carried out on each dataset deploying three well-known supervised classifiers as base learners in each SSL method thus creating a total of 120 datasets (3 labeled ratios for 40 datasets). A brief description of the three supervised classifiers is given below:

- J48, a very effective classification algorithm for building decision trees. It is a Weka implementation of the C4.5 Decision Tree algorithm [30], a widely used classification algorithm categorizing instances to a predefined set of classes according to their attribute values from the root of a tree down to a leaf. The accuracy of a leaf corresponds to the percentage of correctly classified instances of the training set.
- JRip, a well-known inference and rule-based algorithm which was originally introduced by Cohen [8]. It is a java optimized version of the RIPPER (Repeated Incremental Pruning to Produce Error Reduction) algorithm, implemented in Weka. JRip is considered to be a very effective algorithm, especially on large samples with noisy data. Moreover, RIPPER produces error

rates competitive with C4.5 and better running times.

– kNN, a representative instance-structured learning algorithm [1] based on the assumption that similar examples are close to each other. More specifically, a distance function is used to predict the output class of an instance through identifying the most frequently found class among the k nearest neighbors of it. The function should minimize the distance between two equally classified examples.

Studies have shown that the above classifiers constitute some of the most effective and widely used data mining algorithms [35] for classification problems. Moreover, Co-training, Self-training and Tri-training algorithms often achieve impressive results when C4.5 is used as base learner, while Tri-training also performs well with kNN as base learner [33]. The configuration parameters of all the SSL methods and base learners used in the experiments is presented in Table 1. Regarding the base learners, the default parameter settings included in the Weka software were applied.

Table 1
Parameters configuration for all SSL methods and base learners

Algorithm	Parameters
kNN	Number of neighbors=3
	Euclidean distance
J48	Confidence factor used for pruning=0.25
	Minimum number of instances per leaf=2
	Number of folds used for reduced-error pruning=3
	Pruning is performed after tree building
JRip	Number of optimization runs=2
	Number of folds used for reduced-error pruning=3
	Minimum total weight of the instances in a rule=2.0
	Pruning is performed after tree building
Self-training	MaxIter=40
Co-training	MaxIter=40
	Initial unlabeled pool=75
Tri-training	No parameters specified

The experimental results using 10%, 20% and 30% labeled ratio are presented in Tables 3, 4 and 5 respectively.

Table 2
Brief description of datasets

Dataset	#Instances	#Features	#Classes
automobile	205	26	7
banana	5300	2	2
breast	286	9	2
bupa	345	6	2
cleveland	297	13	5
coil2000	9822	85	2
contraceptive	1473	9	3
crx	125	15	2
dermatology	366	33	6
german	1000	20	2
glass	214	9	7
haberman	306	3	2
heart	270	13	2
hepatitis	155	19	2
housevotes	435	16	2
iris	150	4	3
led7digit	500	7	10
magic	19020	10	2
mammographic	961	5	2
monk2	432	6	2
movement	360	90	15
mushroom	8124	22	2
page-blocks	5472	10	5
pendigits	10992	16	10
phoneme	5404	5	2
pima	768	8	2
ring	7400	20	2
satimage	6435	36	7
segment	2310	19	7
spambase	4597	55	2
splice	3190	60	3
texture	5500	40	11
thyroid	7200	21	3
tic-tac-toe	958	9	2
twonorm	7400	20	2
vehicle	846	18	4
wisconsin	683	9	2
wine	178	13	3
yeast	1484	8	10
zoo	101	17	7

Table 3
Classification accuracy (labeled ratio 10%)

