2010 International Conference on Technologies and Applications of Artificial Intelligence

A Passive-Aggressive Algorithm for Semi-supervised Learning

Chien-Chung Chang and Yuh-Jye Lee and Hsing-Kuo Pao
Department of Computer Science and Information Engineering
National Taiwan University of Science and Technology
Taipei, 10607, Taiwan
Email: {D9115009, yuh-jye, pao}@mail.ntust.edu.tw

***Abstract*—In this paper, we proposed a novel semi-supervised learning algorithm, named passive-aggressive semi-supervised learner, which consists of the concepts of passive-aggressive, down-weighting, and multi-view scheme. Our approach performs the labeling and training procedures iteratively. In labeling procedure, we use two views, known as teacher's classifiers for consensus training to obtain a set of guessed labeled points. In training procedure, we use the idea of down-weighting to retrain the third view, i.e., student's classifier by the given initial labeled and guessed labeled points. Based on the idea of passive-aggressive algorithm, we would also like the new retrained classifier to be held as near as possible to the original classifier produced by the initial labeled data. The experiment results showed that our method only uses a small portion of the labeled training data points, but its test accuracy is comparable to the pure supervised learning scheme that uses all the labeled data points for training.**

***Index Terms*—passive-aggressive; down-weighting; co-training; consensus training; incremental reduced support vector machine; multi-view; reduced set.**

I. Introduction

Beyond what the supervised learning can offer, many real applications need to deal with both labeled and unlabeled data simultaneously[1], such as text mining, bioinformatics, computer vision, and image retrieval [1], [2]. 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. Semi-supervised learning (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*.

In this paper, we proposed a novel SSL algorithm, named passive-aggressive semi-supervised learner (*PASL*). The proposed method combines the concepts of passive-aggressive (PA) algorithm [3] in part and down-weighting with our multi-view learning framework.

The PA algorithm [3] is a margin based learning scheme, which is often used for on-line learning. On one hand, the on-line PA algorithm modifies the current classifier to correctly classify the current example by updating the weight vector. On the other, the new classifier must remain as close as possible to the current classifier. Similar to the scenario of on-line learning, SSL uses both labeled and unlabeled data to improve prediction performance and simultaneously we would like the new generated SSL classifier to be held as *near* as possible to the original classifier produced by the labeled data. Based on this idea, the PA algorithm is used to deal with the SSL problem. Formally, in our method, we add the term $\frac{1}{2}\|\mathbf{w}-\mathbf{w}_L\|_2^2$ into the objective function of the standard SVMs, where $\mathbf{w}$ is the normal to the new classifier and $\mathbf{w}_L$ is the normal to the original classifier built for the labeled data.

In most SSL method, we need to directly or indirectly label the unlabeled data to form the final classifier. However, in most cases, we are not sure whether the guessed labeled data have the correct label information. Hence, to reduce the effect that the wrongly guessed labeled points join the training procedure, we consider how to separate the influence of the guessed labeled data from the effect of the given initial labeled points. This goal can be achieved by introducing the concept of down-weighting. In our approach, we give higher penalty weight to the initial labeled points if they are misclassified in training procedure. In contrast, we give lower penalty weight to the guessed labeled points.

Our approach is based on a multi-view framework. Among the various SSL algorithms that have been proposed, the multi-view method is one of the most widely used approach.

[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.

978-0-7695-4253-9/10 $26.00 © 2010 IEEE
DOI 10.1109/TAAI.2010.61

It splits data attributes into several attribute subsets, called *views*, to improve the learning performance. In the *co-training* algorithm [4], classifiers of different views learn about the decision boundaries from each other. 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*. Our scheme is based on these two concepts. Moreover, in contrast to existing approaches, we propose a method that selects multi-views in the feature space rather than in the input space, borrowing the language of the support vector machines (SVMs) [5]. Technically, we apply the RSVM algorithm [6], [7] to select different views to realize the proposed algorithm.

