Towards Making Unlabeled Data Never Hurt

Yu-Feng Li and Zhi-Hua Zhou, *Fellow, IEEE*

Abstract—It is usually expected that learning performance can be improved by exploiting unlabeled data, particularly when the number of labeled data is limited. However, it has been reported that, in some cases existing semi-supervised learning approaches perform even worse than supervised ones which only use labeled data. For this reason, it is desirable to develop *safe* semi-supervised learning approaches that will not significantly reduce learning performance when unlabeled data are used. This paper focuses on improving the safeness of semi-supervised support vector machines (S3VMs). First, the S3VM-us approach is proposed. It employs a conservative strategy and uses only the unlabeled instances that are very likely to be helpful, while avoiding the use of highly risky ones. This approach improves safeness but its performance improvement using unlabeled data is often much smaller than S3VMs. In order to develop a safe and well-performing approach, we examine the fundamental assumption of S3VMs, i.e., low-density separation. Based on the observation that multiple good candidate low-density separators may be identified from training data, safe semi-supervised support vector machines (S4VMs) are here proposed. This approach uses multiple low-density separators to approximate the ground-truth decision boundary and maximizes the improvement in performance of inductive SVMs for any candidate separator. Under the assumption employed by S3VMs, it is here shown that S4VMs are provably safe and that the performance improvement using unlabeled data can be maximized. An out-of-sample extension of S4VMs is also presented. This extension allows S4VMs to make predictions on unseen instances. Our empirical study on a broad range of data shows that the overall performance of S4VMs is highly competitive with S3VMs, whereas in contrast to S3VMs which hurt performance significantly in many cases, S4VMs rarely perform worse than inductive SVMs.

Index Terms—Unlabeled data, semi-supervised learning, safe, S3VMs, S4VMs

1 Introduction

Traditional supervised learning often assumes that large numbers of labeled data are readily available for training. In many practical applications, however, the acquisition of class labels is expensive because the labeling process requires human effort and expertise. For example, in computer-aided medical diagnosis, large numbers of X-ray images can be obtained from routine examinations, but it is costly and difficult for physicians to mark all focuses in all images. In this case, training with only labeled data may not lead to a good performance. It is possible to employ *semi-supervised learning* [10], [34], [51], [52] that exploits the wide availability of unlabeled data to improve performance. During the past decade, semi-supervised learning has attracted significant attention. It has been found useful in many applications, including text categorization [23], image retrieval [42], bioinformatics [24], and natural language processing [19].

Existing semi-supervised approaches can be roughly grouped into four categories. The first category is generative methods, e.g., [35], [36]. These methods extend supervised generative models by incorporating unlabeled data, and estimate model parameters and labels using techniques such as the EM algorithm [17]. The second category is graph-based methods, e.g., [2], [7], [34], [48], [53]. These methods encode both the labeled and unlabeled instances in a graph and then assign class labels to the unlabeled data such that their inconsistencies with both the labeled data and the underlying graph are minimized. The third category is disagreement-based methods, e.g., [8], [50]. These methods typically involve multiple learners and improve them through the exploitation of disagreement among the learners. The fourth category is semi-supervised support vector machines (S3VMs), e.g., [4], [23]. They use unlabeled data to regularize the decision boundary so that it can pass through low-density regions [12].

It is generally accepted that by using unlabeled data, semi-supervised learning can help improve the performance, particularly when the number of labeled data is limited. Many empirical studies, however, show that there are cases in which the use of unlabeled data decreases the performance [7], [11], [13], [14], [16], [20], [36], [47], [50]. Such phenomena undeniably encumber the deployment of semi-supervised learning in real applications, especially tasks requiring high reliability, because users usually require that new techniques (such as semi-supervised learning) should perform at least as well as existing techniques (such as pure supervised learning). For this reason, it is desirable to have *safe* semi-supervised learning approaches which never reduce learning performance significantly when using unlabeled data. This is a challenging task, and only a few authors have explicitly tried to reduce the chance of performance degeneration [14], [27], even though there are already many studies on semi-supervised learning. *Safe*, here means that the generalization performance is never

• *Y.-F. Li and Z.-H. Zhou are with the National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China. E-mail: {liyf,zhouzh}@lamda.nju.edu.cn*

statistically significantly worse than methods using only labeled data. It is meaningless to talk about a single trial, because for a single trial, even exploiting more labeled data might result in a worse performance.

Cozman et al. [16] discussed the reason why unlabeled data can increase classification error for generative methods. They conjectured that the performance degeneration is caused by incorrect model assumptions, because fitting unlabeled data based on an incorrect model assumption will mislead the learning process. However, it is very difficult to make a correct model assumption without sufficient domain knowledge. For graph-based methods, researchers realized that graph construction is the crucial problem. However, developing a good graph in general situations remains an open problem. Disagreement-based methods usually use pseudo-labels of unlabeled data provided by multiple learners to enhance the labeled data set. In this way, incorrect pseudo-labels may disrupt the learning process. One possible solution is to use data editing techniques to examine data that may have been pseudo-labeled [27]. However, such solutions work well only on dense data. This is because data editing techniques usually rely on the data neighboring information. With S3VMs, the correctness of the optimization objective has been studied on very small data sets [11]. However, there is no clear solution that can be used to prevent performance from degeneration when using unlabeled data. There are also some general discussions on the usefulness of unlabeled data from a theoretical perspective [1], [3], [38]. In particular, in [1], the authors showed that when unlabeled data provide a good regularizer, a purely inductive supervised SVM on labeled data using such a regularizer guarantee a good generalization. Deriving such a good regularizer, however, remains an open problem.

Particularly, S3VMs have been widely applied to many tasks [10], and their representative algorithm, TSVM [23], has won the Ten-Year Best Paper Award for machine learning in 2009. Most research efforts on S3VMs address its complexity [11], [15], [23], [28], with little effort on its safeness, although many empirical studies have shown that S3VMs also reduce performance, sometimes even seriously [10], [42], [47].

This paper focuses on improving the safeness of S3VMs. First, because the main use of unlabeled data is to determine data distribution, it is here conjectured that the degradation of the performance degeneration of S3VMs is caused by unlabeled instances that are obscure or misleading for the discovery of the underlying distribution. For this reason, the S3VM with unlabeled data selection (S3VM-us) approach is here proposed. It uses hierarchical clustering to estimate the reliability of unlabeled instances and then removes the ones with the lowest reliability.