Dataset	Self-Train (JRip)	Co-Train (JRip)	Tri-Train (JRip)	CST-Voting (JRip)	Self-Train (C4.5)	Co-Train (C4.5)	Tri-Train (C4.5)	CST-Voting (C4.5)	Self-Train (3NN)	Co-Train (3NN)	Tri-Train (3NN)	CST-Voting (3NN)
automobile	49.21	74.17	70.42	69.83	50.37	68.63	77.42	71.75	72.29	67.25	72.92	72.29
banana	72.83	74.40	72.74	72.96	74.94	77.83	74.51	75.13	74.75	78.38	74.96	75.34
breast	71.71	73.05	69.29	72.07	74.15	73.44	73.42	74.15	75.17	75.86	74.80	75.85
bupa	57.11	57.11	57.13	57.11	56.25	56.25	56.84	56.25	57.11	57.11	57.13	57.11
cleveland	54.45	53.78	54.45	54.45	53.43	54.09	55.44	54.76	56.75	56.76	57.73	57.09
coil2000	94.00	94.02	94.01	94.03	94.03	94.03	94.03	94.03	93.61	93.61	93.72	93.61
contraceptive	46.57	46.30	46.84	47.32	51.46	51.87	51.06	51.73	51.86	47.67	51.25	51.86
crx	87.29	86.38	88.07	87.29	85.74	85.59	85.91	86.06	85.77	84.84	86.06	86.38
dermatology	88.27	86.83	88.51	90.70	93.46	88.85	93.18	93.19	96.46	94.54	96.73	96.46
german	70.10	71.90	69.60	70.40	71.70	72.20	71.80	71.70	72.40	71.90	72.20	72.80
glass	41.95	66.19	66.73	62.94	70.15	54.68	68.74	68.31	68.59	66.17	71.84	72.77
haberman	71.92	71.59	72.57	71.91	72.87	72.87	72.87	72.87	73.55	73.55	74.22	73.55
heart	80.22	78.15	81.17	80.53	78.16	78.15	77.83	80.47	81.20	81.86	81.87	81.20
hepatitis	78.71	81.13	79.88	81.17	82.58	81.33	83.04	82.58	87.63	86.42	85.75	87.71
housevotes	97.03	97.03	97.03	97.03	96.12	96.12	96.12	96.12	91.87	91.87	91.87	91.87
iris	93.33	92.67	91.33	92.67	84.00	92.00	94.00	93.33	94.00	94.00	92.67	94.00
led7digit	50.20	69.60	69.80	70.80	46.00	50.20	70.80	54.40	64.20	61.00	72.40	70.00
magic	82.26	80.90	82.33	82.65	84.13	81.71	84.25	84.34	81.96	80.50	81.57	82.13
mammographic	84.46	82.89	83.98	84.22	85.06	82.77	84.10	85.06	82.53	83.13	83.13	82.41
monk2	67.14	67.14	67.14	67.14	67.14	67.14	67.14	67.14	67.14	67.14	67.14	67.14
movement	46.11	50.56	50.83	51.67	42.78	47.78	59.44	53.89	60.83	51.39	64.44	63.89
mushroom	100.0	100.0	100.0	100.0	100.0	99.89	100.0	100.0	100.0	100.0	100.0	100.0
page-blocks	96.05	95.45	96.51	96.45	96.49	95.19	96.51	96.62	96.25	95.61	96.33	96.18
pendigits	91.50	89.79	91.47	93.44	88.29	85.90	88.29	89.38	96.38	93.50	96.31	96.28
phoneme	80.61	79.46	80.79	81.38	81.31	80.51	81.66	81.70	82.20	80.74	82.24	82.42
pima	74.48	73.96	73.05	74.35	75.91	73.83	74.74	76.04	73.96	73.17	73.44	74.22
ring	91.70	92.42	91.88	93.24	81.49	79.66	81.62	85.42	62.01	60.77	62.77	61.19
satimage	83.56	83.56	82.83	85.45	84.09	83.42	84.46	85.25	89.00	88.61	88.97	89.45
segment	90.43	86.75	92.21	94.03	94.46	90.87	94.11	94.81	92.86	91.00	92.99	93.20
spambase	92.52	91.98	91.83	93.13	93.02	90.57	92.74	93.09	92.96	92.39	93.07	93.24
splice	93.98	94.26	93.70	95.02	94.23	86.36	94.26	94.51	77.02	77.81	77.18	77.68
texture	86.11	85.44	86.11	89.73	86.91	85.35	87.40	89.04	96.22	95.31	96.27	96.45
thyroid	99.15	98.14	99.15	99.19	99.38	97.42	99.25	99.36	98.58	98.93	98.42	98.63
tic-tac-toe	96.97	97.28	97.70	97.60	84.44	84.44	84.54	86.63	98.85	98.02	98.54	98.75
twonorm	84.31	84.05	85.01	89.04	79.74	78.86	80.18	84.64	93.73	93.93	93.78	94.81
vehicle	61.24	60.07	61.36	64.07	67.37	67.38	67.73	68.44	64.46	69.04	67.98	68.46
wisconsin	95.13	93.41	94.84	95.41	95.57	92.42	96.00	96.28	96.28	96.28	96.56	96.42
wine	61.24	60.07	61.36	64.07	93.30	89.35	88.24	91.60	96.05	95.49	96.05	96.05
yeast	74.20	74.53	75.07	74.20	76.15	76.28	75.54	76.21	74.60	74.74	75.48	74.80
zoo	87.09	85.18	87.09	87.09	92.18	79.27	92.18	92.18	95.00	80.18	95.00	95.00

