Neurocomputing 249 (2017) 277–289

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Multi-view based unlabeled data selection using feature transformation methods for semiboost learning☆

Thanh-Binh Le, Sugwon Hong, Sang-Woon Kim*

Department of Computer Engineering, Myongji University, Yongin 17058, Republic of Korea

ARTICLE INFO

Article history:
Received 29 February 2016
Revised 26 January 2017
Accepted 5 April 2017
Available online 12 April 2017

Communicated by Q Wei.

Keywords:
SemiBoost learning
Useful unlabeled data selection
Multiple views of feature set
Feature decomposition methods

ABSTRACT

SemiBoost Mallapragada et al. (2009) is a boosting framework for semi-supervised learning, in which unlabeled data as well as labeled data both contribute to learning. Various strategies have been proposed in the literature to perform the task of selecting useful unlabeled data in SemiBoost. Recently, a multi-view based strategy was proposed in Le and Kim (2016), in which the feature set of the data is decomposed into subsets (i.e., multiple views) using a feature-decomposition method. In the decomposition process, the strategy inevitably results in some loss of information. To avoid this drawback, this paper considered feature-transformation methods, rather than using the decomposition method, to obtain the multiple views. More specifically, in the feature-transformation method, a number of views were obtained from the *entire* feature set using the same number of different mapping functions. After deriving the number of views of the data, each of the views was used for measuring corresponding confidences, for first evaluating examples to be selected. Then, all the confidence levels measured from the multiple views were combined as a weighted average for deriving a target confidence. The experimental results, which were obtained using support vector machines for well-known benchmark data, demonstrate that the proposed mechanism can compensate for the shortcomings of the traditional strategies. In addition, the results demonstrate that when the data is transformed appropriately into multiple views, the strategy can achieve further improvement in results in terms of classification accuracy.

1. Introduction

In semi-supervised learning (SSL) approaches [5,32,43], both a limited number of labeled data (L) and a multitude of unlabeled data (U) are used to learn a classification model. Specifically, in SemiBoost [23], which is a boosting framework for SSL, confidence levels of all U examples are first computed based on the predictions made by an ensemble classifier initialized with L only, and the similarity among the examples of $L \cup U$. Then, a few examples with higher confidence levels are selected to retrain the ensemble classifier together with L. However, it is not guaranteed that adding the selected data to the training data will lead to a situation in which the classification performance can be improved [35]. Therefore, various approaches have been proposed in the literature for selecting a small amount of useful unlabeled data (U_s) from U: these include the self-training [25,30,40] and co-training [3,12,18] approaches, confidence-based approaches [19,20,23], density/distance-based approaches [8,27,28], and other approaches used in active learning (AL) algorithms [7,11,33].

Meanwhile, in many applications, the objects may be represented naturally from multiple viewpoints or even observed from different sensors. These observation patterns have led to the development of various multi-view learning algorithms. More specifically, in [39], a number of representative multi-view learning algorithms in different areas were reviewed and categorized into three groups: co-training [3], multiple kernel learning [2], and subspace learning [15]. In the multi-view learning strategy, different classifiers can be trained on different views; the goal of multi-view learning is to combine classifiers for each view in order to improve overall performance and exceed that of classifiers which are trained on each view separately. The first successful multi-view learning technique is co-training [3]. Other notable multi-view techniques are co-regularization [29], techniques relying on (kernel) canonical correlation analysis [26,31], multi-view Fisher discriminant analysis [9,10], generalized multiview analysis [34], spectral embedding [38], etc.

☆ The work of the first and the second authors was supported by the Human Resources Program in Energy Technology of the Korea Institute of Energy Technology Evaluation and Planning (KETEP), granted financial resource from the Ministry of Trade, Industry & Energy, Republic of Korea (No. 20154030200770). This work was done as a follow-up study of [20].

* Corresponding author.
E-mail address: kimsw@mju.ac.kr (S.-W. Kim).

http://dx.doi.org/10.1016/j.neucom.2017.04.021
0925-2312/

Motivated by this, in [20], in order to select useful U_s efficiently, the confidence levels of all $x \in U$ examples were measured using multiple views extracted from the feature set of the data. In the multi-view based strategy, a small amount of useful U_s are selected as follows: first, multiple views of the data are derived independently through a decomposition method of the feature set of the data; second, each of the views is used to measure the confidence levels of the data; third, all the confidence levels measured from the views are used as a weighted average to derive a target confidence. The central idea of this strategy is that confidence, which cannot be measured from the whole feature set itself, might be obtained from other views extracted from the set. As a result, the performance of the selection strategy strongly depends on the method with which the feature vector is decomposed into multiple views.

From this perspective, it is interesting to consider how in the multi-view based strategy the original set of features can be decomposed into several subsets. Recently, in [39], feature decomposition methods were categorized into three approaches. These included methods of constructing multiple views from meta data through random approaches, as in randomly selecting feature subsets to train different classifiers [16]; methods of reshaping or decomposing the original single-view feature set into multiple views, such as genetic algorithm (GA) based methods [36]; and methods of performing feature set partitioning automatically, such as PMD (pseudo multi-view decomposition) [6]. In addition to the above categories, other methods, including FESCOT (feature selection for co-training) [22], TSFS (two-view sub;space feature splitting using a principal component analysis) [41], and MaxInd (maximizing the independence between two feature subsets) [14], have been proposed in the literature.

In general, generating different views corresponds to feature set partitioning, which generalizes the task of traditional feature selection. However, it is also well known that feature selection/feature extraction inevitably results in some loss of information. In order to address this point, data transformation techniques, including principle component analysis and the Fourier transform technique, for example, have been considered for extracting multiple views of the data. In that case, the orthonormal eigenvectors and the Fourier coefficients are used together, and then both features can help each other when measuring confidence. That is, the confident values that have been inappropriately measured from the latter feature can be verified by referring to the values obtained from the former.

The significant point is that multiple transformation techniques can be used to derive multiple views with minimal loss of information, and, in turn, select U_s from U successfully, instead of using decomposition-based methods. Motivated by this, in this study, an approach for transforming the whole feature set into multiple views using different mapping functions was implemented and empirically compared. More specifically, a strategy of selecting useful U_s based on a multi-view setting was implemented, in which a number of distinct views were achieved using the same number of mapping functions.

The main goal of this paper is to demonstrate that using a transformation-based selection (TBS) method when selecting U_s from U, instead of the data decomposition-based selection (DBS) method (which has been developed in [20]), leads to an improvement in terms of classification accuracies. In DBS, the feature set of the data is independently decomposed into multiple views. As a result, implementing that strategy inevitably leads to losing information for discrimination. Furthermore, with regard to decomposing the feature set, there is no specific way to decide which one is better than the other. Meanwhile, unlike DBS, TBS can generate distinct views by differently mapping the input-feature space to the other subspaces. In particular, since the mapping can be performed with the entire feature set, the data can be projected into another space with minimal loss of information.

In summary, a theoretical comparison of the two methods described above can be made as follows. In order to obtain the two views from single-view sources when selecting U_s from U, TBS as well as DBS can be considered. In TBS, the two views are obtained using two different mapping functions, while, in DBS, they are achieved using feature-splitting methods. However, competitive DBSs are more expensive to calculate and need to be operated with labeled data L, i.e., in supervised modes. Moreover, the method suffers from information loss that occurs in the process of splitting feature sets. In order to avoid these drawbacks, in TBS, using two feature extractors for the single-view source data (e.g., PCA (principal component analysis) and SNE (t-distributed stochastic neighbor embedding) [37]), two different views are extracted first, and then applied to the SemiBoost type learning algorithm. That is, in order to obtain the multiple views with minimal loss of information and, more importantly, in unsupervised mode (i.e., U as well as L can be utilized), TBS can be considered, and then, a transformation-based SemiBoost type classifier can be designed.

The main contribution of this paper is to demonstrate that the classification accuracy of the supervised / semi-supervised classifiers can be improved using the multi-view based selection strategy. In particular, it is demonstrated that when multiple views are derived using the data transformation techniques, classification accuracy can be further improved.

The remainder of the paper is organized as follows: in Section 2, a brief introduction to the selection criteria used in SemiBoost and its modified versions is provided; in Section 3, proposed methods of obtaining multiple views using feature-transformation methods and learning an ensemble classifier are presented continuously; in Section 4, illustrative examples for adjusting the experimental parameters, such as mapping combinations and selection criteria, and the experimental results obtained using the benchmark data are presented; in Section 5, the concluding remarks and limitations that deserve further study are presented.

2. Related work

In this section, the SemiBoost [23] algorithm and the related criteria, which are closely related to the present paper, are briefly overviewed in order to make it complete. The detailed description can also be found in the literature [19,20,23].

2.1. SemiBoost and its variants

The goal of SemiBoost is to iteratively improve the performance of a supervised learning algorithm (named $\mathcal{A}$) by using $U = \{(x_i)\}_{i=1}^{n_u}$ in conjunction with $L = \{(x_i, y_i)\}_{i=1}^{n_l}$, where $x_i \in \mathbb{R}^d$ and $y_i \in \{+1, -1\}$. At each iteration, the top few most confident examples are selected from U and then provided for the learning of a base classifier through the algorithm $\mathcal{A}$. After repeating the select-and-train process, the base classifiers obtained from each iteration are combined linearly into an ensemble classifier.

In order to exclude noninformative samples from this boosting as well as to provide highly reliable pseudo-labeled samples to the classifier, all $x_i \in U$ are evaluated using a criterion of $|p_i - q_i|$. Here, $p_i(\equiv p(x_i))$ and $q_i(\equiv q(x_i))$, which are measured using the pair-wise similarity function, $S(j,k) = e^{-\|x_j - x_k\|_2^2/\sigma^2}$, for all x_j and x_k of the training set, denote the confidence in classifying x_i into a positive class and negative class, respectively.[1] Also, σ is the scale parameter controlling the spread of the function.

[1] Details of p_i and q_i are omitted here in the interest of compactness, but can be found in the literature [19,20,23].

When computing $|p_i - q_i|$, all $x_i \in L$ examples in class $\{+1\}$ and class $\{-1\}$ can be divided into $L^+ = \{(x_i, y_i)|y_i = 1\}_{i=1}^{n_l^+}$ and $L^- = \{(x_i, y_i)|y_i = -1\}_{i=1}^{n_l^-}$, where $n_l^+ + n_l^- = n_l$, and then the computation of the criterion, $\rho_1(x_i) = |p_i - q_i|$, can be re-formulated as follows:

$$\rho_1(x_i) = |X_i^+ - X_i^- + X_i^u|, \tag{1}$$

where $X_i^+ = e^{-2H_i} \sum_{x_j \in L^+} S_{i,j}^{ul}$, $X_i^- = e^{2H_i} \sum_{x_j \in L^-} S_{i,j}^{ul}$, and $X_i^u = \frac{C}{2} \sum_{x_j \in U} S_{i,j}^{uu} (e^{H_j - H_i} - e^{H_i - H_j})$. Here, $H_i (\equiv H(x_i))$ denotes the ensemble classifier; S^{ul} (and S^{uu}) the $n_u \times n_l$ (and $n_u \times n_u$) submatrix of S.