Our empirical studies show that S3VM-us improves the safeness of S3VMs. However, its improvement in performance using unlabeled data is not as considerable as S3VMs. To develop a safe and well-performing approach, we then examine the fundamental assumption of S3VMs, i.e., low-density separation, and get another conjecture on the reason of performance degeneration. Given a few labeled data and many more unlabeled data, there is usually more than one large-margin low-density separator. However, it is hard to determine which one is optimal based on the limited labeled data. Although these low-density separators are all consistent with the limited labeled data, they can be very diverse with respect to the instance space. In this way, incorrect selection may result in a reduced performance. Based on this observation, the S4VMs (Safe S3VMs) approach, the main contribution of this paper, is proposed. S4VMs use multiple low-density separators to approximate the ground-truth decision boundary and maximize the improvement in performance against inductive SVMs for any candidate separator. S4VMs are shown to be safe and to achieve the maximal performance improvement under the low-density assumption of S3VMs. An out-of-sample extension of S4VMs is also presented so that S4VMs can make predictions on unseen instances. Our empirical studies performed on a broad range of data sets show that S4VMs perform highly competitive with S3VMs. More importantly, unlike S3VMs which significantly reduce performance in many cases, S4VMs are rarely inferior to inductive SVMs.

The rest of this paper is organized as follows. S3VMs are briefly introduced in Section 2. The S3VM-us and S4VMs are introduced in Sections 3 and 4. Empirical results are report in Section 5. Conclusions are presented in Section 6.

2 Brief Introduction to S3VMs

Inspired by the success of the large-margin principle [40], S3VMs extend inductive supervised SVMs to semi-supervised learning. They simultaneously learn the optimal decision function and the labels of unlabeled instances such that the decision boundary has a large margin on both the labeled and unlabeled data. It was discovered that S3VMs realize the low-density assumption [12] which states that the decision boundary will go across low-density regions.

Formally, we consider binary classification here. Let $\mathcal{X}$ be the input space and $\mathcal{Y} = \{\pm 1\}$ be the label space. Given a set of l labeled instances $\{\mathbf{x}_i, y_i\}_{i=1}^{l}$ and u unlabeled instances $\{\mathbf{x}_j\}_{j=l+1}^{l+u}$, S3VMs aim to find a decision function $f : \mathcal{X} \to \{\pm 1\}$ and a label assignment on unlabeled instances $\mathbf{y} = \{y_{l+1}, \ldots, y_{l+u}\} \in \mathcal{B}$ such that the following functional is minimized,

$$\min_{\substack{f \in \mathcal{H} \\ \mathbf{y} \in \mathcal{B}}} \frac{1}{2}\|f\|_{\mathcal{H}}^2 + C_1 \sum_{i=1}^{l} \ell(y_i, f(\mathbf{x}_i)) + C_2 \sum_{j=l+1}^{l+u} \ell(y_j, f(\mathbf{x}_j)). \quad (1)$$

Here $\mathcal{B}$ is a set of label assignments obtained from domain knowledge. For example, when the class pro-

portion of unlabeled data is closely related to that of labeled data (also refer to as *balance constraint* [11], [23]), we can set

$$\mathcal{B} = \{\mathbf{y} \in \{\pm 1\}^u | -\beta \leq \frac{\sum_{j=l+1}^{l+u} y_j}{u} - \frac{\sum_{i=1}^{l} y_i}{l} \leq \beta\}$$

where β is a small constant controlling the inconsistency of class proportions. $\mathcal{H}$ is the Reproducing Kernel Hilbert Space (RKHS) induced by a kernel function κ. $\ell(y, f(\mathbf{x})) = \max\{0, 1 - yf(\mathbf{x})\}$ is the hinge loss used in SVMs. C_1 and C_2 are two regularization parameters trading off model complexity and empirical losses on the labeled and unlabeled data, respectively.

Similar to supervised SVMs, S3VMs favor the decision boundary having a large margin on all training data. According to [12], they inherently favor the decision boundary going through low-density regions. Otherwise a large loss will occur with respect to the objective of S3VMs [12].

Unlike supervised SVMs where the training labels are complete, S3VMs need to infer the integer-value labels of the unlabeled instances, resulting in a difficult mixed-integer programming problem. Great efforts have been devoted to coping with the high complexity of S3VMs. Roughly speaking, they can be grouped into four categories. The first kind of approaches is based on global combinatorial optimization. Examples include branch-and-bound methods [4], [11], which solve S3VMs globally and obtain good performance on small data sets. The second kind of approaches is based on global heuristic search, which gradually increases the difficulty of solving the non-convex part in Eq. 1. Examples include TSVM [23] which gradually increases the influence of unlabeled data (i.e., the value of C_2), the deterministic annealing approach [37] which gradually increases the temperature of an entropy function in optimization, and the continuation method [9] which first introduces a surrogate smooth function and then gradually decreases the smoothness of the surrogate function to approach the objective in Eq. 1. The third kind of approaches is based on convex relaxation, which transforms Eq. 1 into a relaxed convex problem. Examples include the semi-definite programming (SDP) relaxation [6], [43], and the minimax relaxation [28], [29], [30] which is tighter and more scalable than the SDP relaxation. The fourth kind of approaches is based on efficient non-convex optimization techniques. Examples include UniverSVM [15] which employs concave-convex procedure (CCCP) [44], and meanS3VM [28] which employs alternating optimization [5].

Because S3VMs involve a complicated optimization task, most previous efforts were devoted to handling the high complexity, whereas few literatures have explicitly studied the safeness of S3VMs.

3 S3VM-us

It is generally accepted that the major utility of unlabeled data is to disclose useful information about the underlying data distribution [10]. When some unlabeled instances are obscure or misleading for the discovery of the underlying distribution, learning performance may be reduced by using those data. Based on this observation, S3VM-us, which tries to exclude highly risky unlabeled instances, is proposed.

In the following, two simple approaches to exclude highly risky unlabeled instances, i.e., the S3VM-c and S3VM-p approaches, are first introduced and by examining the deficiencies of S3VM-c and S3VM-p, S3VM-us is then presented. For the simplicity of notations, the training set is denoted as $\mathcal{D} = \{\{\mathbf{x}_i, y_i\}_{i=1}^{l}, \{\mathbf{x}_j\}_{j=l+1}^{l+u}\}$. The predicted labels for $\mathbf{x}$ by inductive SVM (using labeled data only) and S3VM are denoted as $y^{svm}(\mathbf{x})$ and $y^{s3vm}(\mathbf{x})$, respectively. The transpose of a vector is denoted by the superscript $'$.

3.1 Two Simple Approaches

3.1.1 S3VM-c

The first simple approach S3VM-c is motivated by [38]. It suggests that unlabeled data will be helpful when the *component density sets* are discernible, where component density sets refer to regions of data distribution with non-zero probability density. To implement this idea, in S3VM-c, the component density sets are simulated by clusters obtained with a clustering algorithm, and the discernibility is simulated by a disagreement between S3VM and inductive SVM based on *bias* and *confidence*. It is noteworthy that other simulations are also possible. As Algorithm 1 shows, we rely on the prediction of S3VM if S3VM obtains the same bias but enhances the confidence of the inductive SVM. Otherwise we will rely on the prediction of the inductive SVM.

3.1.2 S3VM-p

The second simple approach S3VM-p is motivated by the confidence estimation in label propagation methods [48], [53], where the confidence can be naturally regarded as a measurement of the reliability of unlabeled data.