Table 4
Classification accuracy (labeled ratio 20%)

Dataset	Self-Train (JRip)	Co-Train (JRip)	Tri-Train (JRip)	CST-Voting (JRip)	Self-Train (C4.5)	Co-Train (C4.5)	Tri-Train (C4.5)	CST-Voting (C4.5)	Self-Train (3NN)	Co-Train (3NN)	Tri-Train (3NN)	CST-Voting (3NN)
automobile	73.58	72.83	68.50	76.67	78.04	68.71	73.71	74.96	72.29	64.04	75.37	72.29
banana	72.94	74.55	73.36	74.19	74.75	81.74	74.26	74.75	74.72	76.66	74.66	75.38
breast	72.38	71.71	68.94	72.03	74.84	73.09	68.23	74.51	74.82	75.50	74.48	75.50
bupa	57.11	57.11	59.17	57.11	56.25	56.25	56.25	56.25	57.11	57.11	59.17	57.11
cleveland	53.78	54.12	54.78	54.12	52.76	53.76	57.37	54.10	57.10	56.43	57.73	56.43
coil2000	94.03	94.01	94.02	94.03	94.03	94.03	94.03	94.03	93.61	93.61	93.73	93.61
contraceptive	46.98	47.25	45.29	46.58	49.62	51.94	52.35	52.00	51.86	51.59	52.00	52.95
crx	86.99	85.01	87.60	86.99	85.89	85.44	86.37	86.06	85.31	85.15	85.29	85.46
dermatology	89.04	85.51	89.87	90.97	93.72	89.62	92.91	93.99	96.46	95.64	95.91	96.19
german	71.90	70.60	67.10	70.00	71.90	72.80	73.50	72.90	72.50	72.50	72.80	72.80
glass	68.07	66.28	63.38	68.12	62.60	59.00	69.18	68.74	67.03	69.00	68.57	70.87
haberman	71.92	72.25	72.91	72.25	72.87	72.87	73.17	72.87	73.55	73.55	73.20	73.55
heart	79.52	76.87	76.55	79.53	78.46	77.22	76.51	77.51	81.86	79.54	81.19	81.53
hepatitis	79.25	82.58	81.21	81.92	82.54	80.04	81.92	81.33	88.33	87.67	82.58	87.04
housevotes	97.03	97.03	97.03	97.03	96.12	96.99	95.69	96.12	91.87	92.28	91.87	91.87
iris	94.00	94.00	94.67	94.67	93.33	95.33	94.00	93.33	94.00	93.33	92.00	93.33
led7digit	65.20	66.20	70.80	70.00	67.40	61.20	71.00	68.60	72.00	73.20	73.20	73.00
magic	82.19	81.70	82.61	82.83	84.02	82.48	83.91	84.24	81.97	80.67	81.40	82.17
mammographic	83.73	82.29	82.65	83.73	84.82	82.65	83.01	84.34	82.77	82.41	83.13	83.01
monk2	67.14	67.14	67.14	67.14	67.14	67.14	67.14	67.14	67.14	67.14	67.14	67.14
movement	24.17	49.44	51.94	45.83	47.22	54.72	54.17	55.28	51.67	47.22	63.06	55.00
mushroom	100.0	100.0	100.0	100.0	100.0	99.89	99.96	100.0	100.0	100.0	100.00	100.0
page-blocks	96.11	96.14	96.03	96.40	96.60	95.32	96.75	96.80	96.27	95.56	96.03	96.20
pendigits	91.26	90.61	90.91	93.26	87.96	86.24	87.62	88.76	96.41	94.04	96.13	96.32
phoneme	81.01	79.68	81.55	81.46	81.48	80.14	81.09	81.37	82.48	82.12	82.70	83.33
pima	76.56	73.44	73.19	75.26	76.96	74.48	75.13	75.91	74.35	75.27	72.00	74.35
ring	91.86	92.34	92.55	93.27	80.76	80.09	81.38	85.05	62.05	60.42	62.91	61.05
satimage	82.98	83.96	83.48	85.63	84.26	84.60	84.18	85.66	88.81	88.30	88.81	89.09
segment	89.74	88.83	91.86	93.16	93.90	92.12	93.98	94.20	92.64	91.00	92.38	92.99
spambase	92.07	91.76	92.18	93.11	92.72	91.46	92.50	93.24	93.00	92.52	93.15	93.28
splice	94.01	93.67	92.48	94.64	94.17	91.29	93.82	94.23	77.12	77.55	75.99	77.15
texture	86.80	84.56	86.25	90.13	86.64	86.65	86.65	89.33	96.11	95.36	95.98	96.47
thyroid	99.22	97.43	99.19	99.22	99.29	98.99	99.36	99.33	98.42	99.10	98.33	98.61
tic-tac-toe	96.97	96.76	97.91	97.29	84.44	81.94	83.29	85.59	98.85	97.60	98.01	98.75
twonorm	83.81	85.24	84.26	89.16	79.78	79.76	79.49	85.22	93.66	93.82	94.15	94.92
vehicle	61.71	65.73	62.88	64.77	68.68	68.57	66.79	69.04	68.46	67.97	67.98	68.57
wisconsin	95.27	95.27	95.70	95.70	95.13	94.28	95.85	96.14	96.28	96.42	96.42	96.42
wine	61.71	65.73	62.88	64.77	91.60	87.09	93.30	92.71	96.05	96.05	96.60	96.60
yeast	74.94	75.01	74.81	74.87	75.40	75.88	74.26	75.14	74.53	74.67	74.94	74.60
zoo	85.09	83.09	87.09	85.09	90.18	82.36	92.18	91.18	92.09	80.27	95.00	93.09

