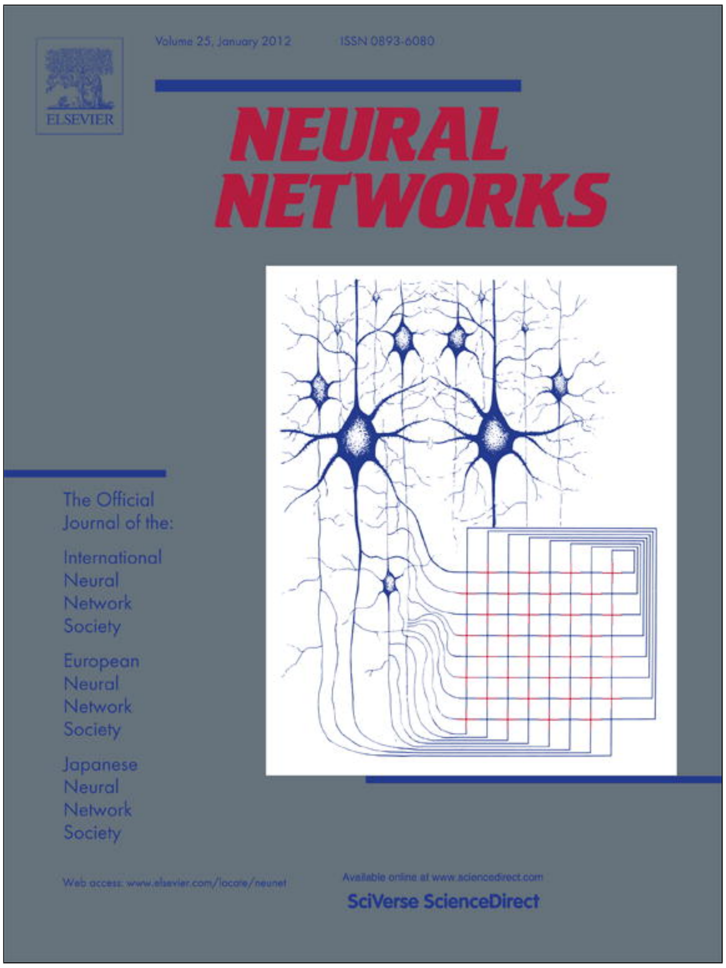
Volume 25, January 2012
ISSN 0893-6080
ELSEVIER
NEURAL NETWORKS
The Official Journal of the:
International Neural Network Society
European Neural Network Society
Japanese Neural Network Society
Web access: www.elsevier.com/locate/neunet
Available online at www.sciencedirect.com
SciVerse ScienceDirect

Neural Networks 25 (2012) 57–69

Contents lists available at SciVerse ScienceDirect

Neural Networks

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

An RSVM based two-teachers–one-student semi-supervised learning algorithm

Chien-Chung Chang*, Hsing-Kuo Pao, Yuh-Jye Lee

Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, 10607, Taiwan

ARTICLE INFO

Article history:
Received 2 June 2010
Received in revised form 17 February 2011
Accepted 27 June 2011

Keywords:
Co-training
Consensus training
Multi-view
Reduced set
Semi-supervised learning
Support vector machines

ABSTRACT

Based on the reduced SVM, we propose a multi-view algorithm, two-teachers–one-student, for semi-supervised learning. With RSVM, different from typical multi-view methods, reduced sets suggest different *views* in the represented kernel feature space rather than in the input space. No label information is necessary when we select reduced sets, and this makes applying RSVM to SSL possible. Our algorithm blends the concepts of co-training and consensus training. Through co-training, the classifiers generated by two views can "teach" the third classifier from the remaining view to learn, and this process is performed for each choice of teachers–student combination. By consensus training, predictions from more than one view can give us higher confidence for labeling unlabeled data. The results show that the proposed *2T1S* achieves high cross-validation accuracy, even compared to the training with all the label information available.

1. Introduction

Semi-supervised learning (SSL) has been one of the most active areas of the learning community in recent years. Beyond what the supervised learning can offer, many real applications need to deal with both labeled and unlabeled data simultaneously.[1] Usually, the amount of labeled data is insufficient and obtaining it is expensive. In contrast, unlabeled data is abundant and easy to collect. For example, we may need to categorize a number of web documents, but only a few of them may be correctly labeled. In another example, determining the functions of biological strings is expensive, and only a small portion of them have been studied (labeled) to date. SSL can help researchers deal with these kinds of problems because it takes advantage of knowing two kinds of data; (1) it uses labeled data to identify the decision boundary between data with different labels; and (2) it uses unlabeled data to determine the data's density, i.e., the data *metric*.

Among the various SSL algorithms that have been proposed, the multi-view approach is one of the most widely used. It splits data attributes into several attribute subsets and each subset of attributes is called a *view*. Combining the information got from each view will improve the performance on the supervised learning task. In the *co-training* algorithm (Blum & Mitchell, 1998), classifiers of different views learn about the decision boundaries from each other. Based on this concept, a number of variants have been developed, e.g., the *tri-training* algorithm (Zhou & Li, 2005). On the other hand, the classifiers of different views can be combined to form an *ensemble* classifier with a high level of confidence. We call this approach *consensus training*.

In this paper, we propose a *two-teachers-one-student* (*2T1S*) method for SSL. The method is a multi-view approach which blends the concepts of co-training and consensus training. Based on the reduced support vector machine (RSVM) (Lee & Huang, 2007; Lee & Mangasarian, 2001a), different from other existing approaches, our method *selects multi-view in the represented kernel feature space rather than in the input space*. In the feature space, we build a classifier according to each different views, with limited labeled data set. Then, based on two of the three classifiers (the teachers), some unlabeled data are marked if the teachers form a consensus answer, and those data are considered as the newly acquired labeled set for training the remaining classifier (the student). We apply the above "teaching" work to all classifier combinations, namely, three choices for the case of two teachers and one student. Ideally, most data points are successfully and correctly labeled as if we have additional labeled data for training in the next run. Clearly, if we do not consider the time cost, the combination of teachers and students can be generalized to the set of more than three views. The whole process is run iteratively and alternately until some stopping criteria are satisfied.

The proposed method is based on RSVM (Lee & Huang, 2007; Lee & Mangasarian, 2001a). In supervised learning, the RSVM is proposed to overcome some major difficulties of the typical support vector machines (SVMs) that confront large data problems due to dealing with a fully dense nonlinear kernel matrix. RSVM

* Corresponding author. 886 2 2733 3141x7323.
E-mail addresses: D9115009@mail.ntust.edu.tw, ccchang@cute.edu.tw (C.-C. Chang), pao@mail.ntust.edu.tw (H.-K. Pao), yuh-jye@mail.ntust.edu.tw (Y.-J. Lee).

[1] In fact, the input of a regular supervised classification actually takes a labeled data set and several fresh data without the class information for prediction, which can be considered as an SSL problem where a *transductive* learning method can be the solution.

0893-6080/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2011.06.019

replaces the full kernel matrix with a smaller rectangular kernel matrix. This rectangular kernel matrix is a low-rank approximation to the full kernel matrix and is generated by a uniform random subset, named *reduced set*. Conceptually, choosing reduced sets means choosing partial attribute sets (views) in the *represented kernel feature space*.[2] Therefore, our approach will not be restricted by the number of data attributes. That is, the proposed multi-view method may deal with a data set even it consists of only a few attributes.

To make our method to have better prediction result, the selected reduced sets (views) had better not "too similar" to each other. Traditionally, given the class information, researchers assume the conditional independence between different views (Zhu, 2005a). In the language of *generative modeling*, they assume that different views are generated independently given the class label. Our approach differs from those methods such as co-training algorithms in that we choose views that are not *linearly dependent* on each other. Working on the RSVM framework, the reduced set is more *representative* if there is a high degree of *dissimilarity* among the data points in the set. We use the IRSVM (Lee, Lo, & Huang, 2003), which is a kind of RSVM algorithm, to select the representative reduced sets from the entire data set (both labeled and unlabeled data) as different views. We need to emphasize that *label information is not required in the selection process of the reduced set*. That makes RSVM perfect for the SSL problems which largely need the help from unlabeled data.

Experiments show that our method's performance is superior to that of other SSL methods. The experiments include comparison of training and prediction using only a limited labeled set and the full labeled set; and comparison of our method's performance with that of other methods. We also use some synthesized data sets to illustrate the effectiveness of our approach on a set with particular properties. Before discussing our method in detail, we introduce the notations used in this work.

Notations and problem setting

By convention, we let $\mathbf{v}$ denote a column vector and $\mathbf{v}'$ denote a row vector. For an SSL problem, we consider an input data set $\mathcal{D}$ of size m, which consists of ℓ labeled points and u unlabeled points. The labeled part is the set

$$\mathcal{D}_L := \{(\mathbf{x}^1, y_1), \ldots, (\mathbf{x}^i, y_i), \ldots, (\mathbf{x}^\ell, y_\ell)\} \subseteq \mathcal{X} \times \mathbb{R},$$

where $\mathcal{X} \subseteq \mathbb{R}^n$ is in the original input space and each pair $(\mathbf{x}^i, y_i)$ is an observation $\mathbf{x}^i = (x_1^i, x_2^i, \ldots, x_n^i) \in \mathcal{X}$ with its response or class label y_i. The unlabeled part is the set

$$\mathcal{D}_U := \{\mathbf{x}^{\ell+1}, \ldots, \mathbf{x}^{(\ell+u)=m}\} \subseteq \mathcal{X}.$$

In most cases, we are interested in the SSL problem when $\ell \ll u$.