From (1), the value of $X_i^+ - X_i^-$ can be considered as the relative measurement for estimating the possibility that x_i belongs to $\{+1\}$ or $\{-1\}$ class. From this consideration, it can be seen that if the value is nearly zero, then x_i could remain on the boundary of the classifier. In order to address this problem, SemiBoost uses X_i^u in (1) to provide more meaningful information for enlarging the margin.

However, using more data is not always beneficial. If the value obtained using X_i^u is very large or X_i^+ is nearly equal to X_i^-, (1) will generate some erroneous data. In this case, the confidence estimation for x_i will depend on the U examples. In order to avoid this, in [19], the criterion is modified by taking a balance among the three terms in (1). This modification can be achieved through balancing the three terms through a reduction in the impact of the third term, especially when $X_i^+ \approx X_i^-$. This idea is motivated from the rule of mapping the selected unlabeled example ($x_i \in U_s$) to a predicted label ($\hat{y}_i$) being viewed as a procedure for obtaining the estimates of a set of conditional probabilities. Using the probability estimates as a penalty cost, the criterion of (1), $\rho_1(x_i)$, can be modified as follows:

$$\rho_2(x_i) = \left|X_i^+ - X_i^- + X_i^u - (1 - p_E(x_i))\right|, \tag{2}$$

where $p_E(x_i)$ denotes the class posterior probability of an instance of x_i (i.e. a *certainty level*) and $1 - p_E(x_i)$ corresponds to the percentage of mistakes when labeling x_i.

2.2. Multi-view based criterion

In multi-view learning strategy [39], for example, in co-training [3], two classifiers, h_1 and h_2, are trained using L_1 and L_2, two views of L, respectively: h_1 is based on L_1, while h_2 is based on L_2. The two classifiers are then evaluated using U. After evaluating the examples of U using h_1 and h_2 separately, a subset of useful unlabeled examples (U_s), which are the most confident, are selected and then added to L for the next iteration, i.e. $\{(x_i, h_1(x_i))\}$ to L_2 and $\{(x_i, h_2(x_i))\}$ to L_1. Both h_1 and h_2 are now re-trained on the expanded L_1 and L_2 sets, and the procedure is repeated until some stopping criterion is met.

Motivated by this, a multi-view based selection strategy is proposed recently in [20], in which the selection is performed as follows: after dividing the feature set into multiple subsets (i.e., views), confidence levels are evaluated first; then, all the confidence levels measured from the multiple views are combined as a weighted average to derive the target confidence. Here, in order to decompose the feature set, various methods are considered, including the random splitting method (RD), genetic algorithm based method (GA), traditional feature selection based method (FS), mutual information based decomposition (MI) [21], etc.

More specifically, the decomposition based selection is described as follows. First, assume that a labeled data set $L = \{(x_i, y_i)\}_{i=1}^n$, $x_i \in \mathbb{R}^d$ and $y_i \in \{+1, -1\}$, consists of two views with their respective feature decompositions:

$$L_1 = \left\{(x_i^{(1)}, y_i)\right\} \quad and \quad L_2 = \left\{(x_i^{(2)}, y_i)\right\}, \tag{3}$$

where $x_i^{(1)} \in \mathbb{R}^{d_1}$, $x_i^{(2)} \in \mathbb{R}^{d_2}$, and $d = \sum_{j=1}^2 d_j$.

Using a RD, for example, L (and U) is divided into L_k (and U_k), $(k = 1, 2)$. Then, after training h_k using L_k, classification error rates of h_k, $\epsilon(h_k)$, are evaluated using L_j, $(j \neq k)$. For all $x_{ki} \in U_k$, after measuring the two quantities, $p_{ki} (\equiv p(x_{ki}))$ and $q_{ki} (\equiv q(x_{ki}))$, a weighted average for $x_i \in U$, $\rho_3(x_i)$, is measured as follows:

$$\rho_3(x_i) = \sum_{k=1}^{2} \frac{1}{\epsilon(h_k)} \rho_k(x_i), \tag{4}$$

where $\rho_k(x_i)$ is measured using (1) or (2) for L_k. Then, $\rho_3(x_i)$ is used as a confidence level for $x_i \in U$.

In (4), however, if $\epsilon(h_k)$ is near zero, then the advantage of the multi-view setting is lost, i.e., the value of ρ_3 becomes solely dependent on the value of ρ_k, and not the values of the other counterparts, ρ_j, $(j \neq k)$. To avoid this, the expected (average) value of information contained in each view is considered in lieu of the classification error rates. When deriving the target confidence in (4), the Shannon entropy can be used as a weight as follows.

Assume that a discrete probability space, L_k, $(k = 1, 2)$, which is composed of two classes having $p_E^{(1)}$ and $p_E^{(2)}$, respectively, as the class-conditional (a posteriori) probability estimated using L_1 and L_2, where $p_E^{(1)} + p_E^{(2)} = 1$. Then, from the definition, the Shannon entropy of L_k is: $H(L_k) = -(p_E^{(1)} \log_2(p_E^{(1)}) + p_E^{(2)} \log_2(p_E^{(2)}))$, which is the information contained in the probability space. Referring to this quantity, another confidence value for $x_i \in U$ in (4), $\rho_4(x_i)$, can be measured as follows:

$$\rho_4(x_i) = \sum_{k=1}^{2} H(L_k) \rho_k(x_i), \tag{5}$$

where $H(L_k) (\equiv H(P(L_k)))$ denotes the Shannon entropy measured from L_k. Also, $\rho_k(x_i)$ is measured using (1) or (2) for L_k.

3. Proposed method

In this section, a multi-view based selection strategy and its SemiBoost type learning algorithm are presented. For easy explanation, the proposed method is presented by dividing it into two algorithms, the selection and learning algorithms as in [20].

3.1. Transformation based methods

First, the feature-transformation based selection algorithm is described. As mentioned previously, the transformation based method is very similar to the decomposition based method, in which the feature set is decomposed into multiple subsets. However, in the transformation based method, a number of views can be obtained from the entire feature set using the same number of mapping functions, including geometrical transformations (e.g., affine transformations, local elastic deformations, Zernike moments, etc.) and statistical transformations (e.g., locality preserving projection, classical multidimensional scaling, neighborhood preserving embedding, etc.). For example, when using two mapping functions (named F_1 and F_2, respectively) to generate two views, a selection algorithm, called the transformation-based selection (TBS), can be summarized as follows (refer to Algorithm 1).

For the sake of completeness, in addition to TBS, the selection algorithm developed in [20], named decomposition-based selection (DBS), is introduced similarly as follows (refer to Algorithm 2). However, in DBS, the criterion in (4) is used as a selection criterion, rather than that in (5). Also, a feature-decomposition method (named f_D) is used.

In the above two algorithms, the processing CPU-time of Step 1 is determined depending on which transformation function (and decomposition method) is used. Meanwhile, the processing CPU-times of the remaining steps are the same as those of the other transformation functions (and decomposition methods).

Algorithm 1 Transformation-based selection (TBS).

Input: labeled data (L), unlabeled data (U), β (% of selected U_s over U), and two mapping functions (F_1 and F_2).
Output: selected unlabeled data and their pseudo-labels (U_s).
Procedure: perform the following steps to select U_s from U by transforming L into L_1 and L_2, where $L \in \mathbb{R}^d$ and $L_k \in \mathbb{R}^{d_k}$.

1. After transforming $L \in \mathbb{R}^d$ into L_k, $(k = 1, 2)$, using F_1 and F_2, respectively, for each view, train a (supervised) classifier, h_k, using $\mathcal{A}$ with L_k.
2. For each L_k, compute the Shannon entropy, $H(L_k)$; also, using the same functions as in transforming L, transform U into U_k for subsequent steps.
3. For all $x_{ki} \in U_k$, after computing p_{ki} and q_{ki} using L_k, U_k and h_k, compute the confidence levels of $x_i \in U$, ρ_i, in (5).
4. After sorting $\{x_i\}$ in decreasing order on key $\{|\rho_i|\}$, choose U_s from top $\beta(\%)$ of U, and estimate pseudo-labels for all $x_j \in U_s$ using $sign(\rho_j)$.

End Algorithm

Algorithm 2 Decomposition-based selection (DBS).

Input: labeled data (L), unlabeled data (U), β(% of selected U_s over U), and a decomposition method (f_D).
Output: selected unlabeled data and their pseudo-labels (U_s).
Procedure: perform the following steps to select U_s from U by decomposing L into L_1 and L_2, where $L \in \mathbb{R}^d$, $L_k \in \mathbb{R}^{d_k}$ and $d = \sum_k d_k$.

1. After dividing $L \in \mathbb{R}^d$ into L_k, $(k = 1, 2)$, using a f_D, for each view, train a (supervised) classifier, h_k, using $\mathcal{A}$ with L_k.
2. For each h_k, compute the error rate, $\epsilon(h_k)$, using L_j, $(j \neq k)$; also, using the same way of decomposing L, divide U into U_k.
3. For all $x_{ki} \in U_k$, after computing p_{ki} and q_{ki} using L_k, U_k and h_k, compute the confidence levels of $x_i \in U$, ρ_i, in (4), i.e. $\rho_i \leftarrow \sum_k \frac{1}{\epsilon(h_k)} \rho_{ki}$.
4. This step is the same as Step 4 in TBS.

End Algorithm

More specifically, the time complexities of both TBS and DBS, in which a pair of F_1 and F_2 and a f_D are employed, respectively, can be analyzed as follows. In Step 1, compared to SemiBoost, the execution time needed for the feature transformation in TBS is additionally required, while, in DBS, the time for the feature decomposition is additionally needed. In general, the latter takes shorter time than the former does. However, in Steps 2 and 3, which consist of computing the p_i and q_i quantities, the time needed for the TBS and DBS algorithms is more than twice equally that of the selection in SemiBoost. Finally, in Step 4, the execution times needed for both algorithms are the same. From this analysis, it can be seen that the required time for the present algorithms is generally greater than that for SemiBoost.

3.2. Learning algorithm

Second, an algorithm that upgrades the traditional SemiBoost classifiers using the TBS (and DBS) algorithm for selecting helpful U_s from U is presented.

First, for TBS, the learning algorithm begins by predicting the pseudo-labels of U using a supervised classifier that has been initialized with L only. After setting the related parameters, e.g. the kernel function and its related conditions, a small amount of U examples are selected as U_s first. Then, a base classifier h is trained using U_s as well as L. After updating the ensemble classifier H using the trained base classifier h for the next iteration, the labels of U examples are predicted again. This select-and-predict process is repeated while increasing j from 0 to $J-1$ (where J is a predefined number of iterations) with an increment of 1. Based on this brief explanation, a SemiBoost type learning algorithm using TBS can be summarized as follows (refer to Algorithm 3).

Algorithm 3 Learning algorithm.

Input: labeled data (L), unlabeled data (U), # iterations (J), and a supervised learning algorithm ($\mathcal{A}$).
Output: final ensemble classifier (H).
Method:
Initialization: train an ensemble classifier, $H^{(0)}$, using L only.
Procedure: repeat the following steps while increasing j from 0 to $J-1$ in increments of 1.