Table 5
Classification accuracy (labeled ratio 30%)

Dataset	Self-Train (JRip)	Co-Train (JRip)	Tri-Train (JRip)	CST-Voting (JRip)	Self-Train (C4.5)	Co-Train (C4.5)	Tri-Train (C4.5)	CST-Voting (C4.5)	Self-Train (3NN)	Co-Train (3NN)	Tri-Train (3NN)	CST-Voting (3NN)
automobile	72.21	74.21	69.79	76.04	76.75	72.46	63.54	75.54	67.25	68.50	62.25	69.79
banana	73.08	78.98	73.19	74.25	74.77	79.09	74.98	74.92	74.81	79.66	74.57	75.75
breast	73.79	75.22	70.70	74.86	74.51	72.39	74.85	74.51	74.82	74.82	73.40	75.17
bupa	56.54	57.11	56.81	56.54	56.25	56.25	57.13	56.25	56.54	57.11	56.81	56.54
cleveland	53.78	53.45	53.78	53.78	54.10	54.43	54.40	54.09	57.10	58.06	59.68	58.40
coil2000	94.04	94.01	94.01	94.03	94.03	94.03	94.03	94.03	93.61	93.61	93.69	93.61
contraceptive	46.10	46.98	44.80	46.64	50.17	49.77	50.45	51.19	50.31	50.78	51.59	51.25
crx	87.91	86.38	87.14	86.84	85.44	85.44	85.76	86.66	86.08	84.85	85.61	86.23
dermatology	88.27	86.04	85.77	91.53	93.20	92.91	92.36	94.55	96.19	96.18	96.46	96.19
german	71.30	70.60	68.10	71.30	72.20	70.90	70.60	71.70	72.20	71.60	72.70	72.10
glass	64.96	63.87	59.76	63.48	68.74	67.32	58.90	66.36	71.34	69.94	64.31	70.39
haberman	72.57	71.59	70.29	72.25	72.87	73.52	73.53	72.87	71.90	73.55	71.26	73.55
heart	81.51	76.89	78.56	81.86	78.46	75.22	77.54	80.14	81.52	81.54	82.19	81.85
hepatitis	81.92	82.50	81.29	83.83	81.92	81.25	83.79	82.50	86.38	85.13	87.13	86.38
housevotes	96.59	97.03	97.03	97.03	96.12	96.12	94.38	96.12	91.87	91.43	91.00	91.43
iris	93.33	96.00	92.00	94.00	94.00	94.00	94.67	95.33	94.00	96.00	92.67	94.00
led7digit	69.00	71.20	69.00	70.80	72.00	67.20	71.60	71.60	72.40	72.40	73.40	73.00
magic	82.42	81.28	82.63	82.88	84.03	83.05	83.69	84.48	82.03	81.24	81.27	82.38
mammographic	84.22	83.61	83.37	84.22	83.61	83.13	83.13	84.58	82.17	82.53	83.86	83.37
monk2	67.14	67.14	67.14	67.14	67.14	67.14	67.14	67.14	67.14	67.14	67.14	67.14
movement	41.11	51.39	51.11	52.78	50.56	55.00	52.78	56.39	61.39	52.50	57.22	58.89
mushroom	100.0	99.98	100.0	100.0	100.0	99.93	100.0	100.0	100.0	100.0	100.0	100.0
page-blocks	96.13	95.82	96.13	96.36	96.38	95.82	96.16	96.40	96.29	96.18	95.98	96.25
pendigits	91.47	90.79	90.65	93.27	88.55	86.83	87.40	89.09	96.39	94.91	95.90	96.30
phoneme	80.94	79.83	80.18	80.96	81.85	80.61	80.94	81.61	82.66	81.18	81.27	81.98
pima	76.04	71.10	75.78	75.78	75.53	75.92	73.56	75.78	73.70	73.83	74.36	75.91
ring	92.42	91.84	92.42	93.47	80.34	80.80	82.16	86.01	62.57	60.99	61.70	61.20
satimage	83.67	83.62	83.20	85.35	84.13	84.27	83.92	85.16	88.75	88.81	88.52	89.03
segment	90.91	89.18	91.17	93.81	93.85	93.16	91.90	94.59	92.94	92.34	91.69	93.25
spambase	92.09	91.72	91.81	92.89	92.83	91.98	91.83	93.24	92.91	92.70	92.85	93.07
splice	94.01	94.29	92.57	94.51	94.11	92.57	93.32	94.45	76.99	77.15	74.01	76.87
texture	87.20	86.45	85.84	90.27	87.69	86.85	86.56	89.38	96.18	95.53	95.73	96.20
thyroid	99.24	99.17	99.15	99.25	99.25	99.11	99.15	99.29	98.56	98.65	98.35	98.72
tic-tac-toe	97.70	97.08	97.81	97.91	84.75	83.51	82.88	87.15	98.85	96.87	96.45	98.33
twonorm	84.58	84.57	85.51	89.41	79.28	79.46	80.34	85.76	93.51	93.92	94.24	94.81
vehicle	64.90	65.72	62.41	65.37	70.69	67.96	68.44	69.97	68.58	68.45	68.33	69.40
wisconsin	95.57	92.70	95.70	95.13	95.57	95.28	95.85	96.71	96.28	96.57	96.13	96.28
wine	62.18	65.73	62.88	65.47	94.41	89.28	90.95	96.60	96.05	95.49	95.49	96.05
yeast	74.53	74.54	74.06	74.60	75.27	74.39	74.93	75.47	74.80	74.94	74.80	75.21
zoo	86.09	83.27	85.09	86.09	91.18	87.27	94.09	91.18	93.18	89.09	92.09	93.09