Let $A \in \mathbb{R}^{m\times n}$ be the data matrix of input attributes; and let each row of A, denoted by A_i, represent the observation $\mathbf{x}^i$. A reduced set is denoted by $\tilde{A} \in \mathbb{R}^{\tilde{m}\times n}$, where $\tilde{m}$ represents the number of reduced points in $\tilde{A}$. For $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{n\times l}$, the kernel $K(A, B)$ maps $\mathbb{R}^{m\times n} \times \mathbb{R}^{n\times l}$ into $\mathbb{R}^{m\times l}$. In particular, if $\mathbf{x}$ and $\mathbf{y}$ are column vectors in $\mathbb{R}^n$, then, $K(\mathbf{x}', \mathbf{y})$ is a real number, $K(A, \mathbf{x})$ is a *column* vector in $\mathbb{R}^m$, and $K(A, A)$ is an $m \times m$ matrix. For the labeled set, $Y = (y_1, \ldots, y_\ell)' \in \{-1, 1\}^\ell$ is the column vector of the corresponding responses in the case of a binary classification problem. The whole process of our method, *2T1S*, works on a represented kernel feature space (functional space) $\mathcal{F}_{\mathcal{B}}$, which is spanned by the full data bases $\mathcal{B} = \{k(\cdot, A_i)\}_{i=1}^m$. The original input space $\mathcal{X}$ is mapped into $\mathcal{F}_{\mathcal{B}}$ via the feature map $\Phi_{\mathcal{B}} : \mathcal{X} \subseteq \mathbb{R}^n \mapsto \mathcal{F}_{\mathcal{B}} \subseteq \mathbb{R}^m$ given by

$$\mathbf{x} \mapsto \Phi_{\mathcal{B}}(\mathbf{x}) := (k(\mathbf{x}, A_1), k(\mathbf{x}, A_2), \ldots, k(\mathbf{x}, A_m)),$$

where the value of $k(\mathbf{x}, A_i),\ i = 1, 2, \ldots, m$ represents the similarity between the data points $\mathbf{x}$ and $\mathbf{x}^i$. A view $\mathcal{V}$ is defined as a subset of data points, that is

$$\mathcal{V} = \{\tilde{A}_1, \tilde{A}_2, \ldots, \tilde{A}_{\tilde{m}}\},$$

which is used to select a subset of bases $\tilde{\mathcal{B}} = \{k(\cdot, \tilde{A}_1), k(\cdot, \tilde{A}_2), \ldots, k(\cdot, \tilde{A}_{\tilde{m}})\}$ from the full data bases $\mathcal{B} = \{k(\cdot, A_j)\}_{j=1}^m$ to build a separating surface prior to training.

In this work, we evaluate the proposed model by two measures, the prediction accuracy on unlabeled data called *training set accuracy*, and the prediction accuracy on fresh unseen data called *test set accuracy*.

The remainder of the paper is organized as follows. In Section 2, we review related works on SSL, and compare our approach with other SSL models. In Section 3, we introduce the framework of our method, including RSVM and the proposed *2T1S* algorithm. We also explain why our method can solve SSL problems effectively. Section 4 describes the numerical experiments and details the results. Section 5 contains some concluding remarks.

2. Previous work

SSL takes advantage of both labeled and unlabeled data to improve prediction performance. The technique has been widely used in a number of applications. For instance, Dong and Bhanu (2005) proposed a new active concept learning algorithm that, combines SSL with a model selection method for image retrieval. Other applications in text mining, bioinformatics, and computer vision are presented in Chapelle, Schölkopf, and Zien (2006) and Zhu (2005a). According to conventional categorization, SSL approaches can be divided into four categories (Chapelle et al., 2006; Zhu, 2005a): *low-density separation* methods, *graph-based* methods, methods for *changing the representation*, and *co-training* methods. Low-density separation methods try to find the decision boundary in the low-density area of data. The transductive support vector machine (TSVM) method (Bennett & Demiriz, 1999) belongs to this category. When the number of labeled points is small compared to the number of unlabeled points and the data points are distributed in clusters, TSVM performs better than SVMs because it utilizes the additional unlabeled data. Although high computational complexity of TSVM is a major drawback, some methods have been proposed to deal with large data sets (Collobert, Sinz, Weston, & Bottou, 2006).

Graph-based methods use a graph to describe a data set. Each node in a graph represents a data point and an edge represents the relationship between a pair of data points. Then, a graph cut algorithm is implemented to find the decision boundary (Blum & Chawla, 2001). Theoretically, finding a cut can be formulated by a soft version, which minimizes an energy function comprised of the cost on labeled data and a regularization term (on all data) (Kolmogorov & Zabin, 2004). An overview of graph-based methods and their applications is provided in Zhu (2005b). One limitation of most typical graph-based methods is that they have only *transductive ability* instead of *inductive ability*, i.e., they only focus on labeling unlabeled data. Without global function, the labeling of new unseen data must be performed by another supervised mechanism. To deal with this problem, some approaches such as (Belkin, Niyogi, & Sindhwani, 2006; Zhao, 2006) were proposed. Another problem is that the traditional graph-based methods is usually sensitive to outliers. Wang and Zhang (2007) proposed a robust self-tuning graph-based SSL method, named (RS^{3}L), which is capable of dealing with outliers.

Methods that change representation are based on the distribution of the input attributes of both labeled and unlabeled data. They try to find an appropriate metric to describe the relationships between data points, and use that metric to find a representation of

[2] One reduced set can decide one view, or be separated into several different views, to be discussed later.

all the data in a new space. The class information is then plugged in the new space to determine the decision boundary. Isomap (Tenenbaum, de Silva, & Langford, 2000), which belongs to the category of *manifold learning*, is a candidate to find such a metric. The metric in this case is defined by the *geodesic distance* between each pair of data points. Principle Component Analysis is another example, which finds the data representation in a low-dimensional space to be with the lowest *reconstruction error* (Alpaydin, 2004; Oliveira, Cozman, & Cohen, 2005). In principle, given a representation in the low-dimensional space, any supervised learning method, such as SVM, can be used to find the labels for both unlabeled data and unseen data.

Finally, the co-training algorithm (Blum & Mitchell, 1998) splits data attributes into several subsets. It is assumed that the attribute subsets are conditionally independent, given the label information. Each subset plays a view and is *sufficient* to learn a classifier. We can therefore use the prediction from one view to help other views learn the label information of unlabeled data. Our method is closely related to the co-training approach. However, in our framework, a reduced set represents a view in the represented kernel feature space. This is one of the major differences between our approach and other co-training methods. Another difference is that, instead of assuming conditional independence between different views, we assume that views are linearly independent of each other. We also study different choices of views in order to select the best co-training combinations, which we discuss later in the paper.

Various methods developed recently are inspired by the co-training scheme. Abdel Hady, Schwenker, and Palm (2010) combined a tree-structured approach and the co-training method to deal with multi-class problems. The *tri-training* algorithm proposed by Zhou and Li (2005) also belongs to the co-training methods. The algorithm starts by using different training sets to build three classifiers, which are considered as distinct views. Then they select training sets from the original labeled set (*the unlabeled part is not included*) via bootstrap sampling. Each classifier is iteratively regenerated by the updated training set, which consists of the original labeled set and the newly estimated labeled points derived by the other two classifiers. The final refined classifiers predict the labels of the test data by a majority voting mechanism.

Many SSL methods assume that class information is a hidden or latent variable, and try to find such information in unlabeled data; e.g., an SSL approach for probabilistic RBF network was proposed by Constantinopoulos and Likas (2008). Following this interpretation, we can regard our approach as an EM or co-EM like procedure (Dempster, Laird, & Rubin, 1977), which means that we alternately use one part of data, with some label information to build a classifier and to label the unlabeled data on another part, then we can use the estimated labeled data to retrain new classifiers in the next run. We need to emphasize that when we build the classifier based on one part of data, we use *both labeled and unlabeled data*. That differs from other EM or co-EM procedures, which use only the labeled data to build classifiers. Thanks to the formulation of RSVM, the label information is not necessary when we compile a reduced set. Hence we may use both labeled and unlabeled data to generate the reduced set, and use labeled data points to test if the generated classifier fits into our constraints. We explain the framework in detail in the next section.

3. *2T1S* approach for SSL

Our method is built on an alternate labeling and training procedure. Given the initial labeled data, we try to label the remaining unlabeled data and use both of labeled and the guessed labeled data for training in the next run. Being a multi-view approach, our method is built on an RSVM, where each reduced set serves as a view in the represented kernel feature space for SSL.[3]

3.1. *RSVM and reduced sets for multi-view learning*

For supervised learning problems, the SVM is one of the most promising algorithms. Taking advantage of the so-called *kernel trick*, the nonlinear SVM classifier is formulated as follows:

$$f(\mathbf{x}) = \sum_{j=1}^{m} u_j k(\mathbf{x}, \mathbf{x}^j) + b, \tag{1}$$

where $k(\mathbf{x}, \mathbf{x}^j)$ is a kernel function that represents the inner product of the images of $\mathbf{x}$ and $\mathbf{x}^j$ in the feature space under a certain nonlinear mapping that we do not need to know explicitly. For convenience, we use the terms "kernel function" and "basis function" interchangeably in this paper. A kernel matrix $K(A, A)$ is defined as $K(A, A)_{ij} = k(A_i, A_j)$, which records all the pairwise inner products (*similarities*) of instances in the represented kernel feature space. The nonlinear SVM classifier is a linear combination of the basis functions, $\{1\} \cup \{k(\cdot, A_j)\}_{j=1}^{m}$. For the linear SVM, the kernel function is defined as $k(\mathbf{x}, \mathbf{z}) = \mathbf{x}'\mathbf{z}$ and $K(A, A) = AA'$. In this paper, we use the radial basis function (RBF) kernel, defined as

$$k(\mathbf{x}, \mathbf{z}) = e^{-\mu\|\mathbf{x}-\mathbf{z}\|_2^2}, \tag{2}$$

where μ is the *width* parameter. A kernel with larger value of μ tends to fit the training data better; however, it may lead to *overfitting*. The coefficients u_j and b in Eq. (1) are determined by solving a quadratic programming problem (Burges, 1998; Vapnik, 1995) or an unconstrained minimization problem (Lee & Mangasarian, 2001b).

Solving the problems with large amounts of data is computationally difficult because it is necessary to deal with a fully dense nonlinear kernel matrix in the optimization problem. To resolve the difficulty, some authors have proposed applying low-rank approximation to the full kernel matrix (Smola & Schölkopf, 2000; Williams & Seeger, 2000). As an alternative, the reduced support vector machine (RSVM) was proposed in Lee and Mangasarian (2001a). RSVM's operations can be divided into two steps. First, it randomly selects a small subset of bases $\tilde{\mathcal{B}} = \{k(\cdot, \tilde{A}_1), k(\cdot, \tilde{A}_2), \ldots, k(\cdot, \tilde{A}_{\tilde{m}})\}$ from the full[4] data bases $\mathcal{B} = \{k(\cdot, A_j)\}_{j=1}^{m}$ to build a separating surface prior to training. This reduced model is formulated as

$$f(\mathbf{x}) = \sum_{j=1}^{\tilde{m}} \tilde{u}_j k(\mathbf{x}, \tilde{A}_j) + b. \tag{3}$$

In contrast to conventional SVMs, RSVM replaces the fully dense square kernel matrix $K(A, A)$ with a small rectangular kernel matrix $K(A, \tilde{A})$, which is used in the nonlinear SVM formulation to avoid the above-mentioned computational difficulty. In the second step, RSVM determines the best coefficients $\tilde{u}_j$ and b in Eq. (3) by solving the unconstrained minimization problem. It considers the entire data set, so the separating surface (3) will adapt to all the data. Hence, even though RSVM only uses a small portion of the kernel bases, it can still retain most of the relevant pattern information in the entire training set. A statistical theory that supports RSVM is discussed in Lee and Huang (2007).