Under the *PASL*, given three views, the classification answers from two classifiers (two teachers) represent the consensus result, which is used to teach the third view (the student) to learn the labels for unlabeled data. This process is performed for each choice of teachers-student combination. After the student learns the data, the newly learned labeled data are added to the student's original labeled data set, as the set of guessed labeled data and they are included for training in the next step if it is part of the teachers' sets in the next step. The whole process is run iteratively and alternately until some stopping criteria are satisfied. Clearly, the combination of teachers and students can be generalized to the set of more than three views.

In principle, most co-training algorithms prefer views that are "not too similar" to each other, given the class information. Traditionally, given the class information, researchers assume the conditional independence between different views [1]. In the language of *generative modeling*, different views are generated independently given the class label. *PASL* differs from those co-training algorithms in that we choose views that are not *linearly dependent* on each other. Second, we choose teachers with consensus answers as the guessed labels for the next step of the iteration.

Another difference of proposed *PASL* is that the whole framework is based on RSVM [6], [7]. To overcome the computational complexity of SVMs, we usually rely on a small size approximation of the inner product matrix AA' or the kernel matrix $K(A, A)$ in the nonlinear case, given the data matrix A and its transpose A'. RSVM chooses several columns of the kernel matrix as the approximation, namely the inner products on the *reduced set*. Theoretically, choosing reduced sets means choosing partial attribute sets in the *feature space*. For SSL, we use RSVM to select different views in the feature space, not in the input space. That is, we treat the reduced sets as different views in the feature space. Initially, we use different reduced sets in turn as well as the limited labeled set as the data sets to build three classifiers that represent three different views. The key point is that *label information is not required in the selection process of the reduced set*. Then, based on the 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). Ideally, most data points are successfully labeled as if we have additional labeled data for training in the next run. We continue the above "teaching" work for all classifier combinations[2].

The experiments include comparison of training and prediction using either a limited labeled set or the full labeled set. 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. By the matrix formulation, let $A \in \mathbb{R}^{m\times n}$ be the input attribute set; and let each row of A, denoted by A_i, represent observation $\mathbf{x}^i$. We use the terms $< \mathbf{x}_i, \mathbf{x}_j >$ and $\mathbf{x}_i'\mathbf{x}_j$ interchangeably to denote the inner product of any two vectors $\mathbf{x}_i$ and $\mathbf{x}_j$. The p-norm of $\mathbf{x}$ will be denoted by $\|\mathbf{x}\|_p$. A column vector of ones of arbitrary dimension will be denoted by bold-face $\mathbf{1}$. The base of the natural logarithm will be denoted by e.

For an SSL problem, we consider an input data set $\mathcal{D}$, which consists of labeled and unlabeled data. The labeled part is the set $\mathcal{D}_L := \{(\mathbf{x}^1, y_1), \ldots, (\mathbf{x}^i, y_i), \ldots, (\mathbf{x}^\ell, y_\ell)\} \subseteq \mathbb{R}^n \times \mathbb{R}$, where each pair $(\mathbf{x}^i, y_i)$ is an observation $\mathbf{x}^i = (x_1^i, x_2^i, \ldots, x_n^i) \in \mathbb{R}^n$ with its response or class label y_i. The unlabeled part is the set $\mathcal{D}_U := \{\mathbf{x}^{\ell+1}, \ldots, \mathbf{x}^{(\ell+u)=m}\} \subseteq \mathbb{R}^n$. In most cases, we are interested in the SSL problem when $\ell \ll u$. 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. For the guessed labeled set, $\hat{Y} = (\hat{y}_1, \ldots, \hat{y}_g)' \in \{-1, 1\}^g$ is the column vector of the corresponding responses in the case of a binary classification problem. An input attribute subset in the space $\mathbb{R}^n$ is called a *view*. A classifier can work on such a view for prediction, and the results from many views can be combined by a "voting" scheme for multi-view learning. When a *feature* space is considered, e.g., for a problem with nonlinear decision boundaries, a view is a subset of attributes in the feature space, not in the input space. That is, a view is the subset of bases in the functional space.