Table 6

Total wins of each algorithm

	10%			20%			30%		
Algorithm	JRip	J48	3NN	JRip	J48	3NN	JRip	J48	3NN
Self-Training	3	4	2	3	7	7	5	4	9
Co-Training	4	4	5	6	4	4	8	8	4
Tri-Training	8	7	**13**	10	9	9	1	2	9
CST-Voting	**19**	**17**	12	**13**	**16**	**11**	**19**	**19**	**13**

The number of wins of each one of the tested methods according to the supervised classifier used as base learner and the ratio of labeled data in the training set is presented in Table 6, while the best scores are highlighted in bold.It should me mentioned that draw cases between algorithms have not been encountered. The above aggregated results show that CST-Voting is by far the most effective method in all cases except the one using 3NN as base learner with a labeled ratio of 10%. In this case, Tri-training(3NN) performs better in 13 datasets, followed by CST-Voting (12 wins), Co-training (5 wins) and Self-training (2 wins). In more detail:

- Depending upon the base classifier, CST-Voting (JRip) scores the best accuracy value in 19, 13 and 19 datasets (a total of 51 out of 99 datasets), using a labeled ratio of 10%, 20% and 30% respectively. CST-Voting(J48) performs better in 17, 16 and 19 datasets (a total of 52 out of 101 datasets), while CST-Voting(3NN) prevails in 12, 11 and 13 datasets (a total of 36 out of 98 datasets) respectively. So, CST-Voting performs better using JRip or J48 as base learners.
- Regarding the ratio of labeled instances in the training set, CST-Voting performs better in 48 out of 98 datasets for 10% labeled ratio, in 40 out of 99 datasets for 20% labeled ratio and in 51 out of 101 datasets for 30% labeled ratio. It is clear that CST-Voting achieves better results for 10% and 30% labeled ratio.

Additionally, a more representative visualization of the classification performance of the compared SSC methods is presented in Figures 1, 2 and 3. Each figure displays a radar chart illustrating the accuracy measure of each tested algorithm according to the supervised classifier used as base learner and the labeled ratio.