Next, we discuss the roles of the reduced sets as different views in our multi-view algorithm. Ideally, to be effective as a

[3] We give only a brief description of RSVM. Please refer to (Lee & Mangasarian, 2001a) for all the details.

[4] It includes both of the labeled and unlabeled data in SSL. Also, no class information is necessary for this construction.

set of kernel bases, the selected kernel functions should not be "too similar" to each other; or, more rigorously, there should be "some degree" of linear independence between them. For a regular supervised learning problem, a reduced set with a higher degree of linear independence between its elements ensures a better classification result. Similarly, when more than one view is involved in an SSL problem, we prefer a view not to be "similar" to another view. In this work, given a reduced set, we can use it as a single view; or we select a subset of the whole set as a single view and the whole reduced set is separated into a few different views.[5] Based on this design, we select views (or a few subsets of a reduced set) with as much linear independence between them as possible. The views with more linear independence are *less likely* to have a uniform predicted result; therefore, they give a result of high confidence when they agree.

We choose a reduced set with dissimilar reduced points. By doing that, we ensure all the reduced sets are also dissimilar to each other. There are various algorithms for selecting a representative reduced set with dissimilar elements, e.g., those proposed in Chien, Chang, and Lee (2010) and Lee et al. (2003). In our SSL application, based on the result of IRSVM (Lee et al., 2003), we obtain a set of multi-view partners or reduced kernel matrices that are *linearly independent* of each other. Note that our view selection (reduced set building) procedure considers *both* labeled and unlabeled data points. It is different from the design of tri-training, where only labeled data is considered during the sampling process and the sampled sets are used for generating the initial classifiers. By including unlabeled data in the view selection procedure, we expect the performance of our method to be superior to that of tri-training. When labeled set is in a limited size, the proposed method should give a more stable result than that of tri-training. The experiment results, reported in Section 4, support our intuition.

3.2. Incremental RSVM algorithm

As mentioned above, the nonlinear RSVM classifier is a linear combination of the basis functions $\{1\} \cup \{k(\cdot, \tilde A_1), k(\cdot, \tilde A_2), \dots, k(\cdot, \tilde A_{\tilde m})\}$. The reduced set is more *representative* if there is a high degree of *dissimilarity* among its. Based on the intuition, the incremental reduced support vector machine (IRSVM) was proposed in Lee et al. (2003). We adopt the IRSVM algorithm, an incremental forward selection style algorithm, to generate various views for our *2T1S* algorithm. It sequentially adds a kernel function to the current basis function set only when the function is *dissimilar* to the current set.

The method starts with a very small reduced set $\tilde A$, typically a size of two. A new data point A_i will only be added to the current reduced set when the extra information carried in the vector $K(A, A_i)$ with respect to the column space of $K(A, \tilde A)$ is greater than a certain positive threshold $\delta > 0$. This can be achieved by solving a least squares problem, which is defined by

$$\min_{\beta \in \mathbb{R}^{\tilde m}} \left\| \tilde K \beta - K(A, A_i) \right\|_2^2, \tag{4}$$

where $\beta \in \mathbb{R}^{\tilde m}$ is a free vector variable; $\tilde K = K(A, \tilde A) \in \mathbb{R}^{m \times \tilde m}$ is the reduced kernel matrix generated by the current reduced set; and $\tilde K \beta \in \mathbb{R}^m$ is a linear combination of the functions $\{K(A, \tilde A_j)\}_{j=1}^{\tilde m}$ that represents the column space of $K(A, \tilde A)$. According to the first order optimality condition (Mangasarian, 1994), finding the optimal solution β^* of the unconstrained minimization problem in (4) is equivalent to solving a system of normal equations:

$$\tilde K' \tilde K \beta = \tilde K' K(A, A_i). \tag{5}$$

Algorithm 1: The IRSVM Algorithm

```
Input:
  A training data matrix $A \in \mathbb{R}^{m \times n}$.
  A threshold $\delta > 0$.
Output:
  A reduced set $\tilde A_{final}$.
  A discriminant model $f(\mathbf{x})$.
Randomly select a very small subset matrix $\tilde A_0 \in \mathbb{R}^{\tilde m \times n}$ from $A$, say
  $\tilde m = 2$, as an initial reduced set.
Generate the reduced kernel matrix $K(A, \tilde A_0)$.
$\tilde A_{new} \leftarrow \tilde A_0$.
repeat
    Choose a point $A_i \in A \setminus \tilde A_{new}$
    $r_i \leftarrow \|K(A, \tilde A_{new})\beta^* - K(A, A_i)\|_2$
    if $r_i > \delta$ then
        $\tilde A_{new} \leftarrow \tilde A_{new} \cup A_i$
    end
until no more point $A_i$ could be added into $\tilde A_{new}$
Construct an RSVM classifier $f(\mathbf{x})$ using the current reduced set
  $\tilde A_{new}$.
$\tilde A_{final} \leftarrow \tilde A_{new}$.
Return $\tilde A_{final}$ and $f(\mathbf{x})$.
```

If the columns of the rectangular kernel matrix generated by the initial reduced set are linearly independent, the IRSVM algorithm will retain the independence property throughout the whole process, so that the least squares problem (4) has a unique solution β^*,

$$\beta^* = (\tilde K' \tilde K)^{-1} \tilde K' K(A, A_i). \tag{6}$$

The distance r from $K(A, A_i)$ to the column space of $\tilde K$ is the square root of the optimal value of (4). It is computed by

$$r = \left\| \tilde K \beta^* - K(A, A_i) \right\|_2. \tag{7}$$

The squared distance can be written in the form $r^2 = (I - P)K(A, A_i)$, where $P = \tilde K(\tilde K' \tilde K)^{-1} \tilde K'$ is the projection matrix of $\mathbb{R}^m$ onto the column space of $\tilde K$. Hence, r^2 can be a measure of the extra information introduced by $K(A, A_i)$ with respect to current reduced kernel $K(A, \tilde A)$. The steps of the IRSVM algorithm are detailed in Algorithm 1. We describe the procedure of applying IRSVM in our *2T1S* algorithm in the next subsection.

3.3. View selection

As mentioned in Section 3.1, an RSVM classifier can be represented as a linear combination of the *selected kernel functions* for the corresponding randomly selected reduced set. To meet our requirements, the selected kernel functions should have low similarity, i.e., there should be high mutual (linear) independence between them. Below, we describe our mechanism for generating three views (or three subsets of reduced set),[6] which will take turns to play the roles of teacher and student in our *2T1S* SSL algorithm. The choice of views can be very flexible (Chien et al., 2010; Lee et al., 2003). In this work, we use IRSVM (Lee et al., 2003) to generate all three views because it guarantees *dissimilar* basis functions (mentioned in Section 3.2) as the *representatives*. We repeat the IRSVM procedure until some stopping criteria are satisfied. In this paper, we stop the algorithm when we have enough reduced points to form a candidate set. We then divide the set into three parts, each of which plays the role of a view in the *2T1S* algorithm.

The detailed procedure for generating the three reduced sets is as follows. Suppose we want a view (a subset of reduced set) whose size is equal to $\tilde m$, we repeat the IRSVM procedure

[5] A subset of a reduced set of course can also be called another reduced set.

[6] Again, a number of more than three views should be easy to generalize.

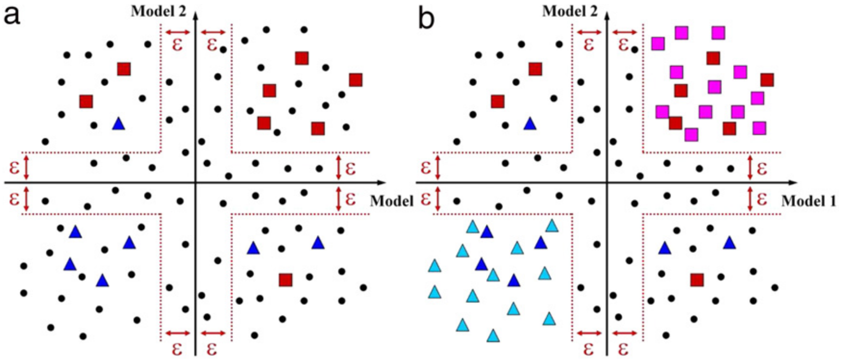

Fig. 1. The visualization of data points (a) before, and (b) after a consensus training, with a consensus level ε. Two axes represent the predictions from two models. The upper-right and the bottom-left corners mean the agreement between the two models. A red square indicates positive data, a dark blue triangle indicates negative data, a black dot indicates unlabeled data, a pink square indicates estimated positive data, and a light blue triangle indicates estimated negative data. We only label the unlabeled data when both models agree the answer (in the upper-right and bottom-left corners). A larger ε implies a larger agreement between two models, but may introduce fewer estimated data points. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

until the number of reduced points is equal to $3\tilde{m}$. Let $\tilde{A}_{final}$ be the set of $3\tilde{m}$ reduced points, which we split into three subsets (views) with size $\tilde{m}$ through a round-robin (interleaving) partition method called $\tilde{B}$, $\tilde{C}$, and $\tilde{D}$. Based on Step 5 of the IRSVM algorithm (Algorithm 1), it is clear that the three subsets of bases, $\{k(\cdot, \tilde{B}_j)\}_{j=1}^{\tilde{m}}$, $\{k(\cdot, \tilde{C}_j)\}_{j=1}^{\tilde{m}}$, and $\{k(\cdot, \tilde{D}_j)\}_{j=1}^{\tilde{m}}$ are mutually exclusive. Since the column spaces of $K(A, \tilde{B})$, $K(A, \tilde{C})$, and $K(A, \tilde{D})$, denoted by $CS(K(A, \tilde{B}))$, $CS(K(A, \tilde{C}))$, and $CS(K(A, \tilde{D}))$, are spanned by the above three mutually exclusive subsets of basis functions, respectively. Thus these hypothesis spaces are orthogonal, and for any two distinct views $\mathcal{V}_i, \mathcal{V}_j \in \{\tilde{B}, \tilde{C}, \tilde{D}\}$, we have

$$CS(K(A, \mathcal{V}_i)) \cap CS(K(A, \mathcal{V}_j)) = \{\mathbf{0}\}. \tag{8}$$

Therefore, all the columns of the kernel matrices generated by these three views are linearly independent of each other. Intuitively, views selected in this manner are likely to suggest labels "independently" for unlabeled data; hence, there is a high level of confidence when they agree on an answer.