1. Select $U_s^{(j)}$ (and their pseudo-labels) from U by invoking TBS, for example, with T (or L for the first iteration) and U.
2. After expanding the training data T using $U_s^{(j)}$, i.e. $T \leftarrow L \cup U_s^{(j)}$, train a base classifier, $h^{(j)}$, using the learning algorithm $\mathcal{A}$.
3. For all $x_i \in U$, after measuring p_i and q_i using L, U, and $h^{(j)}$, compute the step length, $\alpha^{(j)}$, as in SemiBoost [23], i.e. $\alpha^{(j)} \leftarrow \frac{1}{4} \ln \frac{\sum_i p_i \delta(h_{ji},1)+\sum_i q_i \delta(h_{ji},-1)}{\sum_i p_i \delta(h_{ji},-1)+\sum_i q_i \delta(h_{ji},1)}$, where $h_{ji} \equiv h^{(j)}(x_i)$ and $\delta(a,b) = 1$ when $a = b$ and 0 otherwise.
4. Update the ensemble classifier $H^{(j+1)}$ using $H^{(j)}$, $\alpha^{(j)}$, and $h^{(j)}$ for the next iteration, i.e. $H^{(j+1)} \leftarrow H^{(j)} + \alpha^{(j)} h^{(j)}$.

End Algorithm

Next, for DBS, in order to learn an ensemble classifier, the same algorithm as in Algorithm 3 can be used as well. Thus, the description of the DBS learning algorithm is omitted here in order to avoid repetition, but the rationale for this kind of learning is presented in subsequent sections, together with illustrative examples and experimental results obtained using real-world data.

4. Experimental results

In this section, in order to compare the effectiveness of TBS and DBS, experimental results obtained using benchmark data are presented. Illustrative examples are presented first, followed by the results for real-world data.

4.1. Experimental setting

The proposed method, TBS, was tested and compared with traditional methods, including DBS. This was done by performing experiments on well-known benchmark data: the NIST digit-image database (32 × 32-dimensional, ten-class, 500 samples per class) [13] and other multivariate datasets cited from the UCI Machine Learning Repository [1].

In all the experiments conducted in subsequent sections, the training and test steps were performed as follows: first, each dataset was randomly divided into three subsets (i.e., L, L_{test}, and U) with ratios of 5%: 35%: 60%, for example; second, U_s (the top 10% cardinality of U) was selected from U by referring to the confidence values measured in TBS and DBS; third, using the expanded training dataset (i.e., $L \cup U_s$), a base classifier was trained and evaluated, in which support vector machines (and decision trees) were used as the base classifier; fourth, based on the result obtained in the above step, an ensemble classifier, which was initialized with L at the beginning, was updated. After repeating this select-and-train process $J(= 50)$ times, the obtained classification performance of the ensemble classifier was evaluated using L_{test} in terms of classification accuracy.

Table 1
Four ensemble methods designed using the expanded training data ($L \cup U_s$).

Methods	Learning algorithm (learning mode)	How to select U_s from U
SB	SemiBoost [23] (single-view)	Using (1) in the input-feature space
SB2	modified SemiBoost [19] (single-view)	Using (2) in the input-feature space
SB2 – decom	Algorithm 3 (multi-view using DBS)	Using (3) in the decomposed subspaces
SB2 – trans	Algorithm 3 (multi-view using TBS)	Using (5) in the transformed subspaces

For TBS (and DBS), as was done in [19] and [20], just 10% of the available U (i.e., $\beta = 10$) was selected based on the confident level and utilized at the subsequent training steps in Algorithm 1 (and Algorithm 2) (refer to Section 5.3 in [19] for more details on this selection)

Throughout the experiments, for each dataset, the training-and-evaluation was repeated *100* times and the results obtained were averaged, unless stated otherwise. Specifically, in subsequent sections, the ensemble classifiers were designed in *four* ways, designated here as *SB, SB2*, *SB2 – decom*, and *SB2 – trans*, respectively. The details of these approaches are as follows:

1. *SB*: a single-view SemiBoost learning is performed using the SemiBoost algorithm [23] with L and U, in which U_s is selected from U using the criterion of (1).
2. *SB2*: a single-view SemiBoost learning is performed using the modified SemiBoost algorithm [19] with L and U, in which U_s is selected from U using the criterion of (2).
3. *SB2 – decom*: a multi-view SemiBoost learning is performed using Algorithm 3 with L and U, in which U_s is selected from U using the criterion of (3) through the multi-view setting after obtaining multiple views with DBS presented in Algorithm 2 [20].
4. *SB2 – trans*: the same multi-view SemiBoost learning as in *SB2 – decom*, but U_s is selected using the criterion of (5) through the multiple views obtained with TBS in Algorithm 1.

In the above approaches, SB and SB2 were performed in the input-feature space, i.e. the selection of U_s, the training of an ensemble classifier, and its evaluation were all performed in the same space as the input-feature space. However, in SB2-decom (and SB2-trans) the multi-view approach is different: the training and evaluation were similarly performed in the input-feature space, but the selection of U_s was obtained from the two decomposed (and transformed) subspaces using a feature-decomposition method (and two mapping functions). Table 1 summarizes the above.

Further, in order to obtain the two views when measuring the confident values in TBS, *only* four mapping functions were considered. The four mapping functions employed are itemized as follows:

1. *FOU* (Fourier coefficients): one of the well-known descriptors employed for representing closed shapes, which corresponds to the low frequency components of the boundary of the shape.
2. *KLT* (Karhunen–Loève transform): the best known linear feature extractor with which d'-dimensional feature sets are extracted from d-dimensional vectors through a $d \times d'$ transformation matrix [13].
3. *PCA* (principal component analysis): the same mapping as KLT, but the columns of the $d \times d'$ matrix are obtained from the overall covariance matrix (weighted by the class prior probabilities) [13].
4. *SNE* (t-distributed stochastic neighbor embedding): an embedding technique for mapping high-dimensional points into a low-dimensional space based on stochastic selection of similar neighbors [37].

For the above functions, FOU was considered to be an extractor for geometrical information and the others were included as a statistical technique. In particular, KLT and PCA are very similar, except for computing the covariance matrix.

Here, PCA was considered as a linear mapping function, while SNE as a non-linear technique. Both the functions were implemented using the software provided in DRToolbox (i.e., `compute_mapping`)[2] so that the experiment can be reproduced fairly. In all the experiments, for SNE, only two arguments, dataset (a matrix) and dimension (an integer), were given as input parameters. Then, the other arguments, such as the perplexity of the Gaussian kernel and the initial solution, were provided as the 'default' values. In particular, the dimensionality to be reduced was set as that of the original data, meaning that the mapping was done without dimensionality reduction.

Also, in SB2-trans, only two views obtained using pair-wised mapping functions were considered. That is, among FOU, KLT, SNE, and PCA, six combinations, which are referred to as FOU-KLT (shortly F-K), FOU-SNE (F-S), FOU-PCA (F-P), KLT-SNE (K-S), KLT-PCA (K-P), and SNE-PCA (S-P), respectively, were implemented.

For all of the above approaches, in order to evaluate the classification accuracies, an SVM was implemented as the base classifier using the software provided in LIBSVM [4] (i.e., `svmtrain` and `svmpredict`)[3] and using linear/nonlinear kernel functions. More specifically, the (Gaussian) radial basis function kernel ($\Phi(x, x') = exp\exp(-(\|x - x'\|_2^2)/2\sigma^2)$) was used; the scale parameter (σ) was found using the cross-validation method; and the two constants, C^* and C, were set at 0.1 and 100, respectively. In order to further investigate the run-time characteristics of the approaches, decision trees were also considered as the base classifier.

In addition to the above approaches, in order to compare the proposed method with state-of-the-art multi-view co-training methods, CoTrade (confident co-training with data editing) [42], which is a publicly available software package, was also included in the comparison. Here, the two views provided for CoTrade were obtained using the same methods as those used in *SB2 – decom* (and *SB2 – trans*). Also, in the comparison, SL-SVM (supervised learning SVM using L only) was included as a baseline in order to make it complete.

4.2. Preliminary experiments

In order to illustrate the functioning of both DBS and TBS, prior to presenting the classification accuracies of the SemiBoost classifiers, experimental parameters for the methods were first considered and adjusted through an experiment. In order to achieve the goal of this experiment simply but clearly, two 1024-dimensional, two-class datasets, named N3-7 and N32-71, were manually reconstructed from the NIST database. For N3-8, '3' and '7' digits were selected as the two classes, while, for N32-71, both '3' and '2' digits were selected as one class; both '7' and '1' digits were selected as the other class, 1000 samples per class. That is, the data complexities of the two synthetic datasets are different.

[2] https://lvdmaaten.github.io/software/
[3] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

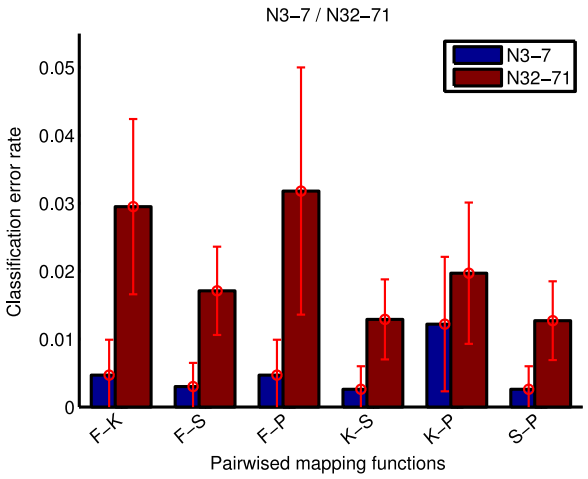

Fig. 1. Plot comparing the classification error rates obtained using six SVM ensembles for N3-7 and N32-71. Here, six combinations, FOU-KLT (F-K), FOU-SNE (F-S), FOU-PCA (F-P), KLT-SNE (K-S), KLT-PCA (K-P), and SNE-PCA (S-P), were considered to obtain the two views when measuring the confidences in TBS.