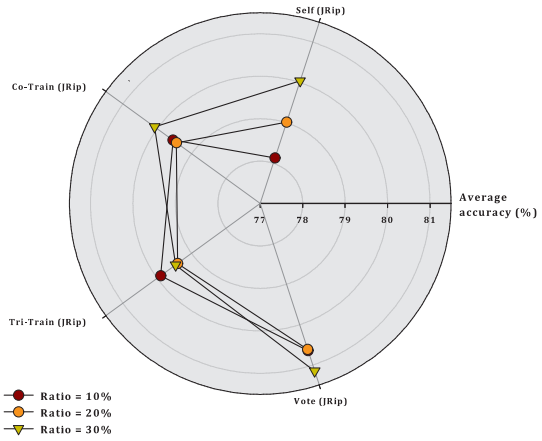

Fig. 1. Comparison of algorithms (JRip base classifier)

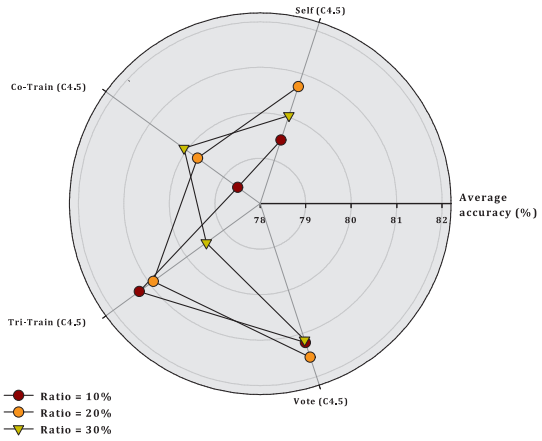

Fig. 2. Comparison of SSL algorithms (J48 base classifier)

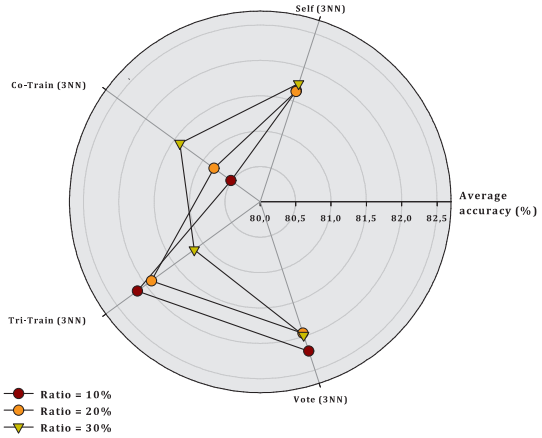

Fig. 3. Comparison of SSL algorithms (3NN base classifier)

To evaluate the performance of the tested algorithms, we performed multiple comparisons of the accuracy results between the proposed method and the set of SSC algorithms used in our study. Therefore, the Friedman Aligned Ranks nonparametric test [19] was applied first and after the Finner post hoc test [11] with a significance level $\alpha = 0.05$. Friedman Aligned Ranks test is considered to be one of the most well-known tools for multiple statistical comparison tests when comparing more than two methods [13]. According to the computed Friedman test results (Tables 7, 8 and 9) the algorithms are ordered from the best performer (lowest ranking value) to the lowest one (highest ranking value). It is observed that CST-Voting prevails in all 9 cases (3 different classifiers and 3 different labeled ratios) indicating its efficiency.

Since the null hypothesis of equivalence of medians of algorithms is rejected, the Finner post hoc statistical procedure is applied to detect the specific differences among the algorithms. Finner test is easy to comprehend and usually offers better results than other tests, such as Holm test [20] or Hochberg test [18], especially when the number of algorithms to be compared is low [13]. The complete post hoc results are also presented in Tables 7, 8 and 9 using CST-Voting as control method. The proposed ensemble method takes precedence over the rest SSL algorithms, since it gives statistically better results in all cases except the one using 3NN as base learner with a labeled ratio of 10%, confirming that the proposed ensemble method obtains higher classification accuracy than its constituent parts.

Table 7
Friedman Aligned Ranks test-Finner post hoc test ($\alpha = 0.05$)

Ratio	Classifier (JRip)	Friedman Ranking	Finner post-hoc test p-value	Null Hypothesis
10%	CST-Voting	45.7125		
	Self-training	80.1875	0.00088	rejected
	Tri-training	97.3750	0.00000	rejected
	Co-training	98.7250	0.00000	rejected
20%	CST-Voting	46.1375		
	Self-training	87.7875	0.00006	rejected
	Tri-training	90.2000	0.00003	rejected
	Co-training	97.8750	0.00000	rejected
30%	CST-Voting	43.9250		
	Tri-training	81.3375	0.00000	rejected
	Co-training	87.6125	0.00000	rejected
	Self-training	109.1250	0.00088	rejected