The remainder of the paper is organized as follows. Section II provides the concepts of PA and down-weighting that are used for *PASL*. In Section III, we introduce the framework of our method, including RSVM and the proposed *PASL* algorithm. Section IV describes the numerical experiments

[2]Three choices for the case of two teachers and one student.

and details the results. Section V contains some concluding remarks.

II. Two Concepts for *PASL*

The proposed *PASL* is a PA-like multi-view SSL algorithm. In our approach, we blend the concepts of PA [3] and down-weighting into our multi-view learning framework.

A. The Concept of Passive-Aggressive Algorithm

The PA algorithm [3] is a margin-based learning algorithm, which is often used in on-line learning. On one hand, the on-line PA algorithm [3] modifies the current classifier $\mathbf{w}_t + b$ in order to correctly classify the current example $\mathbf{x}_t$ by updating the weight vector from $\mathbf{w}_t$ to $\mathbf{w}_{t+1}$. On the other, the new classifier $\mathbf{w}_{t+1} + b$ must be as close as possible to the current classifier $\mathbf{w}_t + b$.

Our method for SSL is inspired in part by the above idea of PA. A typical SSL method directly or indirectly labels the unlabeled data as accurate as possible and combines the labeled data to form a classifier. Inspired by the idea of PA, we train the classifier in a on-line fashion, and in each step of on-line training, we would like the newly generated SSL classifier $\mathbf{w}+b$ to be as close as possible to the classifier $\mathbf{w}_L+b$ produced by using only the labeled data. Hence, the concepts of PA could be introduced for dealing with the SSL problem.

To achieve this purpose, we add the term $\frac{1}{2}\|\mathbf{w}-\mathbf{w}_L\|_2^2$ into the objective function of the standard SVM and the standard SVM will be reformulated as:

$$\begin{array}{ll}\min\limits_{(\mathbf{w},b,\xi)} & C\mathbf{1}'\xi + \frac{1}{2}\|\mathbf{w}\|_2^2 + \frac{1}{2}\|\mathbf{w}-\mathbf{w}_L\|_2^2 \\ s.t. & D(A\mathbf{w}+\mathbf{1}b)+\xi \geq \mathbf{1}, \\ & \xi \geq 0,\end{array} \tag{1}$$

where the components of vector ξ in problem (1) are slack variables, and C is a penalty parameter. We use an diagonal matrix $D, D_{ii} = y_i$ to specify the membership of each input point.

The objective function of (1) can be further simplified as follows:

$$\begin{aligned} & C\mathbf{1}'\xi + \tfrac{1}{2}\|\mathbf{w}\|_2^2 + \tfrac{1}{2}\|\mathbf{w}-\mathbf{w}_L\|_2^2 \\ = \; & C\mathbf{1}'\xi + \tfrac{1}{2}\|\mathbf{w}\|_2^2 + \tfrac{1}{2}\|\mathbf{w}\|_2^2 + \tfrac{1}{2}\|\mathbf{w}_L\|_2^2 - <\mathbf{w},\mathbf{w}_L> \\ = \; & C\mathbf{1}'\xi + \|\mathbf{w}\|_2^2 + \tfrac{1}{2}\|\mathbf{w}_L\|_2^2 - <\mathbf{w},\mathbf{w}_L> \end{aligned} \tag{2}$$

Then, the standard SVM could be reformulated as:

$$\begin{array}{ll}\min\limits_{(\mathbf{w},b,\xi)} & C\mathbf{1}'\xi + \|\mathbf{w}\|_2^2 + \frac{1}{2}\|\mathbf{w}_L\|_2^2 - <\mathbf{w},\mathbf{w}_L> \\ \text{s.t.} & D(A\mathbf{w}+\mathbf{1}b)+\xi \geq \mathbf{1}, \\ & \xi \geq 0.\end{array} \tag{3}$$