To satisfy the *smoothness assumption* (Chapelle et al., 2006) addressed in many SSL work, we may also select a reduced set with the k-means clustering algorithm (Hartigan & Wong, 1979; MacQueen, 1967) heuristically, since we assume that neighboring points in a high-density region are likely to have identical labels. In this sense, it is interesting to construct one view by using *cluster centroids* as the *representative subset* of the entire data set. We can select two views from IRSVM and the centroid view from k-means, as combining two different view selection methods for another way of constructing the views. This method will also be studied in our experiments.

3.4. The 2T1S algorithm = co-training + consensus training

In this subsection, we introduce our proposed *2T1S* algorithm for iterative labeling and training. Our approach is inspired in part by the well-known co-training method (Blum & Mitchell, 1998) for SSL. The co-training method can help us to teach other views to label the unlabeled data, if the two views are not very similar to each other. In addition, more views can help us obtain a better estimation result. That is, we will have more confidence if more views are provided for relatively "independent" predictions. This is called *consensus training*. We combine these two methods, co-training and consensus training, to form our *2T1S* algorithm. In the labeling step, two teachers based on two views are consulted to find a confident result, which is used to label, i.e., to teach the third view (the student) to guess the labels of the unlabeled data. This step is performed on each teachers–student combination. At the end of the process, we have the guessed label information for many of the unlabeled data. We can be a little conservative about the labeling for the unlabeled data by choosing a positive "confidence" value $\varepsilon > 0$, which is called the *consensus level*. That is, we prefer the predicted result to be at least away from the "twilight area" between ε and $-\varepsilon$ (Fig. 1). In each step of building the consensus, the value of ε might be decreased by some rate, such as 0.9, when there is no point reaching a consensus and we still need more estimated labeled points for training. We repeat the "teaching" step until the student classifier cannot "learn" any more from the two teacher classifiers. That is, we repeat the above procedure to label the unlabeled data until the labeling procedure makes no more, or very few, changes. We then use all the labeled data (both the original labeled data and the estimated labeled data) to build the final classifier, which is used for making predictions on the *unseen* data. We describe our *2T1S* algorithm formally in Algorithm 2.

Remark. Below, we summarize the main points discussed in this section, and explain the major differences between our approach and other SSL methods.

1. The whole process of the proposed method, *2T1S*, works on a represented kernel feature space rather than in the input space. Different from other multi-view methods, the views in *2T1S* are defined as subsets of data points rather than subsets of attributes. Compared with other multi-view methods assume that the views are conditionally independent, we prefer different views to be more linearly independent from each other. Hence, we adopt the IRSVM algorithm to select a representative reduced set with dissimilar elements and partition it into three reduced subsets, which play the role of views. Based on the RSVM formulation, the reduced set is used to generate a smaller rectangular kernel matrix to replace the full kernel matrix. The linear independence between pairs of rectangular kernel matrices in the represented kernel feature space is treated as the degree of dissimilarity between different reduced sets.
2. SSL is especially interesting when the unlabeled part is much larger than the labeled part. This is the case when we cannot afford the additional expense of labeling unlabeled data. On the other hand, RSVM is also useful when the amount of data is large. As noted in Lee and Huang (2007) and Lee and Mangasarian (2001a), the reduced set dramatically reduces the amount of SVM computation, without much loss of accuracy from the prediction based on the full matrix. This implies that *our approach should be even more effective when a larger data set or unlabeled data set is involved.*
3. When we generate a reduced set, the label information is *not* necessary. Hence, we select the reduced sets from the entire data set (both labeled and unlabeled data), and only use the

Algorithm 2: The *2T1S* Algorithm

```
Input:
  Initial labeled data $\mathcal{D}_L=\{(\mathbf{x}^i, y_i)\}_{i=1}^{\ell}, \mathbf{x}^i \in \mathbb{R}^n, y_i \in \{-1, 1\}$.
  Initial unlabeled data $\mathcal{D}_U=\{(\mathbf{x}^i)\}_{i=\ell+1}^{m=\ell+u}, \mathbf{x}^i \in \mathbb{R}^n$.
  Initial classifiers $f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x})$.
  A consensus level $0 \leq \varepsilon \leq 1$.
Output:
  The final discriminant model $f(\mathbf{x})$.
$\mathcal{D}_{L_i} \leftarrow \mathcal{D}_L, i = 1, \ldots, 3$.
$iter \leftarrow 1$.
$\mathcal{D}_L^{(0)} \leftarrow \mathcal{D}_L$.
repeat
    for $i \leftarrow 1$ to 3 do
        $u \leftarrow |\mathcal{D}_U|$
        for $j \leftarrow 1$ to $u$ do
            $t_1 \leftarrow mod(i-1, 3) + 1$
            $t_2 \leftarrow mod(i, 3) + 1$
            $s \leftarrow mod(i+1, 3) + 1$
            if $(f_{t_1}(\mathbf{x}^j) \geq \varepsilon$ and $f_{t_2}(\mathbf{x}^j) \geq \varepsilon)$ or
              $(f_{t_1}(\mathbf{x}^j) \leq -\varepsilon$ and $f_{t_2}(\mathbf{x}^j) \leq -\varepsilon)$
            then
                $\mathcal{D}_{L_s} \leftarrow \mathcal{D}_{L_s} \cup \mathbf{x}^j$
                $\mathcal{D}_L \leftarrow \mathcal{D}_L \cup \mathbf{x}^j$
                $\mathcal{D}_U \leftarrow \mathcal{D}_U \setminus \mathbf{x}^j$
            end
        end
        Retrain the classifier $f_s(\mathbf{x})$ with $\mathcal{D}_{L_s}$.
    end
    $\mathcal{D}_L^{(iter)} \leftarrow \mathcal{D}_L$.
    $iter \leftarrow iter + 1$.
until $\mathcal{D}_L^{(iter)} = \mathcal{D}_L^{(iter-1)}$
Construct an RSVM classifier $f(\mathbf{x})$ with the final labeled data set $\mathcal{D}_L$.
Return $f(\mathbf{x})$.
```

labeled part to confirm the effectiveness of the function built from the reduced set basis.

Our novel approach combines the RSVM and the IRSVM algorithms for co-training and consensus training. In the next section, we discuss the experiments, followed by some discussion.

4. Experiment results

To compare the performance of the *2T1S* algorithm with other SSL approaches, we test it on two synthetic data sets and nine publicly available data sets (Asuncion & Newman, 2007). Table 1 summarizes the statistics of the data sets. While most of the data sets are for regular supervised learning, in each data set, we choose part of the labeled data to hide the label information to obtain unlabeled data. We study the performance with different percentages of labeled data with their labels kept for semi-supervised training.

As mentioned in the previous section, we use the IRSVM (Lee et al., 2003) procedure in our experiments to generate three views for our teachers–student combination. As an alternative, we also select two views derived by IRSVM, and one view (the cluster centroids) obtained from the *k*-means clustering result in the *2T1S* algorithm. The sizes of view (measured in number of data points) used in all the experiments are also summarized in Table 1. We use Gaussian kernel functions for RSVM and IRSVM in all the experiments. The initial value of ε, i.e., the margin in RSVM, is set at 1. Recall that we reduce the value of ε by 0.9 when the existing unlabeled data cannot be labeled any further. Besides, we adopt the nested uniform design (UD) model selection method (Huang, Lee, Lin, & Huang, 2007) to select the penalty parameter $\mathcal{C}$ and the Gaussian kernel width parameter μ for RSVM.

Table 1
The statistics of the data sets used in the experiments.

Data set description			
Data set	Instance	Feature	View size (in data points)
Five-group	250	2	25
Checkerboard	1000	2	50
Tic-tac-toe	958	9	60
Vote	435	16	40
Wdbc	569	30	60
Hypothyroid	3163	25	70
Ionosphere	351	34	35
Australian	690	14	50
Pima Indians	768	8	50
German	1000	24	70
BUPA Liver	345	6	35

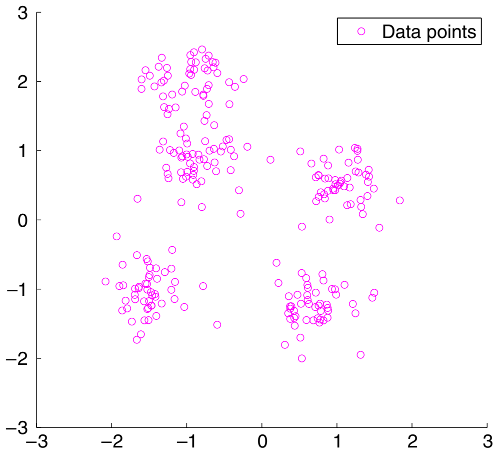

Fig. 2. The distribution of a synthetic five-group data set. Among them, the top cluster and the middle cluster are relatively closer to each other.

In the following evaluation, we use the terms *training set accuracy* and *transductive accuracy* interchangeably to denote the classification accuracy on the training set, which consists of the estimated labeled examples from the unlabeled set $\mathcal{D}_U$ and the original given labeled examples from $\mathcal{D}_L$ (class information included). The term *labeled set accuracy* denotes the classification accuracy on the original given labeled set $\mathcal{D}_L$, and *test set accuracy* or *inductive accuracy* denotes the classification accuracy on the fresh test set, which was not seen before the training commenced.

We conducted three sets of experiments in our evaluation. The first set evaluated our algorithm's performance on some synthetic data sets. It is also used to visually demonstrate the effectiveness of the proposed *2T1S* as a technique for SSL. The second set assessed the algorithm's performance on the real-world data sets mentioned earlier; and the third set compared our method's performance with that of other SSL methods, such as the co-training algorithm and the tri-training algorithm. The results demonstrate that, in terms of prediction power, our method is superior to the compared methods on many real-world data sets.