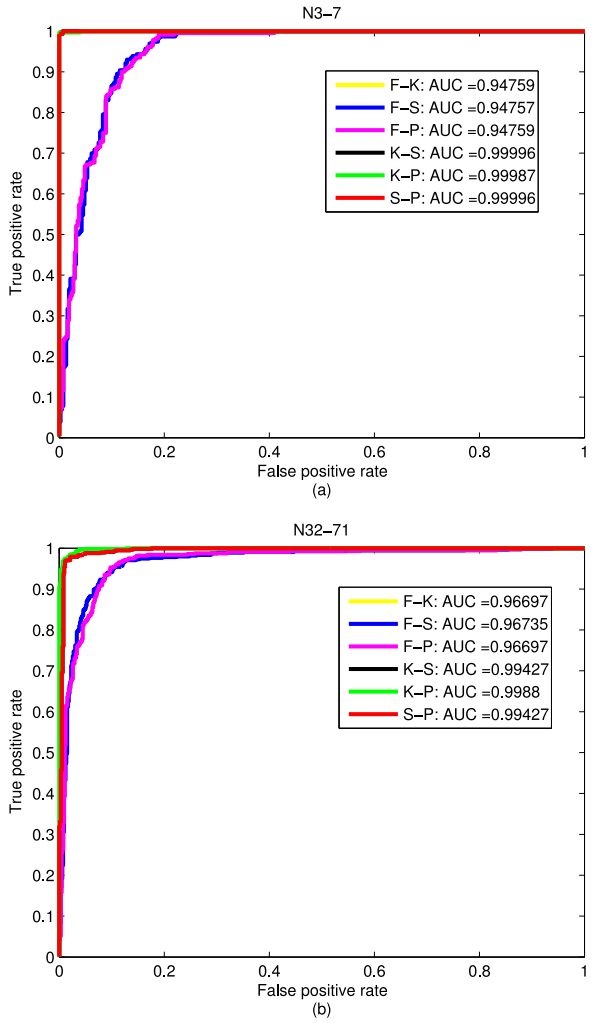

Fig. 2. ROC curves obtained using the six SVM ensembles with different pair-wised combinations for the N3-7 and N32-71 data: (a) N3-7 and (b) N32-71. Here, the x-axis denotes the fraction of false positives out of the total actual negatives (FPR: false positive rate), while the y-axis denotes the fraction of true positives out of the total actual positives (TPR: true positive rate).

4.2.1. View transformations

First, in order to investigate the efficiency of the six combinations, using N3-7 (and N32-71), ensemble classifiers were implemented with SB2-trans and evaluated as follows: first, the dataset was divided into three subsets with ratios of 5%: 35%: 60%; second, after obtaining two views using pair-wised mappings, U_s was selected from U in TBS; third, using $L \cup U_s$, an ensemble classifier was trained, in which SVM was employed as a base classifier. Fig. 1 shows a graphical comparison of the classification error rates obtained using the ensemble classifiers trained with the six combinations for N3-7 and N32-71. Here, the training-and-test process was repeated *100* times.

From the figure, it can be clearly observed that, for the two datasets of N3-7 and N32-71, each combination achieves similar accuracies, meaning that, among the six pair-wised combinations, a specific combination benefits more from TBS than the other ones. More specifically, for both N3-7 and N32-71, the lowest error rate can be obtained commonly using S-P or K-S, at least for this specific classification.

However, the comparison shown in Fig. 1 might be less informative for choosing appropriate mapping functions and, therefore, cannot be helpful for combining the pair-wised combination. Thus, in addition to this comparison, in order to further investigate the efficiencies, receiver operating characteristic (ROC) curves were considered again. Fig. 2 presents a comparison of the ROC curves obtained using the six SVM ensembles with different pair-wised combinations for N3-7 and N32-71. Here, the training-and-test is repeated *100* times and the results obtained are averaged.

From Fig. 2(a) and (b), it should be commonly observed that the classification performance of the SVM ensemble based on some combinations is better than those of the other combinations. More specifically, the values of AUC (area under the receiver operating curve) can be considered, where the larger value results in better accuracy. By referring to the AUC values, the six combinations can be grouped into two: F-K, F-S and F-P and K-S, K-P and S-P. In particular, S-P is one of the largest on both the pictures.

From these observations, in subsequent comparisons of the classification accuracies, the pair-wised combination of the mapping functions was fixed to be S-P for all the datasets. However, it is well-known that there are numerous mapping functions developed for specific needs. Thus, a question arises: *how to choose the most suitable combination, which will work well for a problem in hand?* The theoretical investigation to answer this remains to be done.

4.2.2. Entropy based criterion

Second, in order to investigate the underlying reason for using the entropy based criterion when measuring the target confident value in TBS, rather than using the error based one, a qualitative exploration was made as follows. Using N3-7 (and N32-71), the same ensemble classifier as for Fig. 2 was implemented in SB2-trans using the two criteria of (4) and (5). Here, SVM was employed as a base classifier too and the transformation was performed using S-P only, as mentioned previously. Fig. 3 presents a comparison of the ROC curves obtained using the two criteria for TBS and DBS (referred to as TBS-Ent, TBS-Err, DBS-Ent, and DBS-Err, respectively). In the comparison, the results obtained using DBSs were included as a reference in order to make it complete. From Fig. 3(a) and (b), it should be commonly observed that the classification performance of the SVM ensemble classifiers implemented in TBS using the criterion of (5) is better than that of (4). That is, for both N3-7 and N32-71, the AUC value of TBS-Ent is larger than that of TBS-Err as well as both DBSs.

Based on these observations, in all the experiments subsequently conducted, the ensemble classifiers of SB2-trans were trained using the criterion of (5) with the S-P combination.

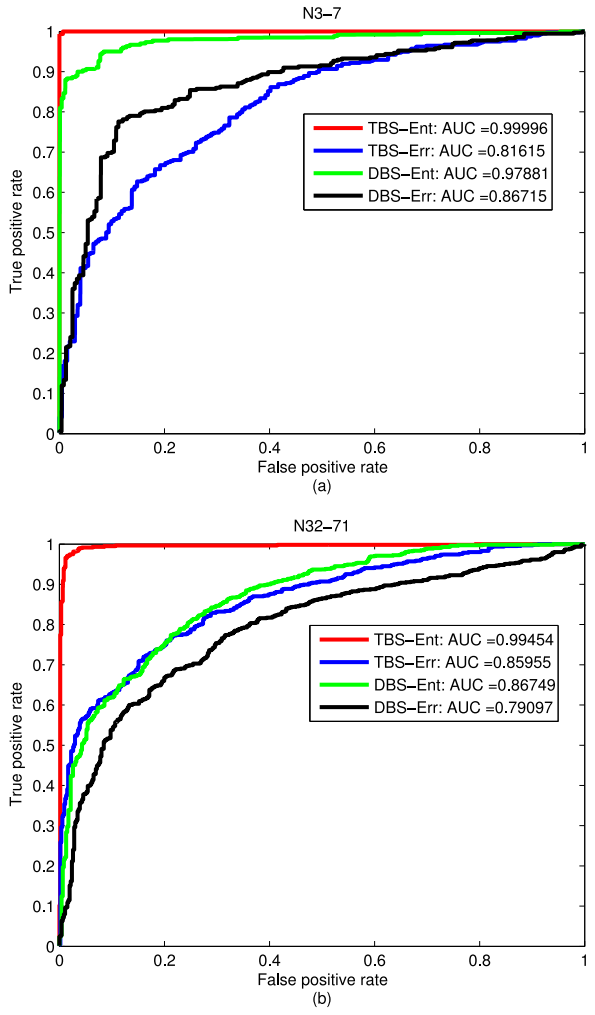

Fig. 3. ROC curves obtained using the two criteria of (4) and (5) for the N3-7 and N32-71 data: (a) N3-7 and (b) N32-71. Here, the *x*-axis denotes the fraction of false positives out of the total actual negatives (FPR: false positive rate), while the *y*-axis denotes the fraction of true positives out of the total actual positives (TPR: true positive rate).

4.2.3. Classification accuracies

Third, as a preliminary experiment, classification accuracies were measured and compared, in which the classification was performed in SB, SB2, SB2-decom, and SB2-trans scenario using the same parameters as observed previously. Particularly, in SB2-decom, the decomposition of the feature set was performed with a RD, while, in SB2-trans, the feature-transformation was performed using S-P, as mentioned. Table 2 presents a numerical comparison of the classification mean error rates (and standard deviations) (%) obtained using N3-7 and N32-71. Here, the training-and-test was also repeated *100* times.

According to the results shown in Table 2, even narrowly, it seems that the ensemble classifier trained in multi-view approaches works better than the others in terms of classification accuracy. More specifically, the values of the column of SB2-trans are the lowest for each dataset. From this observation, it can be noted that using multiple views may lead to an improvement in classification and, in particular, SB2-trans (i.e., TBS) will probably be better than SB2-decom (i.e., DBS) as a selection strategy.

4.3. Experiment # 1 (NIST data)

Using the NIST real-world data, the classification accuracies of the two ensemble classifiers learned in DBS and TBS were compared and analyzed. Generally, for the multi-class data, classification can be performed in two fashions: using either one-against-all (OAA) or one-against-one (OAO) strategies. For the OAA strategy, classification is performed in the positive/negative problem, where a set of positive examples consists of the samples of a selected class and the remainder of the classes is treated as a set of negative ones. In contrast, for the OAO strategy, classification is done with two classes selected in order, and the rest of the classes are discarded.

In this experiment, OAA classification was performed first, followed by that of OAO. Also, the same ensemble classifiers (SVMs) as those employed for Table 2 in Section 4.2 were trained and evaluated empirically. In addition, in order to reduce computational complexity and to simplify the classification task for the paper, 10 % of the data samples were randomly selected. As a consequence, the total number of samples per class was 50, not 500. Table 3 presents a numerical comparison of the OAA classification mean error rates (and standard deviations) (%) obtained using NIST, in which classification was performed using the approaches of SB, SB2, SB2-decom, and SB2-trans, including SL-SVM and CoTrade. In particular, in SB2-decom, RD was used, while, in SB2-trans and CoTrade, S-P was invoked.

The following observations can be obtained from the data presented in Table 3. First, it is necessary to consider the two columns (SB and SB2) of the single-view approaches and the two columns (SB2-decom and SB2-trans) of the multi-view approaches. As a simple comparison, the lowest values that each method earned for the datasets were counted and compared. The numbers of wins that the six methods earned in the competition (i.e., the numbers of the $\star$ marker) were 0, 0, 1, 0, 8, and 1, respectively. From this comparison, it can be observed that SB2-trans obtained the highest count, meaning that the classification accuracy of the ensemble classifier can generally be improved by using TBS.

Second, it is important to consider the columns of SB2-decom, SB2-trans and CoTrade separately. From this consideration, it can be observed that there is no specific approach that yields the best results for *all* the classes of NIST in terms of the classification accuracy. However, more simply, the values of the error rates for each dataset were averaged again and compared (refer to the most below μ's). From this comparison, it can be noted that SB2-trans works *marginally* better than the others in terms of classification accuracy.

In addition to OAA, OAO classification was performed as follows: first, 45 subsets of N$i-j$, $(i=0,\cdots,9;\ j>i$ for each i, where N$i-j$ denotes that 'i' and 'j' digits were selected as the

Table 2
Numerical comparison of the classification error (and standard deviation) rates (%) obtained using the single-view and multi-view approaches for the N3-7 and N32-71 datasets.

Datasets	SL	Single-view SSL		Multi-view SSL		Co-Trade
	SVM	SB	SB2	SB2-decom	SB2-trans	
N3-7	3.29 (1.34)	2.64 (1.52)	2.60 (3.20)	2.07 (1.95)	0.30 (0.36)	0.36 (0.32)
N32-71	4.64 (0.91)	9.20 (4.47)	3.04 (1.43)	3.64 (1.09)	1.10 (0.53)	2.29 (0.49)

Table 3
Numerical comparison of the OAA classification mean (standard deviation) error rates (%) obtained using the six approaches for the NIST data. Here, in order to facilitate the comparison in the table, the lowest error rate in each data is highlighted with a ⋆ marker.