Because the term $\frac{1}{2}\|\mathbf{w}_L\|_2^2$ is a constant, the problem (3) is equivalent to the formulation below:

$$\begin{array}{ll}\min\limits_{(\mathbf{w},b,\xi)} & C\mathbf{1}'\xi + \frac{1}{2}\|\mathbf{w}\|_2^2 - C_{\mathrm{PA}} <\mathbf{w},\mathbf{w}_L> \\ \text{s.t.} & D(A\mathbf{w}+\mathbf{1}b)+\xi \geq \mathbf{1}, \\ & \xi \geq 0,\end{array} \tag{4}$$

where C_{PA} in problem (4) is a weight parameter used for the trade-off between maximizing the margin $\frac{1}{\|\mathbf{w}\|_2}$ and generating a new classifier close to the original classifier.

B. The Technique of Down-weighting

As mentioned above, the guessed labeled points will join the training set to refine the current classifier. However, we do not know whether the guessed labeled data are labeled correctly. Therefore, in our scheme, we also consider how to separate the influence of the guessed labeled data from the effect of the given initial labeled points. This goal could be achieved by using the technique of down-weighting.

In our approach, we give large penalty weight to the initial labeled points if they are misclassified in training procedure because their labels are *explicitly* known. The penalty weight of the guessed labeled points is given lower than the one of the initial labeled data because the correctness of the guessed labels information are *not* certain. Hence, the problem (4) could be rewritten as:

$$\begin{array}{ll}\min\limits_{(\mathbf{w},b,\xi_L,\xi_G)} & C_L\mathbf{1}'\xi_L + C_G\mathbf{1}'\xi_G + \frac{1}{2}\|\mathbf{w}\|_2^2 - C_{\mathrm{PA}} <\mathbf{w},\mathbf{w}_L> \\ \text{s.t.} & D_L(A_L\mathbf{w}+\mathbf{1}b)+\xi_L \geq \mathbf{1}, \\ & D_G(A_G\mathbf{w}+\mathbf{1}b)+\xi_G \geq \mathbf{1}, \\ & \xi_L, \xi_G \geq 0,\end{array} \tag{5}$$

where the components of vectors ξ_L and ξ_G in problem (5) are slack variables for the initial labeled and the guessed labeled points, respectively; also, the parameters C_L and C_G are penalty weights of the initial labeled and the guessed labeled data respectively. The diagonal matrix $D_L, D_{L_{ii}} = y_i$ and $D_G, D_{G_{ii}} = \hat{y}_i$ are to specify the membership of each input initial labeled and guessed labeled point, respectively. In the rest of this paper, we will use the modified SVM model (5) for our SSL work.

III. The Framework of Passive-Aggressive Semi-supervised Learning

Our method is built on an alternately 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, *PASL* is built on the RSVM framework, where each reduced set serves as a view in the feature space for semi-supervised learning[3].

[3]We give only a brief description of RSVM. Please refer to [6] for all the details.

A. 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:

$$f(\mathbf{x}) = \sum_{j=1}^{m} u_j k(\mathbf{x}, \mathbf{x}^j) + b\,, \tag{6}$$

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 (or *similarities*) of instances in the 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

$$k(\mathbf{x}, \mathbf{z}) = e^{-\mu\|\mathbf{x}-\mathbf{z}\|_2^2}\,, \tag{7}$$

where μ is the *width* parameter. A kernel with larger value of μ tends to fit to the training data better; however, it may lead to *overfitting*. The coefficients u_j and b in (6) are determined by solving a quadratic programming problem [5], [8] or an unconstrained minimization problem [9].