4.1. Five-group data set

For the reason of visualizing the effect of our method on *transductive ability*, we first tested our *2T1S* algorithm on a five-group synthetic data set. It is comprised of five clusters and there are 50 points in each cluster, as shown in Fig. 2. As we can see, most data groups have clear boundaries except between the top and the middle groups. In Fig. 3(a1), the amount of labeled data is very limited, identified by two red squares and two blue triangles that are located in separate clusters. After performing SSL, we obtain the result shown in Fig. 3(b1). The result agrees with our expectation, accurately predicting the labels for *all* unlabeled

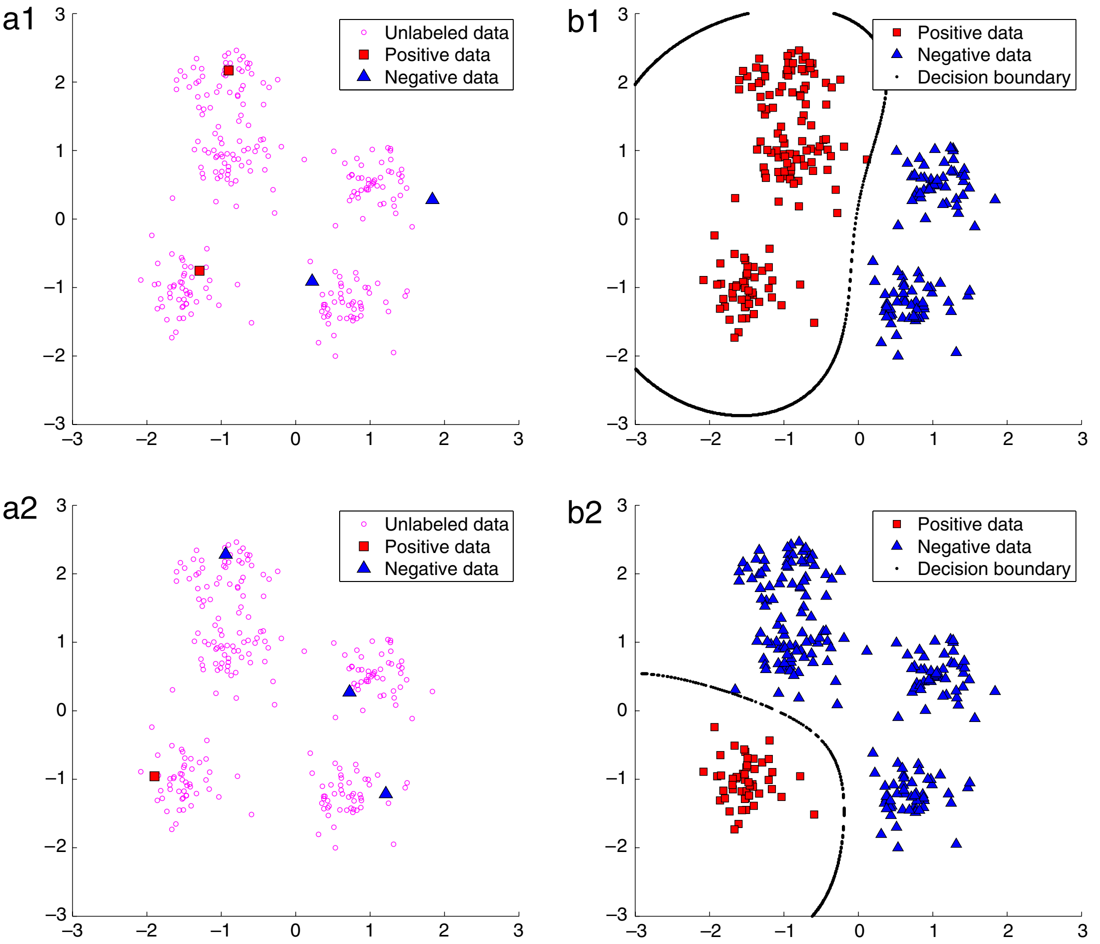

Fig. 3. The synthetic data set used to demonstrate the effectiveness of our method for SSL. We assume that there are only very limited labeled data and each located in a somewhat separate cluster. (a1) and (a2) are two input data sets and (b1) and (b2) are their labeling result respectively. We observe that the decision boundaries tend to go through the low-density areas between groups.

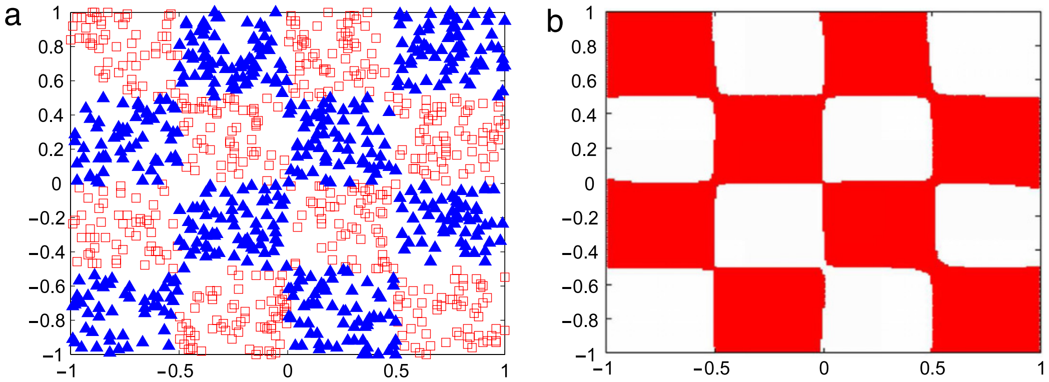

Fig. 4. (a) The 1000-point training data set in $\mathbb{R}^2$ distributed on sixteen black and white squares of a checkerboard. The positive points are denoted by red hollow squares and the negative points are denoted by blue triangles. (b) The prediction result from the supervised RSVM, given *full* (100%) label information. The trial is run 30 times and it shows the best test accuracy of 98.05% on a 39 601-point test set. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

data points. We conducted another experiment on the input with different choices of labeled sets (Fig. 3(a2)), which also yields a perfect classification result (100%), as shown in Fig. 3(b2). The proposed algorithm is capable of identifying appropriate labels for all the data. Clearly, different initial labeled points will produce different decision boundaries. More importantly, we observe that the decision boundaries tend to go through the low-density areas between groups. That is similar to the results obtained by most SSL methods. It is also worth knowing that the points of two clusters, which have the "closest relationship", always have the same label.

4.2. *Checkerboard data set*

To further assess the performance of the *2T1S* algorithm on *inductive ability*, we tested it on another synthetic data set, namely, the checkerboard data set (Ho & Kleinberg, 1996; Kaufman, 1999) shown in Fig. 4(a). The data set consists of 1000 points randomly

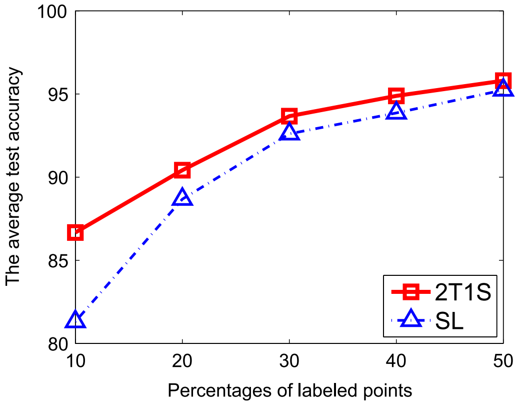

Fig. 5. The comparison results of the average test accuracy between *2T1S* and the pure supervised learning scheme while using the same percentage of labeled points as training set after a series of 30 trials on a 39 601-point test set.

distributed in an $\mathbb{R}^2$, 4 × 4 checkerboard. Fig. 4(b) shows the prediction result from the supervised RSVM, given *full* (100%) label information.

In the first experiment, we randomly selected 20%, 30%, 40%, or 50% from the data set as original labeled data points and treated the remainder as unlabeled points; then, we performed *2T1S* given all data (labeled and unlabeled parts), and the pure supervised learning scheme given only the labeled data. The above series were run 30 trials to obtain average results.

We compared the performance of *2T1S* with that of the pure supervised learning scheme, RSVM on the same sized reduced set. The results are shown in Fig. 5. According to the results, given different percentages of labeled data, the classifiers generated by *2T1S* always have better average test accuracy than the classifiers built by the pure supervised learning scheme with the same labeled data as training set. Here we just illustrate the case where only 30% of the data are labeled, as shown in Fig. 6(a). Fig. 6(b) and Fig. 6(c) show that the unlabeled data points are helpful. Fig. 6(b) shows a poor pattern that approximates a checkerboard obtained by the pure supervised RSVM classifier when the input is 30% of points from the entire training set. The best test set accuracy of this classifier is 93.60% on a test set of 39 601 points.[7] In contrast, our *2T1S* method yields a more accurate representation of the checkerboard depicted in Fig. 6(c), with a best accuracy rate of 97.10% on the same test set. It is very close to the prediction accuracy of 98.05% from supervised RSVM with all the training data available, as shown in Fig. 4(b). Based on the results, we have reason to believe that the estimated label information for the unlabeled points can be used in the training process to construct a nonlinear RSVM classifier that yields better test accuracy than those constructed without using unlabeled data.

Moreover, given the same labeled data, we claim that our *2T1S* and the pure supervised learning scheme (without unlabeled information) produce significantly different expected accuracy on a new example. To verify our viewpoint, we ran the paired t-test to compare the final classifier generated by *2T1S* with the classifier built by the pure supervised learning scheme given the same labeled data. We first randomly selected 10%, 20%, 30%, 40%, or 50% from the data set as original labeled data points and treated the remainder as unlabeled points. Next, we performed *2T1S* given all data (labeled and unlabeled parts) and the pure supervised learning scheme given only the labeled data. We ran the above

[7] It is generated on the side for the evaluation.