Table 8
Friedman Aligned Ranks test-Finner post hoc test ($\alpha = 0.05$)

Ratio	Classifier (J48)	Friedman Ranking	Finner post-hoc test p-value	Null Hypothesis
10%	CST-Voting	49.6500		
	Self-training	70.8500	0.04073	rejected
	Tri-training	84.2250	0.00169	rejected
	Co-training	117.2750	0.00000	rejected
20%	CST-Voting	53.4250		
	Self-training	79.5000	0.02273	rejected
	Tri-training	79.6500	0.02273	rejected
	Co-training	109.4250	0.00000	rejected
30%	CST-Voting	43.9250		
	Tri-training	81.3375	0.00030	rejected
	Co-training	87.6125	0.00005	rejected
	Self-training	109.1250	0.00000	rejected

Table 9
Friedman Aligned Ranks test-Finner post hoc test ($\alpha = 0.05$)

Ratio	Classifier (3NN)	Friedman Ranking	Finner post-hoc test p-value	Null Hypothesis
10%	CST-Voting	58.6500		
	Self-training	67.0500	0.41748	accepted
	Tri-training	82.8750	0.03875	rejected
	Co-training	113.4250	0.00000	rejected
20%	CST-Voting	56.3875		
	Tri-training	79.1000	0.02836	rejected
	Self-training	82.3375	0.02451	rejected
	Co-training	104.1750	0.00010	rejected
30%	CST-Voting	54.1625		
	Tri-training	75.5375	0.03909	rejected
	Co-training	95.1625	0.00015	rejected
	Self-training	97.1375	0.00010	rejected

5. Conclusions

In the present study, a new SSL ensemble algorithm, called CST-Voting, is proposed. Semi-supervised learning is an emerging subfield of machine learning, with a view to building efficient classifiers exploiting a limited pool of labeled data together with a large pool of unlabeled ones. CST-Voting combines three familiar SSL algorithms: Co-training, Self-training and Tri-training. The efficiency of the proposed algorithm was evaluated on a number of benchmark datasets in terms of classification accuracy using JRip, C4.5 and 3NN supervised classifiers as base learners and different ratio of labeled data. A plethora of experiments were carried out indicating the effectiveness of the proposed

ensemble, as confirmed statistically by the Friedman Aligned Ranks nonparametric test and the Finner post hoc test.

CST-Voting outperforms its component algorithms, which use the same base classifiers, confirming the effectiveness of ensemble methods. Co-training, Self-training and Tri-training exploit unlabeled examples through different mechanisms and thereby ensure the ensemble diversity. Therefore, combining the ensemble methodology and SSC algorithms seems to lead to more efficient, stable and robust predictive models.

Since the experiments results are quite encouraging, a next step should be the usage of other supervised classifiers as base learners, such as support vector machines [6,29] and neural networks [3,17]. Moreover, an interesting aspect is the implementation of the method in specific scientific fields applying real world datasets, such as the educational field.

Another interesting aspect for future work is a parallel implementation of the CST-Voting method. Recently, a distributed SSL method with kernel ridge regression has been effectively applied to data subsets that are distributively stored on multiple servers [7]. Implementing each one of the components SSL algorithms in parallel machines is a very important aspect to be studied, as huge amount of data can be processed in significantly less time.

Appendix

A java software tool implementing the proposed ensemble method can be found in `http://www.math.upatras.gr/~livieris/Software/CST_Voting.zip`

References

[1] D. Aha. *Lazy Learning*. Dordrecht: Kluwer Academic Publishers, 1997.

[2] 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 & Soft Computing*, 17, 2011.

[3] C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford, 1995.

[4] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *11th annual conference on Computational learning theory*, pages 92–100. ACM, 1998.

[5] L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[6] C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[7] X. Chang, S.B Lin, and D.X. Zhou. Distributed semi-supervised learning with kernel ridge regression. *J. Mach. Learn. Res*, 2017.

[8] W. Cohen. Fast effective rule induction. In *International Conference on Machine Learning*, pages 115–123, 1995.

[9] T.G. Dietterich. Ensemble methods in machine learning. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems*, volume 1857, pages 1–15. Springer Berlin Heidelberg, 2001.

[10] A. Dong, F. Chung, and Sh. Wang. Semi-supervised classification method through oversampling and common hidden space. *Information Sciences*, 349:216–228, 2016.

[11] H. Finner. On a monotonicity problem in step-down multiple test procedures. *Journal of the American Statistical Association*, 88(423):920–923, 1993.