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 difficulties, some authors have proposed applying low-rank approximation to the full kernel matrix [10], [11]. As an alternative, the reduced support vector machine (RSVM) was proposed in [6]. RSVM's operations can be divided into two steps. First, it randomly selects a small subset of bases $\{k(\cdot, \tilde{A}_1), k(\cdot, \tilde{A}_2), \cdots, k(\cdot, \tilde{A}_{\tilde{m}})\}$ from the full[4] data bases $\{k(\cdot, A_j)\}_{j=1}^{m}$ to build a separating surface prior to training. In contrast to conventional SVMs, RSVM replaces the fully dense square kernel matrix with a small rectangular kernel matrix, which is used in the nonlinear SVM formulation to avoid the above-mentioned computational difficulties. In the second step, RSVM determines the best coefficients of the selected kernel functions by solving the unconstrained minimization problem. It considers the entire data set, so the surface 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 can be found in [7].

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

Next, we discuss the roles of the reduced sets $K(A, \tilde{A}_j) \in \mathbb{R}^{m\times 1}$ as different views in our multi-view algorithm. The value of $K(A, \tilde{A}_j)$ can be interpreted as the *similarity* between all of the training examples and $\tilde{A}_j$. Likewise, the rectangular kernel matrix, which is generated by a reduced set, records the similarity between the entire training set and the reduced set. Ideally, to be effective as a 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 or more than one reduced set is involved in an SSL problem, we prefer the views or the sets to be linearly independent of each other numerically. This suggests that, in the semi-supervised case, views (reduced sets) with more linear independence between them are *less likely* to have a uniform predicted result; therefore, they give a result of high confidence when they agree. There are various algorithms for selecting a representative reduced set with dissimilar elements, e.g., those proposed in [12], [13]. In our SSL application, we choose a set of multi-view partners or reduced kernel matrices that are *linearly independent* of each other. Note that our selection procedure considers *both* labeled and unlabeled data points.

B. View Selection

As mentioned in Section III-A, 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. Next, we describe our mechanism for generating three reduced sets[5] or views, which will take turns to play the roles of teacher and student in our *PASL* algorithm. The choice of reduced sets can be very flexible [12], [13]. We can use IRSVM [12] to generate all three views because it guarantees *dissimilar* basis functions as the *representatives* (Please refer to [12] for all the details.). 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 forms a reduced set that plays a view or role in the *PASL* algorithm.

The detailed procedure for generating the three reduced sets is as follows. Suppose we want a reduced set whose size is equal to $\tilde{m}$. In this case, we repeat the IRSVM procedure until the number of reduced points is equal to $3\tilde{m}$.

[5]Again, a number of more than three reduced sets should be easy to generalize.

Let $\tilde{A}_{3\tilde{m}}$ be the set of $3\tilde{m}$ reduced points, which we split into three subsets (views) through a round-robin (interleaving) partition method called $\tilde{B}_{\tilde{m}}$, $\tilde{C}_{\tilde{m}}$, and $\tilde{D}_{\tilde{m}}$. Based on the IRSVM algorithm, 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 $K(A, \tilde{B}_{\tilde{m}})$, $K(A, \tilde{C}_{\tilde{m}})$, and $K(A, \tilde{D}_{\tilde{m}})$, denoted by $CS(K(A, \tilde{B}_{\tilde{m}}))$, $CS(K(A, \tilde{C}_{\tilde{m}}))$, and $CS(K(A, \tilde{D}_{\tilde{m}}))$, are spanned by the above three mutually exclusive basis functions, respectively. Thus these hypothesis spaces are orthogonal, and for any two distinct views $\mathcal{V}_i, \mathcal{V}_j \in \{\tilde{B}_{\tilde{m}}, \tilde{C}_{\tilde{m}}, \tilde{D}_{\tilde{m}}\}$, 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 reduced sets (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.