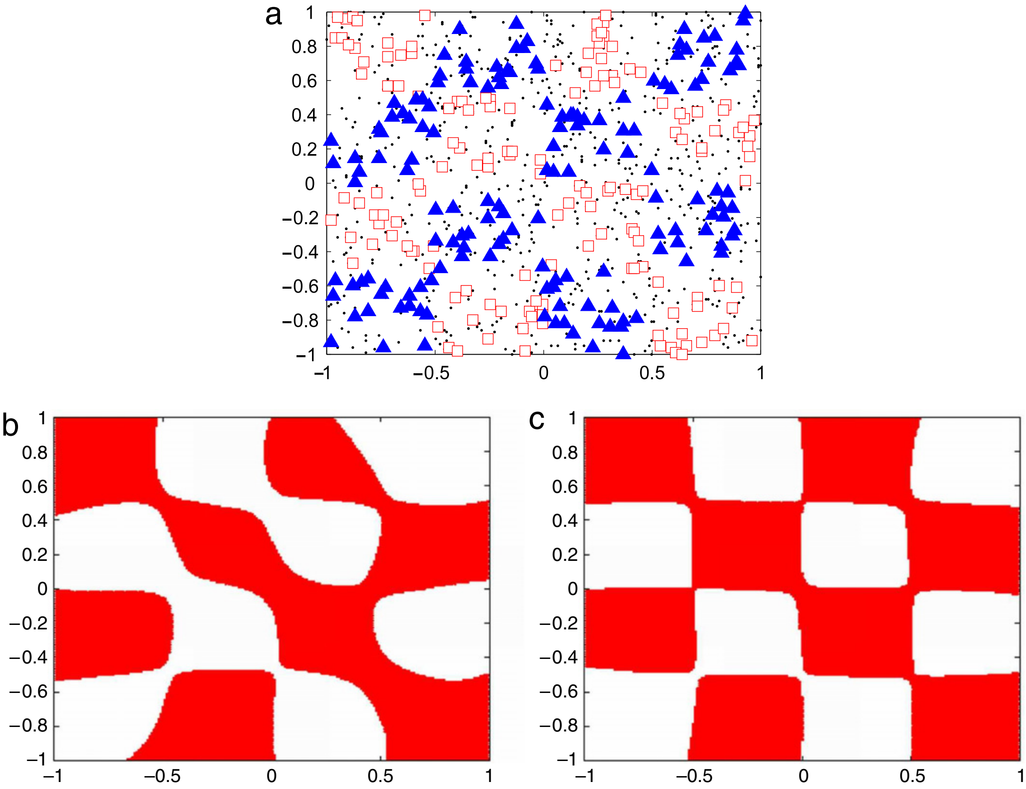

Fig. 6. The checkerboard result: (a) 30% of the points in the entire 1000-point training set, selected at random. The positive points are denoted by red hollow squares and the negative points are denoted by blue triangles; (b) the result from the supervised RSVM, with the accuracy rate of 93.60% given only the labeled points, and (c) the result of *2T1S* given the same labeled set (the remaining points are unlabeled data), with the accuracy rate of 97.10%. The results are the best ones after a series of 30 trials on a 39 601-point test set. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

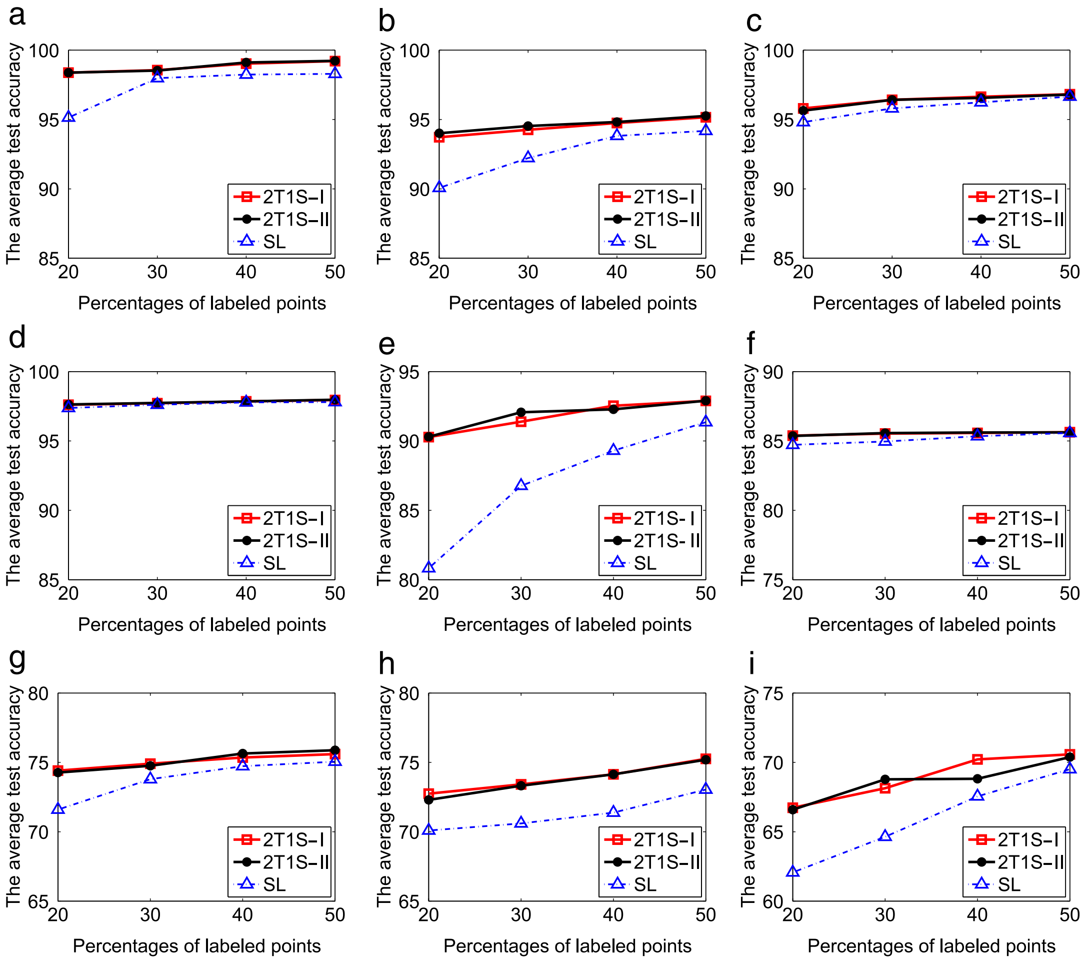

Fig. 7. The comparison results of the average test accuracy between *2T1S* and the pure supervised learning scheme using the same percentage of labeled points as training set after a series of 30 ten-fold cross-validation trials on nine public data sets, (a) Tic-tac-toe (b) Vote (c) Wdbc (d) Hypothyroid (e) Ionosphere (f) Australian (g) Pima Indians (h) German (i) BUPA Liver. There are three methods to compare with: 2T1S-I denotes the *2T1S* with 3 IRSVM views; 2T1S-II denotes the *2T1S* with 2 IRSVM views and 1 centroid view; and the last, SL denotes the pure supervised learning scheme using only the labeled data for training.

Table 2
The *p*-value of paired *t*-test on checkerboard data set that compares the *2T1S* to the pure supervised learning scheme with the same percentage of labeled points as training set.

The *p*-value of paired *t*-test	
Ratio of labeled points (%)	*p*-value
10	7.6887e−03
20	1.9170e−02
30	2.0163e−02
40	2.7328e−02
50	4.8328e−02

series for 20 trials and obtained 20 test accuracy and the results were used for a paired *t*-test. Table 2 shows the numerical results and demonstrates that our method gives the result significantly different from the one derived from the pure supervised learning scheme.

4.3. UCI data sets

In the second set of experiments, we tested *2T1S* on nine UCI data sets (Asuncion & Newman, 2007). We ran ten-fold cross-validation 30 times on each data set. To evaluate the performance of *2T1S*, we compare with that of the pure supervised learning scheme under the same setting. For each fold, we randomly selected 20%, 30%, 40%, or 50% of the data points from the training set as labeled data and treated the remainder as unlabeled data.

Table 3
Ten-fold cross-validation results of the average test accuracy on nine public data sets when we use the supervised RSVM generated by full training points.

Ten-fold test set accuracy ± std (%)	
Data set	Accuracy
Tic-tac-toe	99.75 ± 0.20
Vote	95.34 ± 0.49
Wdbc	97.19 ± 0.25
Hypothyroid	98.08 ± 0.10
Ionosphere	94.16 ± 0.84
Australian	86.24 ± 0.47
Pima Indians	75.87 ± 0.76
German	77.13 ± 0.52
BUPA Liver	73.27 ± 0.80

We show the numerical results of *2T1S* and the comparisons to other supervised learning results from Tables 3–6, also in Fig. 7. First, Table 3 shows the average test accuracy of the supervised RSVM that uses *all* the labeled data points for training. Presumably, it is the best accuracy that we can obtain from SSL methods.

Table 4 details the average training accuracy and the average numbers of final labeled points for *2T1S*, based on ten-fold cross-validation. Note that *2T1S* may not label *all* the unlabeled data due to our conservative policy when there is a lack of confidence or there is no consensus among the multi-view or two teachers. Even so, the numerical results show that our approach could label most of the unlabeled data with high accuracy. Based on the limited

Table 4
Ten-fold cross-validation results of the average training accuracy and the average number of final labeled points on nine public data sets when we use the *2T1S* algorithm. The numbers in parentheses are the data sizes, copied from Table 1.