Digit	SL	Single-view SSL		Multi-view SSL		Co-Trade
subsets	SVM	SB	SB2	SB2-decom	SB2-trans	
N0-all	5.40 (0.77)	8.28 (4.01)	4.71 (2.97)	4.83 (1.50)	⋆2.99 (1.43)	3.79 (2.21)
N1-all	5.37 (1.32)	6.21 (3.02)	6.09 (2.95)	6.78 (2.49)	⋆3.28 (1.14)	5.29 (2.91)
N2-all	8.05 (1.86)	15.98 (5.68)	10.69 (1.55)	9.20 (2.60)	6.44 (2.35)	⋆6.09 (4.57)
N3-all	7.01 (0.94)	8.85 (3.22)	⋆4.94 (1.97)	9.20 (1.22)	5.98 (2.98)	5.98 (2.32)
N4-all	6.44 (1.25)	10.34 (4.47)	7.01 (4.92)	7.70 (4.68)	⋆2.99 (1.37)	5.29 (4.76)
N5-all	8.85 (1.04)	9.80 (1.18)	9.08 (1.31)	11.25 (3.69)	⋆6.85 (2.89)	8.97 (3.00)
N6-all	9.89 (1.37)	10.31 (1.18)	9.08 (1.03)	12.80 (3.89)	⋆8.89 (0.63)	9.97 (2.36)
N7-all	6.67 (2.39)	8.05 (2.26)	7.47 (1.82)	11.26 (3.03)	⋆6.62 (3.25)	7.70 (2.06)
N8-all	9.66 (0.48)	15.86 (6.55)	14.25 (5.23)	14.14 (5.25)	⋆8.95 (3.05)	11.95 (3.00)
N9-all	8.39 (0.96)	9.89 (1.88)	9.89 (1.03)	11.72 (3.85)	⋆8.34 (1.08)	10.57 (2.36)
# wins	0	0	1	0	8	1
means (μ)	7.07 (1.24)	10.22 (3.49)	7.63 (2.28)	8.82 (2.96)	5.50 (2.00)	6.62 (2.66)

Table 4
Some statistics observed from the OAO classification error (and standard deviation) rates (%) for the 45 Ni − j datasets.

Items	SL	Single-view SSL		Multi-view SSL		Co-Trade
	SVM	SB	SB2	SB2-decom	SB2-trans	
# wins	0	0	0	0	29	17
Means (μ)	3.12 (1.38)	4.20 (2.74)	3.31 (2.03)	3.33 (2.31)	1.57 (2.04)	1.72 (1.17)
Max / min	7.43 / 0.28	12.94 / 0.29	8.37 / 0.26	5.89 / 1.71	4.35 / 0.15	4.01 / 0.18

two classes), were generated and divided into three subsets with ratios of 5%: 35%: 60%; second, for each Ni − j dataset, the same ensemble classifiers as those for Table 3 were trained and evaluated; third, the training-and-test process was repeated *100* times and the results obtained were averaged. However, rather than tabulating the experimental results obtained, in the interest of readability, the results of the OAO classification were summarized as shown in Table 4.

In Table 4, as compared in Table 3, the lowest values (i.e., the numbers of the ⋆ marker) that each method earned for the 45 datasets were compared first. From this comparison, it can be observed again that SB2-trans obtained the highest count, meaning that the classification accuracy of the ensemble classifier can be improved using TBS. Next, from the comparison of the mean values for SB2-trans and SB2-decom, it can also be noted that SB2-trans works *marginally* better than SB2-decom in terms of classification accuracy.

4.3.1. Individual and in-combination transformations

Meanwhile, in SB2-trans for OAA and OAO, the classification steps were performed in two different spaces: original input-feature spaces and transformed subspaces. The training of an ensemble classifier and its evaluation were performed in the input space, but the selection of U_s was obtained from the two subspaces that were obtained using SNE and PCA, respectively. Thus, it is interesting to investigate the run-time characteristic of a case in which the selection was achieved from the individual subspace obtained using SNE and PCA *separately* (refer to as SB2-SNE and SB2-PCA, respectively), rather than using them *in-combination* (refer to as SB2-trans with SNE-PCA). That is, in SB2-SNE (and SB2-PCA), the U_s selection was achieved from the subspace obtained using SNE (and PCA) individually, and then the training and test steps were performed in the input-feature space. Fig. 4 shows a graphical comparison of the classification error rates obtained using SB2-SNE and SB2-PCA and SB2-trans with SNE-PCA for NIST.

A significant observation from both pictures is that, for *all* of the OAA and OAO classifications, the combination method, SNE-PCA (S-P), outperforms the individual ones, SNE and PCA, in terms

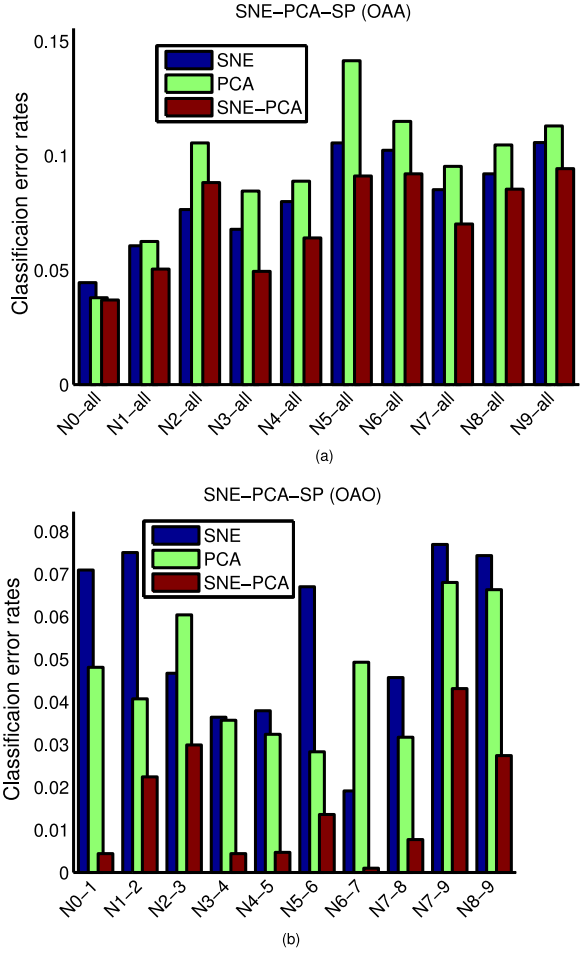

Fig. 4. Plots comparing the classification error rates obtained using SB2-SNE, SB2-PCA, and SB2-trans-S-P for OAA and OAO classifications. Here, in (b), the results of *ten* datasets are selected from those of the 45 Ni − j's.

Table 5
Some statistics observed from Table 4 in [20] comparing the classification error (and standard deviation) rates (%) among the SemiBoosts for the *ten* UCI datasets.

Items	SL	Single-view SemiBoosts (SB)		Multi-view SemiBoosts (MB)		CoTrade (CT)	
	SVM	SB	SB2	MB-RD	MB-FS	CT-RD	CT-FS
# wins	1	0	1	2	7	0	1
Means	34.09	28.20	27.45	22.58	22.14	30.90	28.59
(μ)	(1.69)	(3.38)	(2.93)	(2.99)	(2.56)	(3.43)	(4.93)
Max	48.33	42.22	42.53	42.32	42.09	47.38	42.22
Min	6.94	5.03	4.94	4.64	4.33	6.45	6.45

of classification accuracy. That is, all the heights of the error rate bars of the former (i.e., the third bars for each dataset) are *always* smaller than those of the latter (i.e., the first and the second bars). From this observation, it can be noted that, when measuring the confident values, TBS benefits more from multiple views of the data than DBS does.

In particular, compared to the two transformed single-view approaches (refer to the heights of the SB2-SNE and SB2-PCA bars), the multi-view approach (refer to the height of the SB2-trans-S-P bar) can provide the lowest error rate. This means that the strategy might be enforced by using the two views together, which were obtained through SNE and PCA, respectively.

In summary, the experimental results obtained using the NIST real-world data demonstrate that the transformation based methods can measure the confident values more efficiently from the transformed multiple views, as compared with measuring them individually from the original input-features, or the decomposition-based multiple views.

4.4. Experiment # 2 (UCI Data)

In order to further illustrate the functioning of both DBS and TBS, using a set of public domain datasets cited from the UCI Machine Learning Repository [1], their classification accuracies were evaluated again and compared.

The cited UCI datasets are: *Australian Credit Approval* (690/14/2), *Credit-A* (653/14/2), *Ecoli* (336/7/8)†, *Glass* (214/9/6)†, *Heart* (270/13/2), *Pima* (768/8/2), *Quanlity* (4898/11/7)†, *Segment* (2310/19/7)†, *Vehicle* (846/18/4)†, and *Vowel* (528/10/11)†, where the three numbers in brackets represent the values of # dimensions, # samples, and # classes, respectively. However, in order to treat all the datasets as a binary classification, the multi-class objects marked with a † symbol were divided into two subgroups by taking the balance. The details of these datasets are omitted here, but can be found in the related literature [1,5] and [20].

Using these UCI datasets, as was done in [20], various selection methods, including single-view approaches (SB), multi-view approaches (MB), and the approaches of CoTrade (CTs), were implemented and compared (refer to Table 4 in [20]). More specifically, for SBs, SemiBoost (SB) [23] and its variant (SB2) [19] were executed for the original feature set. Also, for MBs, multi-view based selections were performed together with different feature-decomposition methods, such as random methods (RD) and individual feature selection methods (FS) (which are referred to as MB-RD and MB-FS, respectively). In addition, for CTs, CoTrade [42] was implemented with the RD and FS schemes (which are referred to as CT-RD and CT-FS, respectively). From the comparison in [20], certain kinds of statistical information can be observed and summarized as in Table 5.

From the data presented in Table 5, the following observations can be made: first, among the three selection categories of SB, MB and CT, MB is generally superior to the others in terms of the classification accuracy; second, among the feature-decomposition methods of MB, none of them is superior to the others, i.e. the classification error rates of both MB-RD and MB-FS are very similar.

From these observations, the same experiment that was performed in Sections 4.2 and 4.3 was repeated and evaluated as follows: first, each UCI dataset was divided into three subsets with ratios of 20%: 20%: 60%; second, SVM ensemble classifiers were trained and evaluated. Table 6 presents a numerical comparison of the classification mean error rates (and standard deviations) (%) obtained using the SVM ensembles for the UCI data. Here, the training-and-test process was repeated *100* times.

In Table 6, the following observations can be made: first, the lowest values (i.e., the numbers of the ⋆ marker) that each method earned for the datasets were counted and compared. From this comparison, as in Table 3, it can be observed again that SB2-trans received the highest count, meaning that the classification accuracy of the ensemble classifier can be improved; second, when considering the three columns of SB2-decom, SB2-trans and CoTrade separately, it can also be clearly observed that for almost *all* the datasets, SB2-trans performs better than both SB2-decom and CoTrade.

Table 6
Numerical comparison of the classification mean (standard deviation) error rates (%) obtained using the SVM ensembles for the UCI data.