Formally, to estimate the confidence of unlabeled data, let $\mathbf{y}_l = [y_1, \ldots, y_l]' \in \{\pm 1\}^{l \times 1}$ and $\mathbf{F}^l = [(\mathbf{y}_l + 1)/2, (1 - \mathbf{y}_l)/2] \in \{0, 1\}^{l \times 2}$ be the vector- and matrix-form of labeled data, respectively. Let $\mathbf{W} = [w_{ij}] \in \mathbb{R}^{(l+u)\times(l+u)}$ be the similarity matrix of training data, and $\mathbf{\Lambda} = \mathbf{D} - \mathbf{W}$ the laplacian matrix of $\mathbf{W}$, where $\mathbf{D}$ is a diagonal matrix with entries $d_i = \sum_{j=1}^{l+u} w_{ij}$, $i = 1, \ldots, l+u$. According to [53], the predictions of unlabeled data $\mathbf{F}^u$ are derived as,

$$\mathbf{F}^u = \mathbf{\Lambda}_{u,u}^{-1} \mathbf{W}_{u,l} \mathbf{F}^l, \tag{2}$$

where $\mathbf{\Lambda}_{u,u}$ refers to a sub-matrix of $\mathbf{\Lambda}$ on the block of unlabeled data, $\mathbf{W}_{u,l}$ refers to a sub-matrix of $\mathbf{W}$

Algorithm 1 S3VM-c

Input: y^{svm}, y^{s3vm}, $\mathcal{D}$, k;

1: Perform clustering (e.g., using k-means) on $\mathcal{D}$. Denote $\mathcal{C}_1, \ldots, \mathcal{C}_k$ as the data indices of each cluster.
2: For each cluster $i = 1, \ldots, k$, calculate the label bias lb and the confidence cf of SVM and S3VM according to

$$lb_i^{s(3)vm} = \operatorname{sgn}\left(\sum_{j\in\mathcal{C}_i} y^{s(3)vm}(\mathbf{x}_j)\right)$$
$$cf_i^{s(3)vm} = \left|\sum_{j\in\mathcal{C}_i} y^{s(3)vm}(\mathbf{x}_j)\right|.$$

3: If $lb_i^{s3vm} = lb_i^{svm}$ and $cf_i^{s3vm} > cf_i^{svm}$, assign the unlabeled instances in $\mathcal{C}_i$ with the S3VM predictions. Otherwise assign with the SVM predictions.

Algorithm 2 S3VM-p

Input: y^{svm}, y^{s3vm}, $\mathcal{D}$, $\mathbf{W}$, ν;

1: Perform label propagation (e.g., using [53]) with similarity matrix $\mathbf{W}$. Obtain the predicted label $y^{lp}(\mathbf{x}_j)$ and confidence h_{j-l} for each unlabeled instance $\mathbf{x}_j$, $j = l+1, \ldots, l+u$.
2: Update $\mathbf{h} = [h_1, \ldots, h_u]$ according to

$$h_{j-l} = y^{s3vm}(\mathbf{x}_j) y^{lp}(\mathbf{x}_j) h_{j-l}, \quad j = l+1, \ldots, l+u.$$

Denote c as the number of nonnegative elements in $\mathbf{h}$.
3: Sort $\mathbf{h}$ in descending order. Pick up the top-$\min\{\nu u, c\}$ unlabeled instances and assign with the S3VM predictions. Others are assigned with the SVM predictions.

on the block between labeled and unlabeled data, and $\mathbf{\Lambda}_{u,u}^{-1}$ refers to the inverse matrix of $\mathbf{\Lambda}_{u,u}$. In $\mathbf{F}^u$, note that the two entries of each row refer to the confidence estimations belonging to two different classes. We then assign each unlabeled instance $\mathbf{x}_j$ with the label $y^{lp}(\mathbf{x}_j) = \operatorname{sgn}(\mathbf{F}^u_{j-l,1} - \mathbf{F}^u_{j-l,2})$, and the confidence $h_{j-l} = |\mathbf{F}^u_{j-l,1} - \mathbf{F}^u_{j-l,2}|$, $j = l+1, \ldots, l+u$. As Algorithm 2 shows, after confidence estimation, similar to S3VM-c, we consider the risk of unlabeled data by *bias* and *confidence*. If S3VM obtains the same bias of label propagation and the confidence is high, we use the S3VM prediction. Otherwise we use the inductive SVM prediction instead.

3.2 S3VM-us

S3VM-c and S3VM-p have not been reported before. Our empirical studies show that they are capable of reducing the chances of performance degeneration. However, they both suffer from some deficiencies. S3VM-c works in a local manner and the relations between clusters are never considered. In S3VM-p, as stated in [41], the confidence estimated with label propagation methods might be incorrect if the label initialization is highly imbalanced. Moreover, both S3VM-c and S3VM-p heavily rely on S3VM predictions. This might be risky when S3VM suffers from a serious reduced performance.

The examination of the deficiencies of S3VM-c and S3VM-p suggests us to exploit the relations between clusters and reduce the sensitivity to the label initialization. This motivates our S3VM-us approach.

Algorithm 3 S3VM-us

Input: y^{svm}, y^{s3vm}, $\mathcal{D}$, ϵ;

1: Let $\mathcal{S} = \{\mathbf{x}_j | y^{svm}(\mathbf{x}_j) \neq y^{s3vm}(\mathbf{x}_j), j = l+1, \ldots, l+u\}$ be a set of unlabeled instances that are inconsistent labeled by SVM and S3VM.
2: Perform hierarchical clustering (e.g., using the single linkage method [22]) on $\mathcal{D}$. Denote $\{\mathcal{Z}_i\}_{i=1}^{l+u-1}$ as the sets of instance indices for the clusters merged during the hierarchical clustering process.
3: For each $\mathbf{x}_j \in \mathcal{S}$, denote $\mathcal{Z}_{p_{j-l}}$ (resp. $\mathcal{Z}_{n_{j-l}}$) as the first set that contains $\mathbf{x}_j$ and at least one positive (resp. negative) labeled example. Denote t_{j-l} as $n_{j-l} - p_{j-l}$.
4: Let $\mathcal{B} = \{\mathbf{x}_j \in \mathcal{S} | \, |t_{j-l}| \geq \epsilon | l+u |, j = l+1, \ldots, l+u\}$ be the set of unlabeled instances owning high reliabilities.
5: If $\sum_{\mathbf{x}_j \in \mathcal{B}} \left(y^{s3vm}(\mathbf{x}_j) - y^{svm}(\mathbf{x}_j)\right) t_{j-l} \geq 0$, use the S3VM prediction for $\mathbf{x} \in \mathcal{B}$. Otherwise the SVM prediction.
6: For $\mathbf{x} \notin \mathcal{B}$, assign with the SVM prediction.

As Algorithm 3 shows, S3VM-us employs hierarchical clustering [22]. It first initializes each single instance as a cluster and then merges two of the clusters with the shortest distance. This process repeats until all the instances are merged into one cluster. It is not hard to validate that hierarchical clustering considers the between-cluster relations. Moreover, since hierarchical clustering is an unsupervised method, it does not suffer from the label initialization problem.