[12] Y. Gao, J. Ma, and A.L. Yuille. Semi-supervised sparse representation based classification for face recognition with insufficient labeled samples. *IEEE Transactions on Image Processing*, 26(5):2545–2560, 2017.

[13] S. García, A. Fernández, J. Luengo, and F. Herrera. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, 180(10):2044–2064, 2010.

[14] A.B. Goldberg and X. Zhu. *New directions in semi-supervised learning*. PhD thesis, University of Wisconsin–Madison, 2010.

[15] M.F.A. Hady and F. Schwenker. Combining committee-based semi-supervised learning and active learning. *Journal of Computer Science and Technology*, 25(4):681–698, 2010.

[16] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten. The WEKA data mining software: n update. *SIGKDD Explorations Newsletters*, 11:10–18, 2009.

[17] S. Haykin. *Neural Networks: A comprehensive foundation*. Macmillan College Publishing Company, New York, 1994.

[18] Y. Hochberg. A sharper bonferroni procedure for multiple tests of significance. *Biometrika*, 75(4):800–802, 1988.

[19] J.L. Hodges and E.L. Lehmann. Rank methods for combination of independent experiments in analysis of variance. *The Annals of Mathematical Statistics*, 33(2):482–497, 1962.

[20] S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70, 1979.

[21] S. Kotsiantis, K. Patriarcheas, and M. Xenos. A combinational incremental ensemble of classifiers as a technique for predicting students performance in distance education. *Knowledge-Based Systems*, 23(6):529–535, 2010.

[22] 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*, 37(6):1088–1098, 2007.

[23] M. Lichman. UCI machine learning repository, 2013.

[24] Y. Ma, W. Pan, S. Zhu, H. Yin, and J. Luo. An improved semi-supervised learning method for software defect prediction. *Journal of Intelligent & Fuzzy Systems*, 27(5):2473–2480, 2014.

[25] C.J. Merz. Using correspondence analysis to combine classifiers. *Machine Learning*, 36:33–58, 1999.

[26] A. Mey and M. Loog. A soft-labeled self-training approach. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 2604–2609. IEEE, 2016.

[27] V. Ng and C. Cardie. Weakly supervised natural language learning without redundant views. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 94–101. Association for Computational Linguistics, 2003.

[28] K. Nigam and R. Ghani. Analyzing the effectiveness and applicability of co-training. In *Proceedings of the ninth international conference on Information and knowledge management*, pages 86–93. ACM, 2000.

[29] J.C. Platt. *Advances in Kernel Methods - Support Vector Learning*. MIT Press, Cambridge, Massachusetts, 1998.

[30] J.R. Quinlan. *C4.5: Programs for machine learning*. Morgan Kaufmann, San Francisco, 1993.

[31] M. Sokolova, N. Japkowicz, and S. Szpakowicz. Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In *Australian conference on artificial intelligence*, volume 4304, pages 1015–1021, 2006.

[32] L. Todorovski and S. Džeroski. Combining classifiers with meta decision trees. *Machine Learning*, 50(3):223–249, 2002.

[33] I. Triguero, S. García, and F. Herrera. Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. *Knowledge and Information Systems*, 42(2):245–284, 2015.

[34] I.H. Witten, E. Frank, M.A. Hall, and C.J. Pal. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[35] X. Wu, V. Kumar, J.R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G.J. McLachlan, A.F.M. Ng, B. Liu, P.S. Yu, Z.H. Zhou, M. Steinbach, D.J. Hand, and D. Steinberg. Top 10 algorithms in data mining. *Knowledge and information systems*, 14(1):1–37, 2008.

[36] Y. Xu, Y. Zhang, Z. Yang, X. Pan, and G. Li. Imbalanced and semi-supervised classification for prognosis of ACLF. *Journal of Intelligent & Fuzzy Systems*, 28(2):737–745, 2015.

[37] D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd Annual Meeting on Association for Computational Linguistics*, pages 189–196. Association for Computational Linguistics, 1995.

[38] Z. Yu, Y. Lu, J. Zhang, J. You, H.S. Wong, Y. Wang, and G. Han. Progressive semisupervised learning of multiple classifiers. *IEEE transactions on cybernetics*, 2017.

[39] Yan Zhou and Sally Goldman. Democratic co-learning. In *Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on*, pages 594–602. IEEE, 2004.

[40] Z.H. Zhou. When semi-supervised learning meets ensemble learning. *Frontiers of Electrical and Electronic Engineering in China*, 6(1):6–16, 2011.

[41] Z.H. Zhou and M. Li. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on knowledge and Data Engineering*, 17(11):1529–1541, 2005.

[42] X. Zhu and A.B. Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009.