UCI	SL	Single-view SSL		Multi-view SSL		Co-Trade
datasets	SVM	SB	SB2	SB2-decom	SB2-trans	
Australian	45.33 (1.48)	34.38 (4.77)	32.48 (3.73)	32.12 (4.84)	⋆30.55 (3.77)	37.01 (3.85)
CreditA	45.62 (1.66)	39.62 (6.84)	37.62 (5.30)	37.77 (5.56)	⋆35.92 (6.15)	42.92 (4.67)
Ecoli	3.48 (0.73)	5.30 (1.79)	4.70 (2.08)	4.55 (1.75)	⋆3.18 (1.12)	4.03 (1.43)
Glass	35.33 (7.40)	35.57 (7.73)	35.81 (9.51)	34.43 (7.86)	⋆32.95 (8.80)	45.00 (12.37)
Heart	44.26 (0.59)	38.15 (7.05)	39.44 (8.14)	39.00 (6.06)	⋆37.26 (7.94)	44.26 (7.48)
Pima	34.64 (0.00)	35.42 (3.89)	33.20 (3.68)	30.76 (3.23)	⋆30.51 (3.87)	32.88 (3.54)
Quality	38.11 (9.36)	39.36 (9.27)	38.92 (8.50)	37.26 (10.12)	⋆36.01 (10.03)	42.55 (9.85)
Segment	34.94 (1.89)	7.27 (2.62)	7.49 (1.84)	7.75 (1.70)	⋆6.54 (1.38)	7.88 (1.37)
Vehicle	47.92 (0.81)	14.40 (2.75)	14.56 (2.70)	14.80 (2.54)	⋆12.57 (2.96)	23.21 (3.26)
Vowel	8.90 (3.78)	7.29 (3.57)	6.14 (2.81)	7.24 (3.87)	⋆5.10 (2.88)	7.71 (2.66)
# wins	0	0	0	0	10	0
means (μ)	33.85 (2.77)	25.68 (5.03)	25.04 (4.83)	24.57 (4.75)	23.06 (4.89)	28.75 (5.05)

Table 7
Numerical comparison of the classification mean (standard deviation) error rates (%) obtained using the CRT (classification and regression tree) ensembles for the UCI data.

UCI	SL	Single-view SSL		Multi-view SSL		Co-Trade
datasets	SVM	SB	SB2	SB2-decom	SB2-trans	
Australian	18.87 (4.60)	17.96 (4.29)	17.46 (4.43)	17.68 (3.62)	*15.81 (3.68)	29.93 (5.66)
CreditA	18.23 (4.30)	18.58 (4.46)	18.55 (4.74)	18.54 (4.30)	*15.66 (3.71)	33.74 (4.55)
Ecoli	9.15 (4.76)	9.70 (4.74)	9.24 (4.35)	9.36 (5.15)	6.91 (3.76)	*6.64 (3.22)
Glass	33.62 (8.12)	34.19 (9.45)	33.62 (8.07)	33.86 (9.27)	*29.57 (7.96)	34.05 (8.24)
Heart	28.56 (6.49)	28.59 (7.16)	28.41 (7.75)	28.19 (7.32)	*25.00 (5.94)	33.70 (7.18)
Pima	30.77 (4.23)	32.25 (4.16)	29.36 (4.08)	30.80 (4.54)	*27.37 (4.65)	31.16 (3.57)
Quality	42.72 (1.18)	40.59 (1.20)	39.76 (0.85)	40.56 (1.68)	*38.73 (0.89)	42.52 (1.20)
Segment	8.11 (1.78)	7.65 (1.57)	7.63 (1.84)	7.22 (1.72)	*4.20 (0.87)	7.60 (1.75)
Vehicle	15.07 (3.19)	14.98 (3.41)	14.68 (3.50)	14.36 (2.83)	*10.14 (2.75)	17.05 (3.53)
Vowel	17.31 (4.77)	17.26 (5.02)	16.08 (4.00)	17.05 (4.52)	*10.30 (3.28)	12.36 (4.07)
# wins	0	0	0	0	9	1
Means (μ)	22.24 (4.34)	22.18 (4.55)	21.48 (4.36)	21.76 (4.49)	18.37 (3.75)	24.88 (4.30)

Meanwhile, it is well-known that, empirically, ensembles tend to yield better results when the base classifiers are weak learners. From this viewpoint, rather than using the SVM, a SemiBoost classifier using weak learners such as decision trees as the base classifier was trained and evaluated under the same conditions as those in Table 6. Table 7 presents a numerical comparison of the classification mean error rates (and standard deviations) (%) obtained using the SemiBoost classifiers for the UCI data. Here, `classregtree` (classification and regression tree: CRT) package provided in Matlab was used as the weak learner.

In Table 7, the same observations can be made as observed in Table 6. Although it is hard to quantitatively compare the values, in order to determine the significance of the difference in both, the Student's statistical two-sample test [17] can be used. For the two learning algorithms generating μ_1 (σ_1) and μ_2 (σ_2), p-values were computed first. Next, using the p-values, a decision was made as to whether or not to accept the null hypothesis that *H0:* μ_1 (σ_1) = μ_2 (σ_2). As a consequence, the lower p-value of the hypothesis was more strongly negative. Figs. 5 and 6 present a comparison of the p-values obtained from the two t-tests using the error rates in Tables 6 and 7, respectively.

The observations obtained from the plots shown in Figs. 5 and 6 are as follows. First, consider the comparison of the p-values of $Pr\,(SB2 - trans < SB2 - decomp)$ (refer to Fig. 5 (a)). From the figure, it can be observed that, compared to SB2-decom, SB2-trans wins 7/10 datasets at the significance level of 0.05; the height of the bars is under the red dash-dotted line.

Second, consider the comparison of the p-values of $Pr\,(SB2 - trans < CoTrade)$ (refer to Fig. 5 (b)). From the figure, it can also be observed that, among the ten datasets, compared to CoTrade, SB2-trans has *all* datasets winning at a significance level of 0.01; the height of the bars is under the blue solid line. From these observations, it can be noted that for *all* the datasets SB2-trans performs better than CoTrade in terms of classification accuracy at the significance level of 1%, while for the *seven* datasets SB2-trans performs better than SB2-decom at the significance level of 1%.

Next, from Fig. 6, the comparison is similar to that shown in Fig. 5. Therefore, the description is omitted here in order to avoid repetition. However, it is worthwhile to note that, when using CRT ensembles rather than using SVM ones, SB2-trans performs better than both SB2-decom and CoTrade for the *almost* all datasets at the significance level of 1%. The only exception is of the *Ecoli* dataset when compared to CoTrade (refer to Fig. 6 (b)).

4.4.1. Individual and in-combination transformations

In addition to the above t-test comparison, as was done in Section 4.3.1 the results obtained in the original input-feature spaces and the transformed subspaces were compared, as follows. First, Fig. 7 shows a graphical comparison of the classification error rates obtained using the SVM and CRT ensembles trained in both DBS and TBS. More specifically, in DBS, the feature set was decomposed into two subsets (i.e., two views) using RD or MI, while, in TBS, the two views were obtained using SNE-PCA (i.e., S-P).

From the picture, a significant observation is that, for almost all of the UCI datasets, TBS outperforms both DBSs, i.e. DBS-RD and

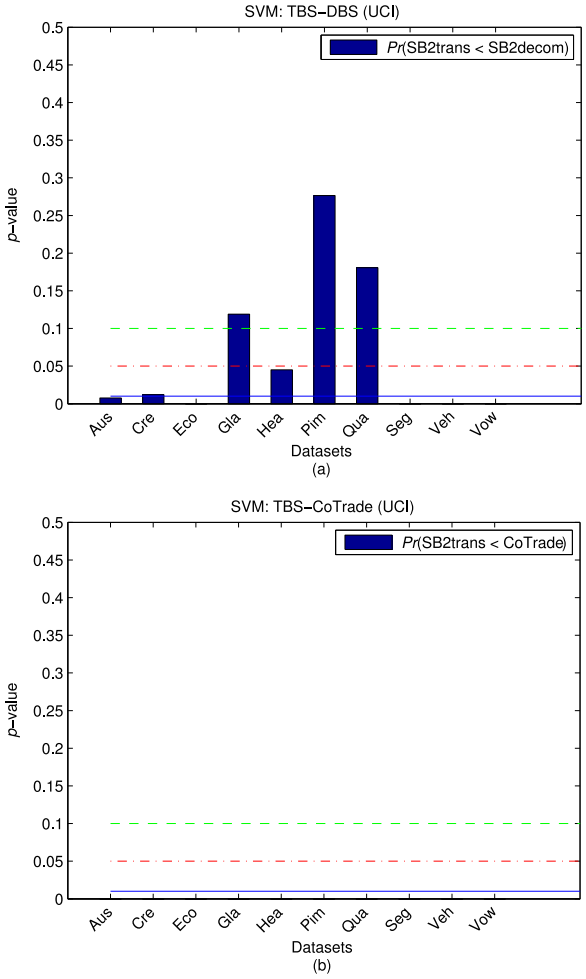

Fig. 5. Plots comparing the p-values obtained from the t-test using the error rates in Table 6: (a) $Pr(SB2trans < SB2decom)$ and (b) $Pr(SB2trans < CoTrade)$. Here, the datasets are represented with three letter acronym. Also, nonappearance means $p \approx 0.0$.

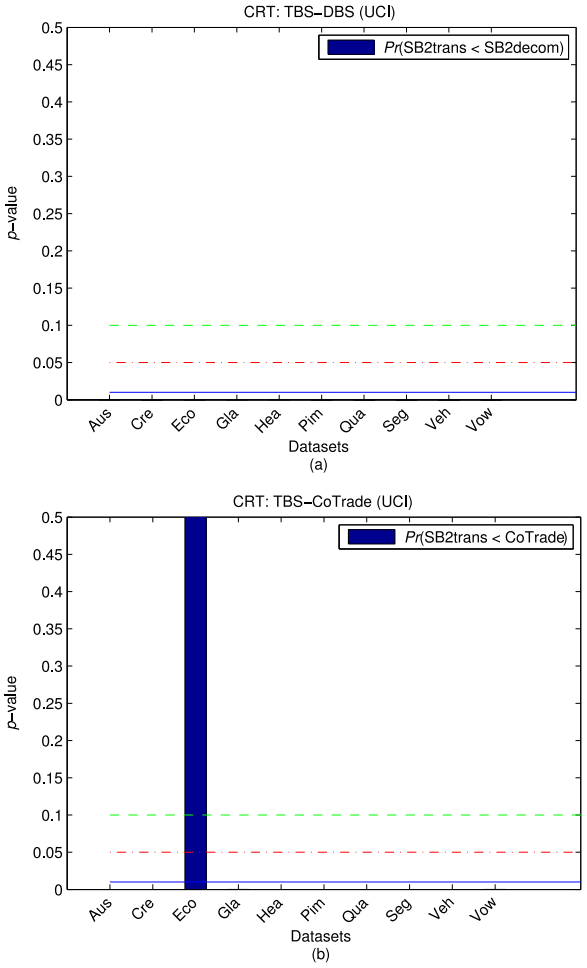

Fig. 6. Plots comparing the p-values obtained from the t-test using the error rates in Table 7: (a) $Pr(SB2trans < SB2decom)$ and (b) $Pr(SB2trans < CoTrade)$. Here, the datasets are represented with three letter acronym. Also, nonappearance means $p \approx 0.0$.