To estimate the reliability on unlabeled instances, let p_{j-l} and n_{j-l} denote the lengths of paths from an unlabeled instance $\mathbf{x}_j$ to its nearest positive and negative labeled instances, respectively. The difference between p_{i-l} and n_{i-l} is simply taken as an estimation of reliability. Intuitively, the larger the difference between p_{j-l} and n_{j-l}, the higher the reliability on labeling $\mathbf{x}_j$.

Our empirical studies in Section 5 show that S3VM-us effectively improves the safeness of S3VMs. However, its improvement in performance is often marginal when compared with existing S3VMs. To develop safe and well-performing methods, it might be insufficient to purely rely on the selection of unlabeled instances. This motivates us to develop the S4VM approach presented in the next section.

4 S4VMs

As previously mentioned, the underlying assumption of S3VMs is low-density separation. That is, the ground-truth is realized by a large-margin low-density separator. However, as illustrated in Figure 1, given limited labeled data and many more unlabeled data, there usually exist multiple large-margin low-density separators. Although these separators all coincide well with the labeled data, they could be quite diverse with respect to the feature space, and thus an inadequate selection may lead to a serious performance reduction. This observation incites us the

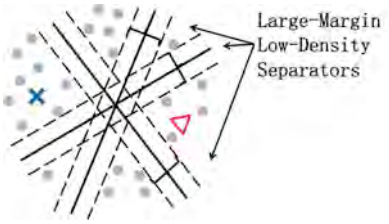


Fig. 1. There are multiple large-margin low-density separators coinciding well with labeled data (cross and triangle).

Algorithm 4 S4VM

Input: $\mathcal{D} = \{\{\mathbf{x}_i, y_i\}_{i=1}^{l}, \{\mathbf{x}_j\}_{j=l+1}^{l+u}\}$;
Output: $\mathbf{y}$.

1: Generate a pool of diverse large-margin low-density separators $\{\hat{\mathbf{y}}_t\}_{t=1}^{T}$ for $\mathcal{D}$.
2: Assign the labels $\mathbf{y} = \{y_{l+1}, \ldots, y_{l+u}\}$ to unlabeled instances such that the improvement in performance for any separator $\hat{\mathbf{y}}_t$, $t = 1, \ldots, T$, is maximized.

design of S4VMs. Specifically, S4VMs first generate a pool of diverse large-margin low-density separators, and then try to maximize the improvement in performance for any separator. The pseudo-code of S4VM is summarized in Algorithm 4.

In the following, we will first introduce how to build S4VMs given a pool of diverse large-margin low-density separators, and then present two different implementations for generating the pool.

4.1 Building S4VMs from a Pool of Separators

Let $\mathbf{y}^*$ be the ground-truth label assignment and $\mathbf{y}^{svm}$ be the predictive labels of inductive SVM on unlabeled instances. For any label assignment of unlabeled instances $\mathbf{y} = \{y_{l+1}, \ldots, y_{l+u}\}$, denote $gain(\mathbf{y}, \mathbf{y}^*, \mathbf{y}^{svm})$ and $loss(\mathbf{y}, \mathbf{y}^*, \mathbf{y}^{svm})$ as the gained and lost accuracies compared to the inductive SVM. Our goal is to learn a label assignment $\mathbf{y}$ such that the improved performance against the inductive SVM is maximized,

$$\max_{\mathbf{y}\in\{\pm1\}^u} \; gain(\mathbf{y}, \mathbf{y}^*, \mathbf{y}^{svm}) - \lambda \; loss(\mathbf{y}, \mathbf{y}^*, \mathbf{y}^{svm}), \quad (3)$$

where λ is a parameter for trading-off how much risk the user would like to undertake. In the sequel, we will denote $gain(\mathbf{y}, \hat{\mathbf{y}}, \mathbf{y}^{svm}) - \lambda \; loss(\mathbf{y}, \hat{\mathbf{y}}, \mathbf{y}^{svm})$ as $J(\mathbf{y}, \hat{\mathbf{y}}, \mathbf{y}^{svm})$, for the simplicity of notations.

The difficulty in solving Eq. 3 lies in the fact that the ground-truth $\mathbf{y}^*$ is unknown. Otherwise it is trivial to output $\mathbf{y} = \mathbf{y}^*$ as the optimal solution. Given a pool of T low-density separators $\{\hat{\mathbf{y}}_t\}_{t=1}^{T}$, as employed by existing S3VMs, here we assume that the ground-truth $\mathbf{y}^*$ is realized by a low-density separator, i.e., $\mathbf{y}^* \in \mathcal{M} \triangleq \{\hat{\mathbf{y}}_t\}_{t=1}^{T}$. Without further domain knowledge in distinguishing these separators, we then maximize the *worst-case* improvement over inductive SVM (Eq. 4), and denote $\bar{\mathbf{y}}$ as the optimal solution.

$$\bar{\mathbf{y}} = \arg\max_{\mathbf{y}\in\{\pm1\}^u} \min_{\hat{\mathbf{y}}\in\mathcal{M}} \; J(\mathbf{y}, \hat{\mathbf{y}}, \mathbf{y}^{svm}). \quad (4)$$

The following theorem shows that by taking the low-density assumption as typical S3VMs, i.e., $\mathbf{y}^* \in \{\hat{\mathbf{y}}_t\}_{t=1}^{T}$, S4VMs are provably safe.

Theorem 1: If $\mathbf{y}^* \in \{\hat{\mathbf{y}}_t\}_{t=1}^{T}$ and $\lambda \geq 1$, the accuracy of $\bar{\mathbf{y}}$ is never worse than that of $\mathbf{y}^{svm}$.

Proof: Note that $\bar{\mathbf{y}}$ is the optimal solution and $J(\mathbf{y}^{svm}, \hat{\mathbf{y}}, \mathbf{y}^{svm})$ is zero for any $\hat{\mathbf{y}}$, we have

$$\min_{\hat{\mathbf{y}}\in\mathcal{M}} J(\bar{\mathbf{y}}, \hat{\mathbf{y}}, \mathbf{y}^{svm}) \geq \min_{\hat{\mathbf{y}}\in\mathcal{M}} J(\mathbf{y}^{svm}, \hat{\mathbf{y}}, \mathbf{y}^{svm}) = 0. \quad (5)$$

Further note that $\mathbf{y}^* \in \mathcal{M}$, we have

$$J(\bar{\mathbf{y}}, \mathbf{y}^*, \mathbf{y}^{svm}) \geq \min_{\hat{\mathbf{y}}\in\mathcal{M}} J(\bar{\mathbf{y}}, \hat{\mathbf{y}}, \mathbf{y}^{svm}). \quad (6)$$

From Eqs. 5 and 6, $J(\bar{\mathbf{y}}, \mathbf{y}^*, \mathbf{y}^{svm}) \geq 0$, i.e., $gain(\bar{\mathbf{y}}, \mathbf{y}^*, \mathbf{y}^{svm}) \geq \lambda \; loss(\bar{\mathbf{y}}, \mathbf{y}^*, \mathbf{y}^{svm})$. Recall that $\lambda \geq 1$, we then have $gain(\bar{\mathbf{y}}, \mathbf{y}^*, \mathbf{y}^{svm}) \geq loss(\bar{\mathbf{y}}, \mathbf{y}^*, \mathbf{y}^{svm})$ and thus the theorem is proved. □

According to Theorem 1, it is easy to get the following proposition.