C. The Multi-view Learning for PASL

In this subsection, we introduce the *PASL* algorithm for iterative labeling and training. This approach is inspired in part by the well-known co-training method [4] 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 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 the *PASL* algorithm. In the labeling step, two teachers from 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 repeat the "teaching" step until the student classifier can not "learn" any more from the two teacher classifiers. That is, we repeat the above procedure to label the unlabeled data until the labeled part makes no more, or very few, changes. We then use all the labeled data to build the final classifier, which is used for making predictions on the *unseen* data. We describe the complete *PASL* algorithm formally in Algorithm 1.

Algorithm 1: The *PASL* 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 $f_1(\mathbf{x})$, $f_2(\mathbf{x})$, $f_3(\mathbf{x})$.

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**
    for $j \leftarrow 1$ **to** u **do**
      $t_1 \leftarrow \mathrm{mod}(i-1, 3) + 1$
      $t_2 \leftarrow \mathrm{mod}(i, 3) + 1$
      $s \leftarrow \mathrm{mod}(i+1, 3) + 1$
      if $(f_{t_1}(\mathbf{x}^j) > 0$ **and** $f_{t_2}(\mathbf{x}^j) > 0)$ **or**
        $(f_{t_1}(\mathbf{x}^j) < 0$ **and** $f_{t_2}(\mathbf{x}^j) < 0)$
      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 $f(\mathbf{x})$.

IV. Experiment Results

To demonstrate the performance of the *PASL* algorithm, we test it on four publicly available data sets from the UCI machine learning repository [14]. Table I 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.

TABLE I
The statistics of the data sets used in the experiments.

Data Set Description			
Data Set	Instance	Feature	Reduced Set Size
Ionosphere	351	34	35
Cleveland Heart	297	13	30
BUPA Liver	345	6	35
Pima Indians	768	8	50

As mentioned in the previous section, we use the IRSVM [12] procedure in our experiments to generate three views for our teachers-student combination. The sizes of the reduced sets used in all the experiments are also summarized in Table I. We use Gaussian kernel functions for RSVM

TABLE II
THE AVERAGE CPU TIMES OF THE *PASL* ALGORITHM ON FOUR PUBLIC DATA SETS USING 20%, 30%, 40%, AND 50% OF THE TRAINING SET AS LABELED DATA.

CPU Time $\pm$ Std (sec)				
Data Set	20%	30%	40%	50%
Ionosphere	2.51±0.10	2.45±0.08	2.47±0.05	2.50±0.10
Cleveland Heart	0.19±0.02	0.18±0.01	0.19±0.02	0.19±0.01
BUPA Liver	3.48±0.18	3.47±0.21	3.54±0.20	3.54±0.13
Pima Indians	14.43±0.93	14.63±0.90	14.79±1.03	15.05±1.09

TABLE III
THE AVERAGE NUMBER OF ITERATIONS OF THE *PASL* ALGORITHM ON FOUR PUBLIC DATA SETS USING 20%, 30%, 40%, AND 50% OF THE TRAINING SET AS LABELED DATA.

The Number of Iterations $\pm$ Std (%)				
Data Set	20%	30%	40%	50%
Ionosphere	2.12±0.17	2.13±0.19	2.04±0.07	2.02±0.06
Cleveland Heart	2.07±0.14	2.03±0.07	1.99±0.07	1.93±0.09
BUPA Liver	2.04±0.09	2.01±0.04	1.99±0.08	1.97±0.05
Pima Indians	2.06±0.18	2.00±0.03	2.00±0.06	1.99±0.04

and IRSVM in all the experiments. Besides, we adopt the nested uniform design (UD) model selection method [15] to select the penalty parameter $\mathcal{C}$ and the Gaussian kernel width parameter μ for RSVM. To evaluate the performance of *PASL*, we compare with that of the pure supervised learning scheme under the same setting. We ran tenfold cross-validation 30 times on each data set. 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.

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 $\mathcal{D}_L$; while *test set accuracy* or *inductive accuracy* denotes the classification accuracy on the fresh test set, which was not seen before the training commenced.