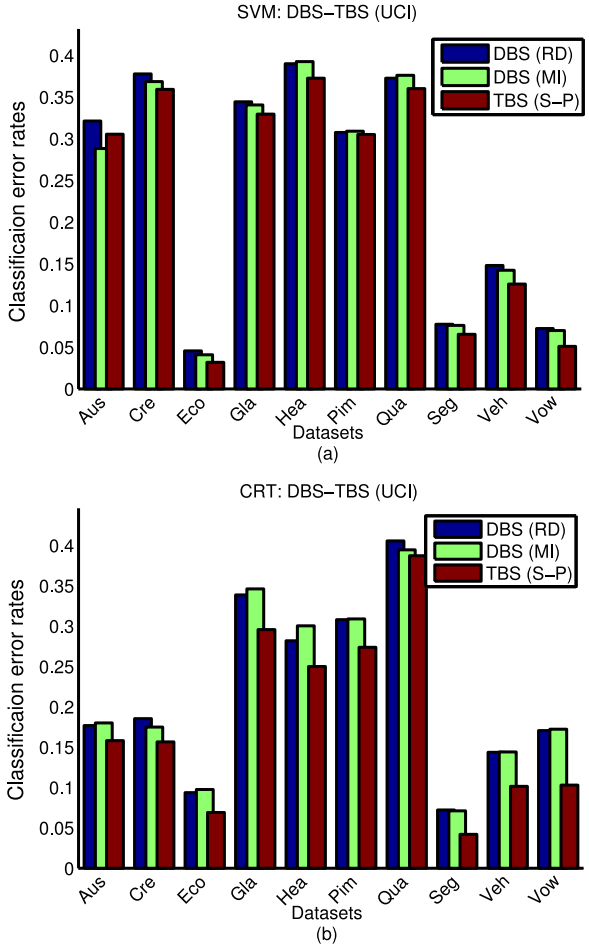

Fig. 7. Plot comparing the classification error rates measured using the SVM and CRT ensembles trained in DBS and TBS for the UCI data: (a) SVM ensemble classifiers and (b) CRT ensemble classifiers.

DBS-MI. Meanwhile, the comparison of the classification error rates obtained using both DBSs shows similar results.[4] This is in agreement with the comparison results observed in Table 5, meaning that there is no specific approach that yields the best results for all the families of decompositions.

Second, Fig. 8 shows a graphical comparison of the classification error rates obtained using the SNE and PCA mappings *separately*, and using them *in-combination* for the UCI data.

From Fig. 8 (a) and (b), as observed from Fig. 4, it can be observed again that, for all of the UCI data, SNE-PCA (S-P) outperforms both SNE and PCA. From this observation, the rationale for employing the TBS developed in the present work as a selection strategy is clear, rather than employing DBS or the traditional single view strategies, such as SemiBoost and its variant.

4.4.2. Comparison with other supervised classifiers

In Table 6 (and Table 7), the classification accuracies of the SVM (and CRT) ensemble classifiers were compared to those of the boosting based classifiers (semi-supervised learning approaches), in which SL-SVM trained with L only was included as a reference. In addition to these comparisons, the accuracies of the ensemble classifiers were further compared to those of traditional supervised classifiers, such as a k-nearest neighbor classifier (kNN) and a deep architecture neural network (deepNN). For the comparison, the two supervised classifiers have been implemented using publicly available R packages, i.e., `knn` and `darch`.[5] In particular, in order to make a fair comparison, the number of the user-relevant parameters is restricted to be as small as possible. That is, for kNN, the value of $k = 3$ was heuristically chosen. For deepNN, only the number of hidden layers was selected as a three-layer feed-forward architecture, and the back-propagation algorithm was used as the fine-tuning function for the architecture. The remaining arguments, such as the learning rate, the learning rate scale factors, etc. were determined as default values. Here, the three-layer architecture was implemented as follows: (INP, H1, H2, H3, OUT). The numbers of the neurons of the INP and OUT layers (i.e., #INP and #OUT) were determined by referring to the dimensionality (#features) and the number of the classes, respectively. Then, the numbers of the neurons of the three hidden layers (i.e., #H1, #H2, and #H3) were determined respectively as follows [24]: #H1 = #OUT*(r^3); #H2 = #OUT*(r^2); #H3 = #OUT*r, where r = (#INP/#OUT)^(1/4). Fig. 9 shows a graphical comparison of the classification performances between the ensemble classifiers and the supervised classifiers of kNN and deepNN.

From Fig. 9(a) and (b), it can be noted that for almost all datasets, classification accuracies of the SVM and CRT ensembles are superior to those of kNN and deepNN, i.e., bar graphs are lower for SVM and CRT than they are for others. However, for

[4] Here, in the interest of readability, the classification errors obtained using the RD and MI decompositions were presented in the figures, but the results of the other methods, such as GA and FS, were excluded.

[5] https://www.r-project.org/

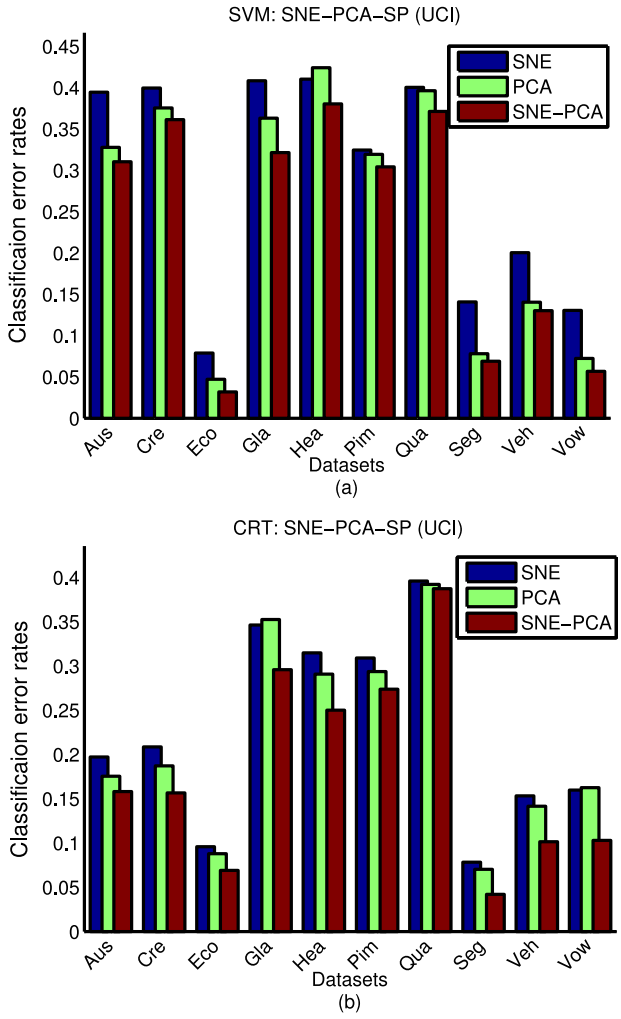

Fig. 8. Plots comparing the classification error rates obtained using both the mapping functions (SNE and PCA) separately and in combination for the UCI data: (a) SVM ensemble classifiers and (b) CRT ensemble classifiers.

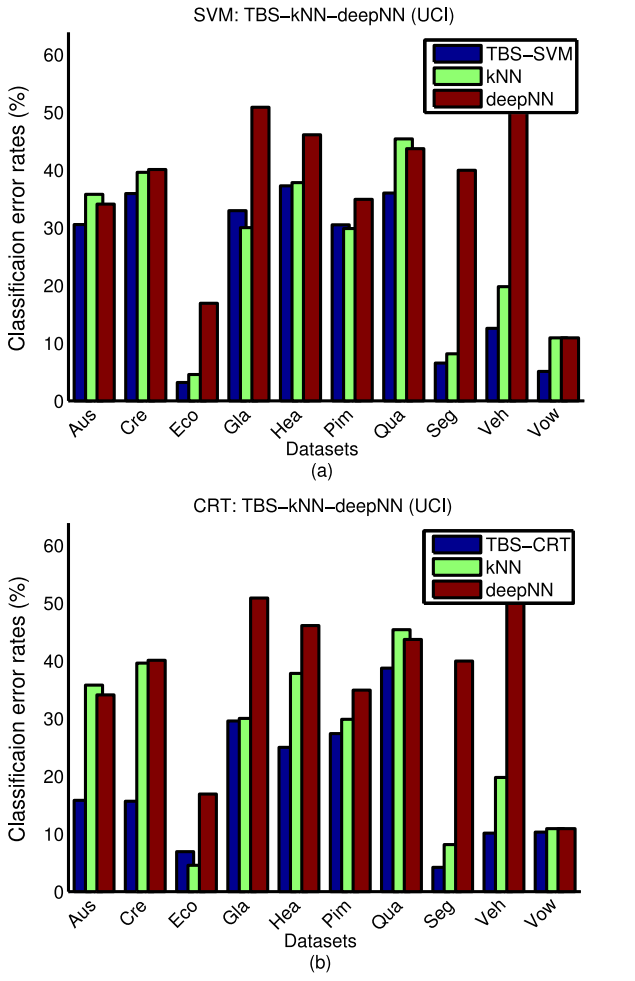

Fig. 9. Plots comparing the classification error rates obtained using the ensemble classifiers and the two supervised classifiers (i.e., *k*NN and deepNN) for the UCI data: (a) SVM ensemble classifiers and (b) CRT ensemble classifiers. Here, *k*NN and deepNN are trained using L only, while the SVM and CRT ensemble classifiers are trained using U_s as well as L in SB2-trans.

some datasets, the best accuracy was obtained with *k*NN, rather than SVM or CRT. (As an example, please refer to the accuracies achieved for the Gla and Pim datasets in Fig. 9(a) and for the Eco dataset in Fig. 9(b).) Furthermore, it should be mentioned that for the Gla and Veh datasets, the neural networks did not work, which means that the selection of the optimal parameters for the networks is sensitive to application. From these considerations, it should be mentioned that, in general, the classification performance of the SVM (and CRT) ensemble classifier can be improved using the proposed feature-transformation method when selecting U_s based on the multi-views extracted from the feature set.

In summary, from the above observations, it can be noted that SB2-trans works better when compared to SB2-decom, at least for the classification of handwritten digits and certain kinds of UCI data. This excellence seems to originate from the fact that, in SB2-trans, the confident values for selection can be measured more efficiently from the transformed multiple views, rather than measuring them from the simply decomposed subsets or the original feature set. However, the problem of figuring out how they can help each other when computing the values remains to be analyzed.

5. Conclusions

In an effort to efficiently select useful unlabeled data using multiple views, transformation-based selection (TBS) and decomposition-based selection (DBS) strategies were considered and empirically compared in this paper.