Proposition 1: If $\mathbf{y}^* \in \{\hat{\mathbf{y}}_t\}_{t=1}^{T}$ and $\lambda \geq 1$, the accuracy of any $\mathbf{y}$ satisfying $\min_{\hat{\mathbf{y}}\in\mathcal{M}} J(\mathbf{y}, \hat{\mathbf{y}}, \mathbf{y}^{svm}) \geq 0$, is never worse than that of $\mathbf{y}^{svm}$.

Simply outputting the predictive results of the inductive SVM would be also safe but evidently not useful. Thus, it is important to study the performance improvement of S4VMs. The following proposition shows that S4VMs achieve the maximal performance improvement in the worst cases.

Proposition 2: If $\mathbf{y}^* \in \{\hat{\mathbf{y}}_t\}_{t=1}^{T}$ and $\lambda = 1$, the accuracy of $\bar{\mathbf{y}}$ achieves the maximal performance improvement over that of $\mathbf{y}^{svm}$ in the worst cases.

It is noteworthy that S4VMs are somewhat relevant to ensemble methods [49], and the spirit of S4VMs is not specific to S3VMs, which may also be extended to other semi-supervised learning methods.

In the following, we will present the optimization of Eq. 4 and an out-of-sample extension of S4VMs in Sections 4.1.1 and 4.1.2, respectively.

4.1.1 Optimization

Note that the $gain(\mathbf{y}, \hat{\mathbf{y}}, \mathbf{y}^{svm})$ and $loss(\mathbf{y}, \hat{\mathbf{y}}, \mathbf{y}^{svm})$ are linear functions with respect to $\mathbf{y}$, i.e.,

$$\begin{aligned} gain(\mathbf{y}, \hat{\mathbf{y}}, \mathbf{y}^{svm}) &= \sum\nolimits_{j=l+1}^{l+u} I(y_j = \hat{y}_j) I(\hat{y}_j \neq y_j^{svm}) \\ &= \sum\nolimits_{j=l+1}^{l+u} \frac{1 + y_j \hat{y}_j}{2} \frac{1 - y_j^{svm} \hat{y}_j}{2}, \\ loss(\mathbf{y}, \hat{\mathbf{y}}, \mathbf{y}^{svm}) &= \sum\nolimits_{j=l+1}^{l+u} I(y_j \neq \hat{y}_j) I(\hat{y}_j = y_j^{svm}) \\ &= \sum\nolimits_{j=l+1}^{l+u} \frac{1 - y_j \hat{y}_j}{2} \frac{1 + y_j^{svm} \hat{y}_j}{2}. \end{aligned}$$

Hence, $J(\mathbf{y}, \hat{\mathbf{y}}_t, \mathbf{y}^{svm})$ is also linear to $\mathbf{y}$ and can be cast as $\mathbf{c}_t' \mathbf{y} + d_t$, where $\mathbf{c}_t = \frac{1}{4}[(1+\lambda)\hat{\mathbf{y}}_t + (\lambda - 1)\mathbf{y}^{svm}]$ and $d_t = \frac{1}{4}[-(1+\lambda)\hat{\mathbf{y}}_t' \mathbf{y}^{svm} + (1-\lambda)]$.

By introducing an additional variable τ, the inner minimization in Eq. 4 can be reformulated as a maximization problem, and Eq. 4 becomes,

$$\begin{aligned} \max_{\mathbf{y}} \max_{\tau} \quad & \tau \\ \text{s. t.} \quad & \tau \leq \mathbf{c}_t'\mathbf{y} + d_t, \forall t = 1, \ldots, T; \;\; \mathbf{y} \in \{\pm 1\}^u. \end{aligned} \tag{7}$$

Though Eq. 7 is still a difficult mixed-integer linear programming problem, according to Proposition 1, optimal solutions are not necessary for achieving safeness. A simple method is then presented. Specifically, we first relax the integer-form of constraint $\{\pm 1\}^u$ into its convex hull $[-1, 1]^u$, and obtain the optimal solution of the resultant convex linear programming problem. We then project it back to an integer solution with the minimum distance. If the objective value of the resultant integer solution is smaller than zero, $\mathbf{y}^{svm}$ is output as the final solution. It is not hard to verify that our solution satisfies Proposition 1.

It is notable that prior knowledge on low-density separators can be easily incorporated into our framework. Specifically, by introducing the dual variables $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_T]' \geq \mathbf{0}$ for the constraints in Eq. 7, one can have the Lagrangian of Eq. 7 as,

$$L(\tau, \mathbf{y}, \boldsymbol{\alpha}) = \tau - \sum\nolimits_{t=1}^{T} \alpha_t(\tau - \mathbf{c}_t'\mathbf{y} - d_t). \tag{8}$$

Setting the partial derivation *w.r.t.* τ to zero, we have,

$$\partial L / \partial \tau = 1 - \sum\nolimits_{t=1}^{T} \alpha_t = 0. \tag{9}$$

With Eq. 9, the inner maximization of Eq. 7 can be replaced by its dual and Eq. 7 becomes,

$$\max\nolimits_{\mathbf{y} \in \{\pm 1\}^u} \min\nolimits_{\sum_{t=1}^{T} \alpha_t = 1, \boldsymbol{\alpha} \geq \mathbf{0}} \sum\nolimits_{t=1}^{T} \alpha_t(\mathbf{c}_t'\mathbf{y} + d_t). \tag{10}$$

Here α_t can be interpreted as a probability that $\hat{\mathbf{y}}_t$ discloses the ground-truth solution. Hence, if prior knowledge about the probabilities $\boldsymbol{\alpha}$ is available, one can readily learn the optimal $\mathbf{y}$ with respect to the target in Eq. 10 using the known $\boldsymbol{\alpha}$.

4.1.2 Out-of-Sample Extension

Eq. 4 works in the transductive setting [40] which could not make predictions on unseen instances. To overcome this, an out-of-sample extension (also named as *induction* extension [52]) of S4VMs is presented.