Ten-fold training set accuracy ± std (%)					
Ten-fold number of labeled points ± std					
Data set	Method	20%	30%	40%	50%
Tic-tac-toe (958)	I	98.70 ± 0.27	98.99 ± 0.39	99.56 ± 0.37	99.72 ± 0.31
		859.6 ± 1.20	859.2 ± 1.02	859.2 ± 0.81	859.1 ± 0.83
	II	98.72 ± 0.29	99.03 ± 0.41	99.64 ± 0.33	99.75 ± 0.31
		859.4 ± 1.16	859.2 ± 1.00	859.1 ± 1.11	859.6 ± 0.83
Vote (435)	I	95.73 ± 1.02	96.77 ± 0.56	97.36 ± 0.50	97.66 ± 0.43
		386.4 ± 1.58	386.1 ± 1.34	385.9 ± 1.30	386.4 ± 1.08
	II	96.07 ± 0.84	96.95 ± 0.66	97.46 ± 0.42	97.71 ± 0.41
		386.2 ± 1.75	386.1 ± 1.40	385.8 ± 1.18	386.3 ± 0.98
Wdbc (569)	I	97.56 ± 0.66	97.97 ± 0.40	98.05 ± 0.33	98.35 ± 0.31
		495.8 ± 4.28	496.3 ± 3.74	497.2 ± 3.64	498.3 ± 3.09
	II	97.64 ± 0.54	97.97 ± 0.46	98.08 ± 0.35	98.40 ± 0.31
		494.3 ± 3.84	496.7 ± 4.06	496.0 ± 2.59	497.1 ± 3.69
Hypothyroid (3163)	I	98.01 ± 0.25	98.19 ± 0.21	98.28 ± 0.18	98.41 ± 0.13
		2838.7 ± 2.31	2837.9 ± 1.93	2838.3 ± 1.83	2838.1 ± 1.90
	II	98.02 ± 0.25	98.20 ± 0.19	98.28 ± 0.17	98.39 ± 0.13
		2839.2 ± 2.23	2837.9 ± 1.97	2837.8 ± 2.37	2838.4 ± 1.70
Ionosphere (351)	I	93.08 ± 1.25	94.43 ± 1.22	95.45 ± 0.61	96.24 ± 0.90
		310.3 ± 1.47	310.2 ± 1.19	309.6 ± 1.64	310.0 ± 1.26
	II	92.81 ± 2.00	94.80 ± 0.87	95.60 ± 0.77	96.40 ± 0.67
		311.8 ± 1.63	311.5 ± 1.06	310.6 ± 1.55	310.5 ± 1.21
Australian (690)	I	87.22 ± 1.72	87.52 ± 1.50	87.57 ± 0.83	87.58 ± 0.79
		565.8 ± 49.35	564.5 ± 40.93	576.2 ± 28.33	583.2 ± 18.64
	II	86.93 ± 1.21	87.51 ± 1.16	87.44 ± 0.87	87.59 ± 0.72
		575.2 ± 32.65	570.1 ± 35.71	579.8 ± 27.27	586.2 ± 13.45
Pima Indians (768)	I	76.00 ± 0.69	76.59 ± 0.87	77.05 ± 0.59	77.57 ± 0.44
		682.7 ± 2.16	682.7 ± 1.66	682.4 ± 1.90	681.7 ± 1.77
	II	75.70 ± 0.99	76.45 ± 0.55	77.13 ± 0.31	77.38 ± 0.42
		684.9 ± 1.55	684.5 ± 1.54	684.2 ± 1.65	683.1 ± 2.21
German (1000)	I	74.68 ± 1.26	75.69 ± 1.06	76.56 ± 0.69	77.57 ± 0.60
		893.4 ± 1.58	892.8 ± 2.29	891.3 ± 1.54	891.7 ± 2.36
	II	74.35 ± 1.26	75.31 ± 1.16	76.42 ± 0.76	77.54 ± 0.58
		892.9 ± 2.06	893.2 ± 1.74	892.0 ± 2.11	891.1 ± 1.89
BUPA Liver (345)	I	70.80 ± 1.59	72.50 ± 1.66	73.94 ± 1.01	74.58 ± 0.91
		302.4 ± 2.23	301.9 ± 2.17	301.3 ± 1.55	302.1 ± 2.29
	II	69.65 ± 2.60	71.81 ± 1.76	73.21 ± 0.98	74.41 ± 0.87
		304.4 ± 2.14	303.9 ± 1.60	303.4 ± 1.43	302.7 ± 1.66

Method I: *2T1S* with 3 IRSVM views.
Method II: *2T1S* with 2 IRSVM views and 1 centroid view.

Table 5
The *p*-value of paired *t*-test on nine public data sets that compare the *2T1S* to the pure supervised learning scheme with the same percentage of labeled points as training set.

The *p*-value of paired *t*-test					
Data set	10%	20%	30%	40%	50%
Tic-tac-toe	1.6110e−06	1.4752e−05	6.7880e−04	9.2723e−07	1.4781e−06
Vote	1.1628e−02	4.0239e−05	8.2673e−04	6.7592e−03	1.0047e−02
Wdbc	4.2084e−03	1.1409e−04	2.1328e−04	5.6431e−04	6.0704e−03
Hypothyroid	3.6649e−03	7.6264e−03	9.3122e−03	1.7328e−02	3.8069e−02
Ionosphere	1.1962e−07	1.2798e−06	2.3424e−05	3.6566e−04	1.2766e−03
Australian	4.3463e−03	6.8507e−03	1.0451e−02	2.8118e−02	4.2275e−02
Pima Indians	1.5813e−03	1.5267e−03	1.8858e−02	7.9130e−03	7.7174e−03
German	5.4718e−03	1.6562e−09	2.9461e−09	1.5230e−08	2.0123e−08
BUPA Liver	8.1846e−07	1.2703e−04	6.1119e−03	1.2014e−03	2.2617e−02

but informative estimated labeled data as well as the original labeled data, we build the final classifier. The results show that the final classifier from *2T1S* has its test accuracy comparable to the test accuracy from supervised learning using the entire labeled set available for training. Note that the CPU times required to implement *2T1S* in this set of experiments are from 1.66 to 33.99 s. The results show that although our method needs to retrain iteratively, the time cost is still acceptable.

On the other hand, Fig. 7 demonstrates the comparison results of average test accuracy between the *2T1S* and the pure supervised learning scheme using the same percentages of labeled data, but without the unlabeled part. The test accuracy of the classifiers built by *2T1S*, with 20%, 30%, 40%, and 50% of the labeled data points available for training (also with the unlabeled part) are higher than the test accuracy of the pure supervised learning classifiers using only the labeled data points. That shows that

Table 6
The results of the labeled set accuracy from each individual view of the *2T1S* algorithm on nine public data sets. In which, all views are selected from the IRSVM.

Labeled set accuracy% of supervised learning with different views	⟨View1⟩ Accuracy				100%[a]
	[View2] Accuracy				
	(View3) Accuracy				
Data set	20%	30%	40%	50%	
Tic-tac-toe	⟨100.0⟩ [100.0] [100.0]	⟨100.0⟩ [100.0] [100.0]	⟨100.0⟩ [100.0] [100.0]	⟨100.0⟩ [100.0] [100.0]	100.0
Vote	⟨100.0⟩ [100.0] (100.0)	⟨100.0⟩ [100.0] (99.24)	⟨100.0⟩ [99.43] (100.0)	⟨100.0⟩ [100.0] (99.08)	96.30
Wdbc	⟨100.0⟩ [99.13] (99.13)	⟨99.42⟩ [99.42] (98.84)	⟨99.12⟩ [98.68] (98.68)	⟨99.30⟩ [99.30] (98.60)	97.53
Hypothyroid	⟨99.53⟩ [99.37] (99.37)	⟨99.26⟩ [99.16] (99.16)	⟨98.89⟩ [98.97] (98.97)	⟨98.86⟩ [98.86] (98.67)	98.46
Ionosphere	⟨100.0⟩ [100.0] (100.0)	⟨99.06⟩ [100.0] (100.0)	⟨99.29⟩ [98.58] (97.87)	⟨98.30⟩ [97.73] (97.16)	96.07
Australian	⟨90.65⟩ [90.65] (91.37)	⟨91.83⟩ [91.35] (90.87)	⟨90.61⟩ [90.61] (87.00)	⟨89.60⟩ [88.44] (89.02)	86.39
Pima Indians	⟨85.71⟩ [85.06] (84.42)	⟨83.12⟩ [82.25] (81.39)	⟨81.82⟩ [81.82] (81.49)	⟨83.07⟩ [82.03] (80.47)	77.36
German	⟨86.50⟩ [84.50] (86.00)	⟨84.33⟩ [84.00] (85.67)	⟨83.25⟩ [84.00] (83.00)	⟨81.40⟩ [81.80] (81.20)	79.62
BUPA Liver	⟨84.06⟩ [81.16] (82.61)	⟨79.81⟩ [81.73] (80.77)	⟨78.99⟩ [80.43] (78.99)	⟨78.61⟩ [79.19] (79.19)	76.59

[a] 100% means performing the RSVM with entire training labeled set.

the proposed *2T1S* does take advantage of using the unlabeled data information.

Next, we ran the paired t-test to compare the final classifier generated by *2T1S* with the classifier built by the supervised learning given the same labeled data (but without the unlabeled part). For each approach and each experiment, we first performed ten-fold cross-validation 20 times, then used these 20 average test accuracy to examine the paired t-test. The numerical results are shown in Table 5. These results demonstrate that our method is significant different from the pure supervised learning scheme with only labeled data.

It would be interesting to check if one view in *2T1S* would be sufficient to learn a good classifier. The prediction power of each view is shown in Table 6. We compare the classifiers generated by different views with the classifier built by the entire data set. In this test, *all views are selected from the IRSVM*. The numerical results show that the classifiers constructed by different views have similar labeled set accuracy to that of the classifier based on the entire labeled set. In other words, the selected views are sufficient to learn a classifier. In our design, two teachers decide the label, and the extra labeled result should provide useful information for meaningful retraining.

4.4. Comparison of 2T1S with co-training and tri-training algorithms

In the third set of experiments, we compared *2T1S* with co-training and tri-training algorithms (Zhou & Li, 2005) on nine UCI data sets (Asuncion & Newman, 2007). We adopted the experimental procedure described in Zhou and Li (2005). First, we randomly selected 25% of the data points in the data set as the test set and treated the remaining 75% as the training set. Then, we randomly selected 20%, 40%, 60%, or 80% of the data points in the training set as labeled data and treated the remainder as unlabeled data. We applied the *2T1S* algorithm three times for each ratio of labeled data points in order to train data points to obtain the average performance.

Fig. 8 demonstrates the comparison results of average test accuracy among the *2T1S*, co-training, and tri-training algorithms.[8] We only show the best results from the co-training and tri-training each with J4.8, BP neural networks, and Naïve Bayes as base learner in turn. The results show that, for most of the data sets, our RSVM based *2T1S* algorithm can predict the test set with higher accuracy than those from either co-training or tri-training.

5. Conclusion

We have proposed an RSVM based *2T1S* algorithm for semi-supervised learning. Considering the limited and expensive labeled data and the massive but cheaper unlabeled data simultaneously, the proposed *2T1S* method can achieve high accuracy rates on both transductive learning (measured by training accuracy) and inductive learning (measured by test accuracy). The *2T1S* algorithm is built on a framework of RSVM and IRSVM. The reduced set is used to build the views in the view selection process. Unlike other multi-view methods, *2T1S* selects views in the represented kernel feature space rather than in the input space. Moreover, instead of requiring conditional independence between views, our algorithm finds views that are linearly independent of each other. As a result, the predictions based on different views can have the result with high confidence by independent judgments.

[8] The numerical results for the co-training and tri-training algorithms are quoted from Zhou and Li (2005).