More specifically, experiments were performed using an SVM (and decision tree) ensemble classifier learned with the TBS and DBS algorithms for handwritten digit data and UCI benchmark data; in TBS, the feature set was transformed into multiple views using the well-known feature selection/extraction functions, such as stochastic neighbor embedding (SNE) and principal component analysis (PCA), while, in DBS, multiple views were derived using the traditional decomposition methods, including the random (RD) and mutual information (MI) based methods.

The experimental results demonstrated that both the TBS and DBS methods can compensate for the shortcomings of the traditional selection strategies. In particular, the results demonstrated that when the data is transformed efficiently, unlabeled data selection can benefit more from TBS than DBS for certain kinds of object classification. This benefit seems to be grounded in the fact that the two views, which are obtained using SNE and PCA respectively, help each other when computing the confidence levels in TBS.

Although the ensemble classifier trained using TBS can be improved in terms of classification accuracy, more study should be carried out. A significant task is to find an optimal or near-optimal combination of mapping functions for the task at hand to measure correct confidence labels. In addition, the strategy has

limitations in details that support its technical reliability, and the experiments performed were limited. Thus, a theoretical investigation of this empirical behavior deserves further study.

Acknowledgments

The authors would like to thank the anonymous Referees for their valuable comments, which improved the quality and readability of the paper.

References

[1] A. Asuncion, D.J. Newman, UCI Machine Learning Repository, University of California, School of Information and Computer Science, Irvine, CA, 2007.
[2] F.R. Bach, G.R.G. Lanckriet, M.I. Jordan, Multiple kernel learning, conic duality, and the SMO algorithm, in: Proceedings of the 21st International Conference on Machine Learning, ACM, 6, 2004.
[3] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, in: Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT 98), Madison, WI (1998) 92–100.
[4] C.C. Chang, C.J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intel. Syst. Technol. 2 (3) (2011) 27:1–27:27.
[5] O. Chapelle, B. Schölkopf, A. Zien, Semi-Supervised Learning, The MIT Press, MA, 2006.
[6] M. Chen, K.Q. Weinberger, Q. Chen, Automatic feature decomposition for single view co-training, in: Proceedings of the 28th International Conference on Machine Learning (ICML-11), Bellevue, WA, 2011, pp. 953–960.
[7] I. Dagan, S.P. Engelson, Committee-based sampling for training probabilistic classifiers, in: Proceedings of the 12th International Conference on Machine Learning, Morgan Kaufmann, Tahoe City, CA, 1995, pp. 150–157.
[8] C.K. Dagli, S. Rajaram, T.S. Huang, Utilizing information theoretic diversity for SVM active learning, in: Proceedings of 18th International Conference on Pattern Recognition (ICPR'06), Hong Kong, China, 2006, pp. 506–511.
[9] T. Diethe, D.R. Hardoon, J. Shawe-Taylor, Multiview fisher discriminant analysis, in: Proceedings of NIPS Workshop on Learning from Multiple Sources, 2008.
[10] T. Diethe, D.R. Hardoon, J. Shawe-Taylor, Constructing nonlinear discriminants from multiple data views, in: Proceedings of European Conference on Machine Learning Knowledge Databases, 2010, pp. 328–343.
[11] P. Donmez, J.G. Carbonell, P.N. Bennett, Dual strategy active learning, in: Proceedings of 18th European Conference on Machine Learning (ECML'07), Warsaw, Poland, 2007, pp. 116–127.
[12] J. Du, C.X. Ling, Z.H. Zhou, When does cotraining work in real data? IEEE Trans. Knowl. Data Eng. 23 (5) (2011) 788–799.
[13] R.P.W. Duin, P. Juszczak, D. de Ridder, P. Paclik, E. Pękalska, D.M.J. Tax, PRTools 4: a matlab toolbox for pattern recognition, Technical Report, Delft University of Technology, The Netherlands, 2004. Can also be referred (as of July 2010) to http://prtools.org/.
[14] F. Feger, I. Koprinska, Co-training using RBI nets and different feature splits, in: Proceedings of the 2006 International Joint Conference on Neural Network, 2006, pp. 1878–1885. Vancouver, BC, Canada
[15] D. Hardoon, S. Szedmak, J. Shawe-Taylor, Canonical correlation analysis; an overview with application to learning methods, Technical Report CSD-TR-03-02, Department of Computer Sciences, University of University of London, Royal Holloway, 2003. Available at http://eprints.soton.ac.uk/259225/1/tech_report03.pdf
[16] T.K. Ho, The random subspace method for constructing decision forests, IEEE Trans. Pattern Anal. Mach. Intell. 20 (8) (1998) 832–844.
[17] P.J. Huber, Robust Statistics, John Wiley & Sons, New York, NY, 1981.
[18] A. Kumar, H. Daumé III, A co-training approach for multi-view spectral clustering, in: Proceedings of 28th International Conference on Machine learning (ICML-11), New York, NY, 2011, pp. 393–400.
[19] T.B. Le, S.W. Kim, Modified criterion to select useful unlabeled data for improving semi-supervised support vector machines, Pattern Recog. Lett. 60–61 (2015) 48–56.
[20] T.B. Le, S.W. Kim, On measuring confidence levels using multiple views of feature set for useful unlabeled data selection, Neurocomputing 173 (3) (2016) 1589–1601.
[21] T.B. Le, S.W. Kim, Feature decomposition using mutual information for semi-boost learning, in: Proceedings of the 30th International Technical Conference on Circuits/System, Computers and Communications (ITC-CSCC2015), Seoul, Korea, 2015, pp. 631–632.
[22] G.Z. Li, T.Y. Liu, Feature selection for co-training, J. Shanghai Univ. 12 (2008) 47–51, doi:10.1007/s11741-008-0110-2.
[23] P.K. Mallapragada, R. Jin, A.K. Jain, Y. Liu, Semiboost: Boosting for semi–supervised learning, IEEE Trans. Pattern Anal. Mach. Intell. 31 (11) (2009) 2000–2014.
[24] T. Masters, Practical Neural Network Recipes in C++, Academic Press, San Diego, CA, 1993.
[25] D. McClosky, E. Charniak, M. Johnson, When is self-training effective for parsing? in: Proceedings of 22nd International Conference on Computational Linguistics (Coling 2008), 2008, pp. 561–568. Manchester, UK
[26] S.M. Kakade, D.P. Foster, Multi-view regression via canonical correlation analysis, in: Proceedings of Computational Learning Theory (COLT 2007), 2007, pp. 561–568.
[27] T. Reitmaier, B. Sick, Active classifier training with the 3DS strategy, in: Proceedings of IEEE Symposium on Computational Intelligence and Data Mining (CIDM 11), 2011, pp. 88–95. France, Paris
[28] T. Reitmaier, B. Sick, Let us know your decision: Pool-based active training of a generative classifier with the selection strategy 4DS, Inf. Sci. 230 (2013) 106–131.
[29] D.S. Rosenberg, P.L. Bartlett, The rademacher complexity of co-regularized kernel classes, J. Mach. Learn. Res. (2007) 396–403.
[30] C. Rosenberg, M. Hebert, H. Schneiderman, Semi-supervised self-training of object detection models, in: Proceedings of 7th IEEE Workshop on Applications of Computer Vision / IEEE Workshop on Motion and Video Computing (WACV/MOTION'05), Breckenridge, CO, 2005, pp. 29–36.
[31] J. Rupnik, J. Shawe-Taylor, Multi-view canonical correlation analysis, in: Proceedings of Slovenian KDD Conf. Data Mining Data Warehouses, 2010, pp. 1–4.
[32] M. Seeger, Learning with labeled and unlabeled data, Technical Report 161327, Institute for Adaptive and Neural Computation, University of Edinburgh, Edinburgh, England, 2002. Available at http://infoscience.epfl.ch/record/161327/files/review.pdf
[33] B. Settles, Active Learning Literature Survey, Computer Sciences Technical Report 1648, University of Wisconsin-Madison, 2010.
[34] A. Sharma, A. Kumar, H. Daume III, D.W. Jacobs, Generalized multiview analysis: A discriminative latent space, in: Proceedings of IEEE Conf. Computer Vision Pattern Recognition, 2012, pp. 2160–2167.
[35] A. Singh, R. Nowak, X. Zhu, D. Koller, D. Schuurmans, Y. Bengio, L. Bottou, Unlabeled data: Now it helps, now it doesn't, in: Advances in Neural Information Processing Systems (NIPS), The MIT Press, London, 2008, pp. 1513–1520.
[36] S. Sun, F. Jin, W. Tu, View construction for multi-view semi-supervised learning, in: Advances in Neural Networks (ISNN 2011), 2011, pp. 595–601.
[37] L.J.P. van der Maaten, Accelerating t-SNE using tree-based algorithms, J. Mach. Learn. Res. 15 (2014) 3221–3245.
[38] T. Xia, D. Tao, T. Mei, Y. Zhang, Multiview spectral embedding, IEEE Trans. Syst. Man Cybern. 40 (6) (2010) 1438–1446.
[39] C. Xu, D. Tao, C. Xu, A survey on multi-view learning, 2013. ArXiv preprint arXiv:1304.5634.
[40] D. Yarowsky, Unsupervised word sense disambiguation rivaling supervised methods, in: Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics (ACL'95), Cambridge, MA, 1995, pp. 189–196.
[41] W. Zhang, Q. Zheng, Tsfs: a novel algorithm for single view co-training, in: Proceedings of the 2009 International Joint Conference on Computational Sciences and Optimization - Volume 01, IEEE Computer Society, Washington DC, USA, 2009, pp. 492–496.
[42] M.L. Zhang, Z.H. Zhou, Cotrade: confident co-training with data editing, IEEE Trans. Syst. Man Cybern., B 41 (6) (2011) 1612–1626.
[43] X. Zhu, A.B. Goldberg, Introduction to Semi-Supervised Learning, Morgan & Claypool, San Rafael, CA, 2009.

Thanh-Binh Le received his B.S. degree in Computer Science and Engineering from The University of Pedagogy – HCMC, Vietnam, in 2009, and the ME and the Ph.D. degrees from Myongji University, Yongin, Korea in 2012 and 2016, respectively, in Computer Engineering. Currently, he is a postdoctoral researcher of University College Dublin, Ireland. His research interests include Pattern Recognition and Machine Learning.

Sugwon Hong received BS in physics at Seoul National University in 1979, MS and Ph.D. in Computer Science at North Carolina State Univ. in 1988, 1992 respectively. His employment experience included Korea Institute of Science and Technology (KIST), Korea Energy Economics Institute (KEEI), SK Energy Ltd. and Electronic and Telecommunication Research Institute (ETRI). Currently he is a professor at Dept. of Computer Engineering, Myongji University since 1995. His major research fields are network protocol, algorithm, and security.

Sang-Woon Kim received the BE degree from Hankook Aviation University, Gyeonggi, Korea in 1978, and the ME and the PhD degrees from Yonsei University, Seoul, Korea in 1980 and 1988, respectively, both in Electronic Engineering. In 1989, he joined the Department of Computer Science and Engineering, Myongji University, Korea and is currently a Full Professor there. His research interests include Statistical Pattern Recognition, Machine Learning, and Avatar Communications in Virtual Worlds. He is the author or coauthor of 47 regular papers and 13 books. He is a Senior Member of the IEEE and a member of the IEIE.