One common practice to achieve this is to freeze the transductive setting on the set of both testing and unlabeled instances [52]. Formally, for any given testing instance $\mathbf{z}$, let $\{\hat{y}_t^{\mathbf{z}}\}_{t=1}^T$ be the predictive labels of multiple low-density separators, and $y^{svm,\mathbf{z}}$ be the predictive label of the inductive SVM. One need to learn a label assignment for both testing and unlabeled instances such that the objective of S4VM is maximized.

$$\begin{aligned} \max\nolimits_{\mathbf{y} \in \{\pm 1\}^u,\, y^{\mathbf{z}} \in \{\pm 1\},\, \tau} \quad & \tau \\ \text{s. t.} \quad & \tau \leq [\mathbf{c}_t, \mathbf{c}^{\mathbf{z}}]'[\mathbf{y}, y^{\mathbf{z}}] + d_t^{\mathbf{z}}, \forall t = 1, \ldots, T, \end{aligned} \tag{11}$$

where $\mathbf{c}^{\mathbf{z}} = \frac{1}{4}[(1+\lambda)\hat{y}_t^{\mathbf{z}} + (\lambda - 1)y^{svm,\mathbf{z}}]$ and $d_t^{\mathbf{z}} = d_t - \frac{1}{4}(1+\lambda)\hat{y}_t^{\mathbf{z}} y^{svm,\mathbf{z}}$. This, however, will be computationally prohibitive especially when there are a large number of instances for testing.

To alleviate the computational load, we present an efficient algorithm for approximate solutions. Specifically, note that when $y^{\mathbf{z}}$ is fixed to $y^{svm,\mathbf{z}}$, Eq. 11 is equivalent to transductive S4VM, i.e., Eq. 7, and thus the solution of Eq. 7 (denoted by $\check{\mathbf{y}}$) provides a quite good approximation to Eq. 11. This observation motivates us to solve the following much simpler problem instead of the complicated one in Eq. 11,

$$\begin{aligned} \max\nolimits_{y^{\mathbf{z}} \in \{\pm 1\},\, \tau} \quad & \tau \\ \text{s. t.} \quad & \tau \leq [\mathbf{c}_t, \mathbf{c}^{\mathbf{z}}]'[\check{\mathbf{y}}, y^{\mathbf{z}}] + d_t^{\mathbf{z}}, \forall t = 1, \ldots, T. \end{aligned} \tag{12}$$

It is efficient to derive the optimal solution of Eq. 12. We just need to enumerate the two possible values of $y^{\mathbf{z}}$ and then pick up the one with the smaller objective value. As will be validated empirically in Section 5.2, our approximation is quite effective.

4.2 Generating the Pool of Diverse Separators

Denote $h(f, \hat{\mathbf{y}})$ as the objective function of S3VMs in Eq. 1 for the sake of simplicity,

$$h(f, \hat{\mathbf{y}}) = \frac{1}{2}\|f\|_{\mathcal{H}}^2 + C_1 \sum_{i=1}^{l} \ell(y_i, f(\mathbf{x}_i)) + C_2 \sum_{j=l+1}^{l+u} \ell(\hat{y}_j, f(\mathbf{x}_j)).$$

To generate a pool of diverse separators $\{f_t\}_{t=1}^T$ and their corresponding label assignments $\{\hat{\mathbf{y}}_t\}_{t=1}^T$, in this paper we consider to minimize the following function,

$$\min_{\{f_t, \hat{\mathbf{y}}_t \in \mathcal{B}\}_{t=1}^T} \sum\nolimits_{t=1}^{T} h(f_t, \hat{\mathbf{y}}_t) + M\Omega(\{\hat{\mathbf{y}}_t\}_{t=1}^T). \tag{13}$$

Here Ω refers to a penalty reflecting the diversity of separators, i.e., the larger the diversity, the smaller the penalty. M is a large constant (e.g., 10^5 in our experiments) enforcing large diversity. It is easy to realize that minimizing Eq. 13 favors the separators with large margins as well as large diversities.

We consider the penalty as a sum of pairwise terms, i.e., $\Omega(\{\hat{\mathbf{y}}_t\}_{t=1}^T) = \sum_{1 \leq t \neq \tilde{t} \leq T} \delta(\frac{\hat{\mathbf{y}}_t'\hat{\mathbf{y}}_{\tilde{t}}}{u} \geq 1 - \varsigma)$ where δ is the indicator function and $\varsigma \in [0, 1]$ is a constant (e.g., 0.5 in our experiments). It is notable that other penalty quantities can be also applicable.

Recall that $f(\mathbf{x}) = \mathbf{w}'\phi(\mathbf{x}) + b$ is a linear model in S3VMs, where $\phi(\mathbf{x})$ is a feature mapping induced by the kernel κ, i.e., $\kappa(\mathbf{x}, \hat{\mathbf{x}}) = \phi(\mathbf{x})'\phi(\hat{\mathbf{x}})$ and b is a bias term. Eq. 13 then becomes

$$\begin{aligned} \min_{\{\mathbf{w}_t, b_t, \hat{\mathbf{y}}_t \in \mathcal{B}\}_{t=1}^T} \sum_{t=1}^{T} \Big(& \frac{1}{2}\|\mathbf{w}_t\|^2 + C_1 \sum_{i=1}^{l} \ell(y_i, \mathbf{w}_t'\phi(\mathbf{x}_i) + b_t) \\ & + C_2 \sum_{j=l+1}^{l+u} \ell(\hat{y}_{t,j}, \mathbf{w}_t'\phi(\mathbf{x}_j) + b_t) \Big) \\ & + M \sum\nolimits_{1 \leq t \neq \tilde{t} \leq T} \delta(\frac{\hat{\mathbf{y}}_t'\hat{\mathbf{y}}_{\tilde{t}}}{u} \geq 1 - \varsigma). \end{aligned} \tag{14}$$

To address Eq. 14, in the sequel, two implementations are presented. One is based on a global simulated annealing search while the other is based on an efficient sampling strategy.

Algorithm 5 Solving Eq. 14 by Simulated Annealing

Input: $\{\mathbf{x}_i, y_i\}_{i=1}^{l}$, $\{\mathbf{x}_j\}_{j=l+1}^{l+u}$, T; neighbour($\{\hat{\mathbf{y}}_t\}_{t=1}^{T}$) returns $\{\hat{\mathbf{y}}_t^{new}\}_{t=1}^{T}$ where each $\hat{\mathbf{y}}_t^{new}$ has one element different from $\hat{\mathbf{y}}_t$; random() returns a random value in the range (0,1);
Output: $\{\hat{\mathbf{y}}_t\}_{t=1}^{T}$.