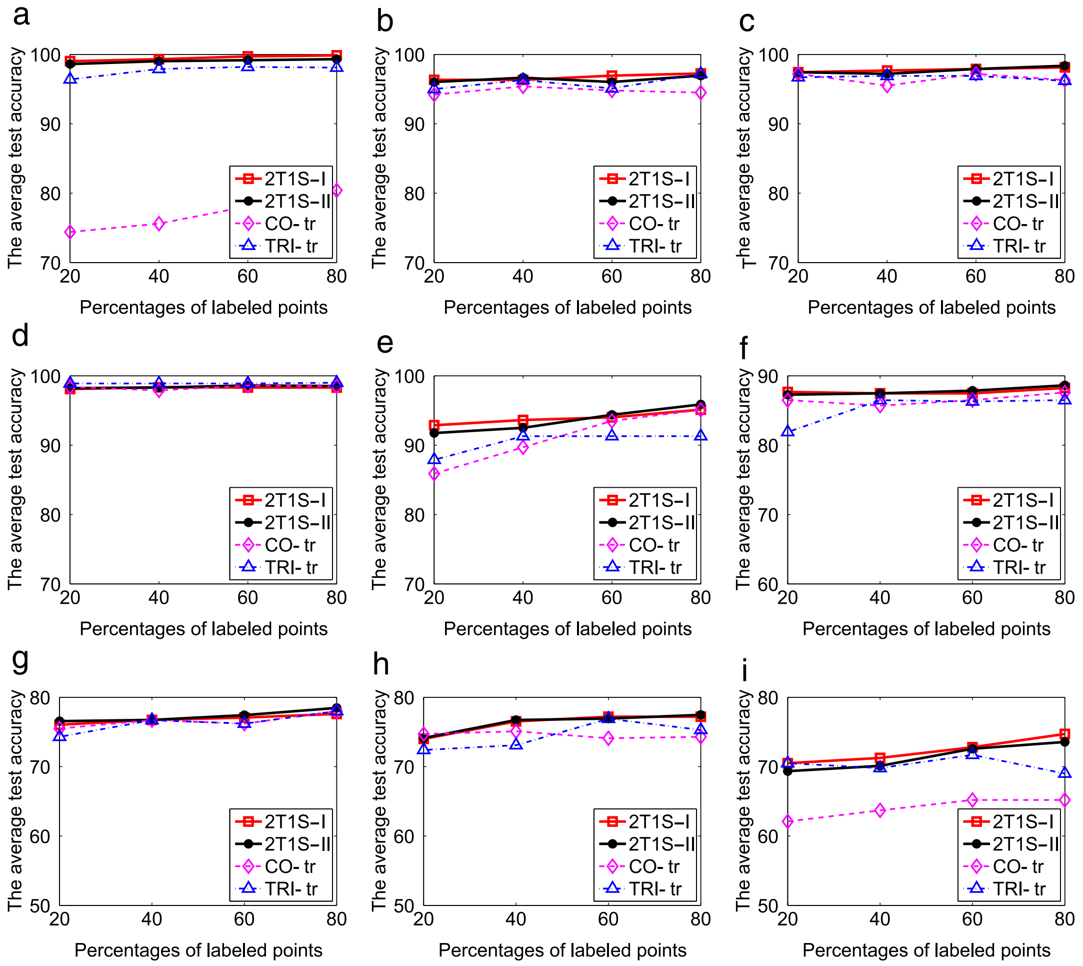

Fig. 8. The comparison results of the average test accuracy among the *2T1S*, co-training, and tri-training algorithms on nine publicly available data sets, using 20%, 40%, 60%, and 80% of the training set as labeled data: (a) Tic-tac-toe (b) Vote (c) Wdbc (d) Hypothyroid (e) Ionosphere (f) Australian (g) Pima Indians (h) German (i) BUPA Liver. Four methods discussed here: 2T1S-I: the one from *2T1S* with 3 IRSVM views; 2T1S-II: the one from *2T1S* with 2 IRSVM views and 1 centroid view; CO-tr: the best results from the co-training with J4.8, BP neural networks, and Naïve Bayes as base learner in turn; TRI-tr: the best results from the tri-training with J4.8, BP neural networks, and Naïve Bayes as base learner in turn.

As a multi-view approach, our method combines the concepts of co-training and consensus training. Co-training helps us label unlabeled data, while consensus training gives us sufficient confidence in the labeling process. We use two teachers for consensus training and one student as the co-training partner. The *2T1S* algorithm alternately labels the unlabeled data based on classifiers on hold and builds the classifiers based on the original labeled data, and the guessed labeled data obtained from previous classification result.

We evaluated the performance of *2T1S* on some synthesized and real-world data sets. The numerical results show that the algorithm uses only a small portion of the labeled data for training, yet it achieves comparable cross-validation accuracy to the algorithm that uses all the labeled data points. We also compared *2T1S* to the co-training and the tri-training algorithms, on nine UCI data sets. The experiment results show that, on most of the data sets, *2T1S* outperforms the other two SSL methods. Therefore, we expect that the *2T1S* algorithm can receive widely attentions from researchers in the SSL learning community.

References

Abdel Hady, M. F., Schwenker, F., & Palm, G. (2010). Semi-supervised learning for tree-structured ensembles of RBF networks with co-training. *Neural Networks*, *23*, 497–509.

Alpaydin, E. (2004). *Introduction to machine learning*. The MIT Press.

Asuncion, A., & Newman, D. (2007). UCI repository of machine learning databases. http://www.ics.uci.edu/~mlearn/MLRepository.html.

Belkin, M., Niyogi, P., & Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from examples. *Journal of Machine Learning Research*, *7*, 2399–2434.

Bennett, K., & Demiriz, A. (1999). Semi-supervised support vector machines. *Advances in Neural Information Processing Systems*, *11*, 368–374.

Blum, A., & Chawla, S. (2001). Learning from labeled and unlabeled data using graph mincuts. In *Proc. 18th international conf. on machine learning* (pp. 19–26). San Francisco, CA: Morgan Kaufmann.

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *COLT: proceedings of the workshop on computational learning theory* (pp. 92–100). Morgan Kaufmann Publishers.

Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, *2*, 121–167.

Chapelle, O., Schölkopf, B., & Zien, A. (Eds.). (2006). *Semi-supervised learning*. Cambridge, MA: MIT Press.

Chien, L.-J., Chang, C.-C., & Lee, Y.-J. (2010). Variant methods of reduced set selection for reduced support vector machines. *Journal of Information Science and Engineering*, *26*, 183–196.

Collobert, R., Sinz, F. H., Weston, J., & Bottou, L. (2006). Large scale transductive svms. *Journal of Machine Learning Research*, *7*, 1687–1712.

Constantinopoulos, C., & Likas, A. (2008). Semi-supervised and active learning with the probabilistic RBF classifier. *Neurocomputing*, *71*, 2489–2498.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, *39*, 1–38.

Dong, A., & Bhanu, B. (2005). Active concept learning in image databases. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *35*, 450–466.

Hartigan, J. A., & Wong, M. A. (1979). A *k*-means clustering algorithm. *Applied Statistics*, *28*, 100–108.

Ho, T. K., & Kleinberg, E. M. (1996). Building projectable classifiers of arbitrary complexity. In *Proceedings of the 13th international conference on pattern recognition. Vienna, Austria* (pp. 880–885). http://cm.bell-labs.com/who/tkh/pubs.html. Checker dataset at: ftp://ftp.cs.wisc.edu/math-prog/cpo-dataset/machine-learn/checker.

Huang, C.-M., Lee, Y.-J., Lin, D. K. J., & Huang, S.-Y. (2007). Model selection for support vector machines via uniform design. A special issue on Machine Learning and Robust Data Mining of Computational Statistics and Data Analysis. vol. 52 (pp. 335–346).

Kaufman, L. (1999). Solving the quadratic programming problem arising in support vector classification. In B. Schölkopf, C. J. C. Burges, & A. J. Smola (Eds.), *Advances in kernel methods—support vector learning* (pp. 147–167). MIT Press.

Kolmogorov, V., & Zabin, R. (2004). What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*, 147–159.

Lee, Y.-J., & Huang, S.-Y. (2007). Reduced support vector machines: a statistical theory. *IEEE Transactions on Neural Networks*, *18*, 1–13.

Lee, Y.-J., Lo, H.-Y., & Huang, S.-Y. (2003). Incremental reduced support vector machines. In *International conference on informatics, cybernetics and systems. ICICS 2003. Kaohsiung, Taiwan*.

Lee, Y.-J., & Mangasarian, O. L. (2001a). RSVM: reduced support vector machines. In *Proceedings of the first SIAM international conference on data mining*.

Lee, Y.-J., & Mangasarian, O. L. (2001b). SSVM: a smooth support vector machine. *Computational Optimization and Applications*, *20*, 5–22.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In L. M. L. Cam, & J. Neyman (Eds.), *Proceedings of the fifth berkeley symposium on mathematical statistics and probability, volume 1: statistics* (pp. 281–297). Berkeley, California: University of California Press.

Mangasarian, O. L. (1994). *Nonlinear programming*. Philadelphia, PA: SIAM.

Oliveira, C. S., Cozman, F. G., & Cohen, I. (2005). Splitting the unsupervised and supervised components of semi-supervised learning. In *Proc. of the 22nd ICML workshop on learning* (pp. 67–73).

Smola, A. J., & Schölkopf, B. (2000). Sparse greedy matrix approximation for machine learning. In *Proc. 17th international conf. on machine learning* (pp. 911–918). San Francisco, CA: Morgan Kaufmann.

Tenenbaum, J. B., de Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, *290*, 2319–2323.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. New York: Springer.

Wang, F., & Zhang, C. (2007). Robust self-tuning semi-supervised learning. *Neurocomputing*, *70*, 2931–2939.

Williams, C. K. I., & Seeger, M. (2000). Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems*, NIPS2000.

Zhao, H. (2006). Combining labeled and unlabeled data with graph embedding. *Neurocomputing*, *69*, 2385–2389.

Zhou, Z.-H., & Li, M. (2005). Tri-training: exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering*, *17*, 1529–1541.

Zhu, X. (2005a). Semi-supervised learning literature survey. *Technical report 1530*. Dept. of Computer Science. University of Wisconsin. Madison. http://www.cs.wisc.edu/~jerryzhu/pub/ssl_survey.pdf.

Zhu, X. (2005b). Semi-supervised learning with graphs. *CMU-LTI-05-192 Ph.D. dissertation*. Carnegie Mellon University. Pittsburgh, PA.