We summarize the numerical results and comparisons of the experiments in Tables II to V. The CPU times required to implement *PASL* in the experiments and the average number of iterations of *PASL* are demonstrated in Table II and III, respectively. The numerical results show that although our method needs to retrain iteratively, the time cost is still acceptable and the average number of iterations are quite small.

Table IV compares the average test accuracy of the *PASL* and the pure supervised learning scheme. The *PASL* algorithm only uses a small portion of the labeled training data points, but its test accuracy is comparable to the pure supervised learning scheme that uses all the labeled data points for training. The test accuracy of the pure supervised learning classifiers generated by 20%, 30%, 40%, and 50% of the training labeled data points was lower than the test accuracy of the classifiers built by *PASL*.

Table V details the average training accuracy and the average numbers of final labeled points for *PASL* based on ten-fold cross-validation. The numerical results show that the *PASL* could label almost *all* the unlabeled data with high accuracy. Based on the limited but informative estimated labeled data as well as the original labeled data, we build the final classifier. The results show that the final classifier achieves a competitive performance.

V. CONCLUSION

We have proposed an *PASL* algorithm for semi-supervised learning. Our method can achieve high accuracy rates on both transductive learning (measured by training accuracy) and inductive learning (measured by test accuracy). The *PASL* algorithm blends the concepts of PA and down-weighting into our multi-view scheme. In *PASL* the reduced sets are chosen as the views in the view selection process. Unlike other multi-view methods, *PASL* selects views in the feature space rather than in the input space. As a multi-view approach, our method combines the concepts of co-training and consensus training. In training procedure, we use the idea of down-weighting to retrain the student's classifier by the given initial labeled and guessed labeled points. Based on the idea of PA algorithm, we would like the new retrained classifier to be held as near as possible to the original classifier produced by the initial labeled data. The *PASL* algorithm alternately labels the unlabeled data based on classifiers on hold and built the classifiers based on the original labeled data, and the guessed labeled data obtained from previous classification results. We evaluated the performance of *PASL* on four publicly available data sets. The numerical results show that the algorithm only uses a small portion of the labeled training data points, yet it achieves comparable cross validation accuracy to the algorithm that uses all the labeled data points.

REFERENCES

[1] X. Zhu, "Semi-supervised learning literature survey," Dept. of Computer Science, University of Wisconsin, Madison, Tech. Rep. 1530, December 2005, http://www.cs.wisc.edu/∼jerryzhu/pub/ssl_survey.pdf.

[2] O. Chapelle, B. Schölkopf, and A. Zien, Eds., *Semi-Supervised Learning*. Cambridge, MA: MIT Press, 2006. [Online]. Available: http://www.kyb.tuebingen.mpg.de/ssl-book

TABLE IV

TEN-FOLD CROSS-VALIDATION RESULTS OF THE AVERAGE TEST ACCURACY ON FOUR PUBLIC DATA SETS WHEN WE USE THE *PASL* ALGORITHM AND THE PURE SUPERVISED LEARNING SCHEME WITH THE SAME PERCENTAGE OF LABELED POINTS AS TRAINING SET. THE NUMERICAL RESULTS OF RSVM GENERATED BY FULL TRAINING POINTS ARE ALSO SHOWN IN THIS TABLE FOR COMPARISON.

Ten-fold Test Set Accuracy ± Std (%)						
Data Set	Method	20%	30%	40%	50%	100%†
Ionosphere	*PASL*	90.45±1.99	91.59±1.36	92.26±1.32	92.29±0.82	-
	SL Alg.	81.26±6.72	86.80±5.87	89.83±2.94	91.64±1.80	94.16±0.84
Cleveland Heart	*PASL*	76.55±2.59	78.58±1.75	79.98±1.94	81.46±1.64	-
	SL Alg.	72.57±1.75	73.99±1.99	76.56±2.31	78.52±2.59	83.61±0.80
BUPA Liver	*PASL*	67.86±2.58	69.36±1.84	69.52±1.70	70.96±1.43	-
	SL Alg.	62.37±3.73	65.37±2.75	68.06±2.24	69.72±2.30	73.27±0.80
Pima Indians	*PASL*	74.67±1.00	75.22±0.88	75.67±0.69	76.03±0.82	-
	SL Alg.	71.54±2.88	73.91±1.96	74.69±1.13	75.36±0.84	75.87±0.76

†:100% means performing the RSVM with entire training labeled set.