Initialize $\{\hat{\mathbf{y}}_t\}_{t=1}^{T}$ randomly, $P \leftarrow 1$, $e \leftarrow 1$, $minP \leftarrow 10^{-8}$, and $emax \leftarrow 300$.
$(\{\hat{\mathbf{y}}_t\}_{t=1}^{T}, o) \leftarrow$ Localsearch($\{\hat{\mathbf{y}}_t\}_{t=1}^{T}$).
$\hat{\mathbf{y}}_t^{best} \leftarrow \hat{\mathbf{y}}_t$, $\forall t = 1, \ldots, T$.
while $P > minP$ **do**
  $\{\hat{\mathbf{y}}_t^{new}\}_{t=1}^{T} \leftarrow$ neighbour($\{\hat{\mathbf{y}}_t\}_{t=1}^{T}$).
  $(\{\hat{\mathbf{y}}_t^{new}\}_{t=1}^{T}, o^{new}) \leftarrow$ Localsearch($\{\hat{\mathbf{y}}_t^{new}\}_{t=1}^{T}$).
  if $o^{new} < o$ **then**
    $o \leftarrow o^{new}$; $\hat{\mathbf{y}}_t^{best} \leftarrow \hat{\mathbf{y}}_t^{new}$, $\hat{\mathbf{y}}_t \leftarrow \hat{\mathbf{y}}_t^{new}$, $\forall t = 1, \ldots, T$.
  else if $\exp(-(o^{new} - o)/P) >$ random() **then**
    $\hat{\mathbf{y}}_t \leftarrow \hat{\mathbf{y}}_t^{new}$, $\forall t = 1, \ldots, T$.
  else
    $e \leftarrow e + 1$.
  end if
  if $e = emax$ **then**
    $P \leftarrow P/2$; $e \leftarrow 1$.
  end if
end while
$\hat{\mathbf{y}}_t \leftarrow \hat{\mathbf{y}}_t^{best}$, $\forall t = 1, \ldots, T$.

Algorithm 6 Localsearch

Input: $\{\hat{\mathbf{y}}_t\}_{t=1}^{T}$; (Denote $[m] = \{1, \ldots, m\}$)
Output: $\{\hat{\mathbf{y}}_t\}_{t=1}^{T}$ and the objective value o.

Fix $\{\hat{\mathbf{y}}_t\}_{t=1}^{T}$, solve multiple individual SVMs $\{\mathbf{w}_t, b_t\}_{t=1}^{T}$ via SVM solver.
Fix $\{\mathbf{w}_t, b_t\}_{t=1}^{T}$, solve $\{\hat{\mathbf{y}}_t\}_{t=1}^{T}$ according to Steps 3-6.
while $\{\hat{\mathbf{y}}_t\}_{t=1}^{T}$ are not converged **do**
  Generate a random permutation of $u \times T$ (denoted by (j, t)'s where $j \in [u]$, $t \in [T]$).
  For each (j, t), optimize $\hat{y}_{t,j+l} \in \{\pm 1\}$ *w.r.t.* Eq. 14.
end while
Output $\{\hat{\mathbf{y}}_t\}_{t=1}^{T}$ and the objective value o of Eq.14.

It is notable that exhaustively searching all possible large-margin low-density separators is prohibitive. Fortunately, according to Theorem 1, generating a large-margin low-density separator to realize the ground-truth is only a sufficient rather than necessary condition to have safe S3VMs. As will be validated in our empirical studies, even on many cases in which the ground-truth is not realized by any of the generated large-margin low-density separators, S4VMs still work quite well.

4.2.1 Global Simulated Annealing Search

Our first implementation to address Eq. 14 is based on global search, e.g., simulated annealing (SA) search [25]. SA is a probabilistic method for approaching global solutions of objective functions which suffer from multiple local minima. Specifically, at each step, SA replaces the current solution by a random nearby solution with a probability. The probability depends on two factors, i.e., the value difference between their corresponding function targets, and a global parameter, i.e., the temperature P, which gradually decreases during the process. When P is large, the current solution almost changes randomly. While as P approaches zero, the changes are increasingly "downhill". In theory, the probability that SA converges to the global solution approaches to 1 as SA procedure is continued [26].

To alleviate the low convergence rate of standard SA, inspired by [37], a deterministic local search scheme is used. Specifically, when $\{\hat{\mathbf{y}}_t\}_{t=1}^{T}$ are fixed, $\{\mathbf{w}_t, b_t\}_{t=1}^{T}$ are solved via multiple individual SVM subroutines. When $\{\mathbf{w}_t, b_t\}_{t=1}^{T}$ are fixed, $\{\hat{\mathbf{y}}_t\}_{t=1}^{T}$ are updated based on local binary search.

Algorithm 5 presents the pseudo-code of our simulated annealing approach for Eq. 14, where the local search subroutine is given in Algorithm 6.

Algorithm 7 Solving Eq. 14 by Representative Sampling

Input: $\{\mathbf{x}_i, y_i\}_{i=1}^{l}$, $\{\mathbf{x}_j\}_{j=l+1}^{l+u}$, T, N;
Output: $\{\hat{\mathbf{y}}_t\}_{t=1}^{T}$.

Randomly sample N number of $\mathbf{y}$'s, i.e., $\mathcal{S} = \{\mathbf{y}_n\}_{n=1}^{N}$.
for $n = 1 : N$ **do**
  while not converged **do**
    Fix $\mathbf{y}_n$, solve $\{\mathbf{w}_n, b_n\}$ via SVM solver.
    Fix $\{\mathbf{w}_n, b_n\}$, update $\mathbf{y}_n$ according to S3VM's objective function via sorting [46].
  end while
end for
Perform clustering (e.g., k-means) for $\mathcal{S}$ where $k = T$.
For each cluster, output the separator (denoted by $\hat{\mathbf{y}}$) with the minimum objective value.

4.2.2 Representative Sampling

To further alleviate the computational burden, our second implementation is based on heuristic representative sampling. Recall that the goal of Eq. 13 can be realized by finding multiple large-margin low-density separators and then keeping only representative ones with large diversity. This motivates us to have a two-stage method, a) search for multiple large-margin low-density separators at first and then b) select the representative separators. Algorithm 7 presents the pseudo-code of our second implementation.

As Algorithm 7 shows, multiple candidate large-margin low-density separators are first obtained by [46]. A clustering algorithm is then applied to identify the representative separators. This approach is simple. As will be validated empirically in Section 5, it is also efficient and effective.

We call our S4VM using simulated annealing as S4VM$_a$, and the one using sampling as S4VM$_s$.

5 Empirical Study

In this section, the proposed approaches are evaluated on a broad range of tasks including five semi-supervised benchmark data sets,[1] *digit1, USPS, BCI, g241c, COIL*, and fifteen UCI data sets[2] and four large scale data sets, *adult, mnist, real-sim, rcv1*. The size of

1. http://www.kyb.tuebingen.mpg.de/ssl-book/
2. http://archive.ics.uci.edu/ml/datasets.html

TABLE 1
Characteristics of the data sets.

Data Sets	# Dim.	# Instance		
		# positive	# negative	total
house	16	108	124	232
heart	9	120	150	270
haberman	14	81	225	306
liverDisorders	6	200	145	345
ionosphere	33	225	126	351
bci	117	200	200	400
house-votes	16	267	168	435
vehicle	16	218	217	435
clean1	166	207	269	476
wdbc	14	357	212	569
isolet	51	300	300	600
breastw	9	239	444	683
austra	15	307	383	690
australian	42	383	307	690
diabetes	8	268	500	768
optdigits	42	572	571	1,143
digit1	241	734	766	1,500
usps	241	300	1,200	1,500
coil	241	750	750	1,500
g241c	241	750	750	1,500
mnist4vs9	629	6,824	6,958	13,782
mnist7vs9	631	7,141	6,825	13,966
mnist3vs8	600	7,293	6,958	14,251
mnist1vs7	652	7,877	7,293	15,170
adult	123	7,841	24,720	32,561
real-sim	20,958	22,238	50,071	72,309
rcv1	47,236	365,951	331,690	697,641

data ranges from 232 to more than $600,000$, and the dimensionality ranges from 6 to more than $40,000$. *mnist* has 45 pairs of binary classification problems, and we focus on its four most difficult pairs [46]. Table 1 summarizes the characteristics of the data sets.