TABLE V

TEN-FOLD CROSS-VALIDATION RESULTS OF THE AVERAGE TRAINING ACCURACY AND THE AVERAGE NUMBER OF FINAL LABELED POINTS ON FOUR PUBLIC DATA SETS WHEN WE USE THE *PASL* ALGORITHM ALGORITHM. THE FIGURES IN PARENTHESES ARE THE NUMBERS OF INSTANCES IN EACH DATA SET, COPIED FROM TABLE I.

Ten-fold Training Set Accuracy ± Std (%) Ten-fold Number of Labeled Points ± Std				
Data Set	20%	30%	40%	50%
Ionosphere	92.49±1.48	93.79±1.10	94.66±0.79	95.08±0.78
(351)	315.9±1.7345e-13	315.9±1.7345e-13	315.9±1.7345e-13	315.9±1.7345e-13
Cleveland Heart	79.83±1.80	82.03±1.33	83.61±1.16	84.72±0.82
(297)	267.3±1.1563e-13	267.3±1.1563e-13	267.3±1.1563e-13	267.3±1.1563e-13
BUPA Liver	71.22±1.96	72.13±1.50	73.13±1.37	74.09±0.90
(345)	310.5±0.00	310.5±0.00	310.5±0.00	310.5±0.00
Pima Indians	75.87±0.93	76.39±0.55	76.98±0.52	77.20±0.42
(768)	691.2±3.4689e-13	691.2±3.4689e-13	691.2±3.4689e-13	691.2±3.4689e-13

[3] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *Journal of Machine Learning Research*, vol. 7, pp. 551–585, 2006.

[4] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *COLT: Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann Publishers*, 1998, pp. 92–100.

[5] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," Data Mining and Knowledge Discovery, vol. 2, no. 2, pp. 121–167, 1998.

[6] Y.-J. Lee and O. L. Mangasarian, "RSVM: Reduced support vector machines," in *Proceedings of the First SIAM International Conference on Data Mining*, 2001.

[7] Y.-J. Lee and S.-Y. Huang, "Reduced support vector machines: A statistical theory," *IEEE Transactions on Neural Networks*, vol. 18, pp. 1–13, 2007.

[8] V. N. Vapnik, The Nature of Statistical Learning Theory. New York: Springer, 1995.

[9] Y.-J. Lee and O. L. Mangasarian, "SSVM: A smooth support vector machine," *Computational Optimization and Applications*, vol. 20, pp. 5–22, 2001.

[10] A. J. Smola and B. Schölkopf, "Sparse greedy matrix approximation for machine learning," in *Proc. 17th International Conf. on Machine Learning*. Morgan Kaufmann, San Francisco, CA, 2000, pp. 911–918.

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

[12] Y.-J. Lee, H.-Y. Lo, and S.-Y. Huang, "Incremental reduced support vector machines," in *International Conference on Informatics, Cybernetics and Systems (ICICS 2003)*, Kaohsiung, Taiwan, 2003.

[13] L.-J. Chien, C.-C. Chang, and Y.-J. Lee, "Variant methods of reduced set selection for reduced support vector machines," *Journal of Information Science and Engineering*, vol. 26, no. 1, pp. 183–196, 2010.

[14] A. Asuncion and D. Newman, "UCI repository of machine learning databases," 2007, http://www.ics.uci.edu/∼mlearn/MLRepository.html.

[15] C.-M. Huang, Y.-J. Lee, D. K. J. Lin, and S.-Y. Huang, "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, 2007.