To satisfy the balance constraint required by S3VMs, for each data set, we randomly select 10 instances whose class proportion is closely related to the whole data set, to be served as labeled instances. The remaining data are served as the unlabeled instances. The experiments repeat for 30 times. The average performance and standard deviation are recorded.

Inductive SVM and S3VM serve as the two baseline approaches. For small and medium scale data sets, LIBSVM[3] [18] and TSVM[4] [23] are employed. For large scale data sets, due to the high computational load of LIBSVM and TSVM, efficient LIBLINEAR[5][21] and UniverSVM[6] [15] serve as baselines instead. Both the linear and RBF kernels are used for small and medium scale data sets, and linear kernel is always used for large scale data sets.

Three S3VM variants using multiple low-density separators are also compared. Specifically, S3VM^{best} presents the best performance among the multiple candidate separators (note that this method is impractical). S3VM^{min} selects the low-density separator with minimum objective value. S3VM^{com} combines the candidate separators using uniform weights.

The parameters are set as follows. Following the setups in [10], the regularization parameter C is fixed to 100 and the width of RBF kernel is set to the average distance between instances for inductive SVM. The regularization parameters C_1, C_2 and β in the balance constraint are fixed to 100, 0.1 and 0.1 for all S3VMs and S4VMs. For S3VM-c, the cluster number k is fixed to 50. For S3VM-p, the parameter η is fixed to 0.1 and the similarity matrix is constructed via Gaussian distance where the width is set to the average distance between instances. For S3VM-us, the parameter ϵ is fixed to 0.1. For S4VM$_a$, the number of separators T and the risk parameter λ are both fixed to 3. For S4VM$_s$, the sampling size N, the number of separators T, and the risk parameter λ are fixed to 100, 10 and 3, respectively. The linear program in S4VMs is conducted using the linprog function in MATLAB.

5.1 Comparison Results

Intensive comparison results are shown in Table 2. Although simulated annealing was used to improve the efficiency of S3VMs [37], it still involves high computational load. Table 2 only reports the performance of S4VM$_a$ on 11 small UCI data sets.

Table 2 shows that S4VM$_a$ performs highly competitive with S3VM. Specifically, S3VM significantly outperforms inductive SVM on 5 of the 11 cases with linear kernel, and 7 of the 11 cases with RBF kernel; while S4VM significantly outperforms inductive SVM on 7 cases for both the linear and RBF kernels.

More importantly, unlike S3VM which causes significant degeneration of the performance on 1 case with linear kernel and 2 cases with RBF kernel, S4VM$_a$ is never inferior to inductive SVM. The Wilcoxon sign tests at 95% significance level confirm that S4VM$_a$ is significantly better than inductive SVM with both linear and RBF kernels, but S3VM does not show such a significance.

Table 2 also shows the highly competitive performance of S4VM$_s$ and S3VM-us compared with S3VM. Specifically, in terms of pairwise comparison, S4VM$_s$ is found to be superior to S3VM on 16 of the 27 cases with linear kernel, and 11 of the 20 cases with RBF kernel. S3VM-us is superior to S3VM on 9 and 8 of the 20 cases with linear and RBF kernel, respectively. In terms of *wins*, with linear kernel, S3VM outperforms inductive SVM on 44% (12/27) of the cases; while S4VM$_s$ and S3VM-us outperform inductive SVM on 59% (16/27) and 45% (9/20), respectively. Similar observations can be found for RBF kernel. On 55%, 55% and 50% of the cases, S3VM, S4VM$_s$ and S3VM-us significantly outperform inductive SVM, which are also competitive.

Unlike S3VM whose performance is found to decrease significantly on 3 cases with linear kernel and 6 cases with RBF kernel, S3VM-us shows decreased performance on only one case, and S4VM$_s$ never show decreased performance. Both S3VM-c and S3VM-p are capable of reducing the chance of performance degeneration, but they do not perform as well as S3VM-

3. http://www.csie.ntu.edu.tw/~cjlin/libsvm/
4. http://svmlight.joachims.org/
5. http://www.csie.ntu.edu.tw/~cjlin/linlinear/
6. http://mloss.org/software/view/19/

TABLE 2
Comparison of accuracy (mean±std.). Entries of semi-supervised methods (S3VM, S3VM-c, S3VM-p, S3VM-us, S3VM_s^{best}, S3VM_s^{min}, S3VM_s^{com}, S4VM_a and S4VM_s) are bolded/underlined if they are significantly better/worse than SVM (paired t-tests at 95% significance level). '-' marks cases suffering from high computational cost or memory overhead.

Linear	SVM	S3VM	S3VM-c	S3VM-p	S3VM-us	S3VM_s^{best}	S3VM_s^{min}	S3VM_s^{com}	S4VM_a	S4VM_s
austra	69.9 ± 7.6	69.6 ± 10.8	69.7 ± 8.7	70.0 ± 7.6	69.8 ± 7.9	**71.7 ± 9.5**	70.6 ± 9.7	70.7 ± 9.8	70.7 ± 8.8	70.7 ± 9.5
australian	75.2 ± 8.6	77.4 ± 9.3	**75.8 ± 8.7**	**75.6 ± 8.2**	75.1 ± 8.5	**80.2 ± 6.7**	76.0 ± 10.3	73.5 ± 10.2	-	75.2 ± 8.7
breastw	94.3 ± 2.0	93.3 ± 0.4	94.6 ± 1.2	94.3 ± 1.9	94.3 ± 1.9	**95.9 ± 1.7**	**95.8 ± 1.7**	93.9 ± 3.4	**94.7 ± 1.8**	**95.0 ± 2.0**
clean1	59.0 ± 6.2	57.6 ± 6.8	59.0 ± 6.4	59.2 ± 6.3	59.1 ± 6.1	**64.7 ± 4.2**	57.8 ± 4.6	57.5 ± 6.1	-	59.2 ± 5.3
diabetes	65.5 ± 5.0	64.8 ± 8.3	65.9 ± 5.1	65.7 ± 4.9	**66.1 ± 5.0**	66.2 ± 5.1	65.3 ± 6.0	64.9 ± 5.7	65.5 ± 5.0	65.9 ± 5.4
haberman	63.5 ± 7.6	61.7 ± 5.0	63.9 ± 7.1	**63.8 ± 7.4**	63.7 ± 6.7	64.4 ± 4.6	62.7 ± 4.3	61.9 ± 6.8	-	63.8 ± 5.7
heart	71.1 ± 6.5	73.1 ± 6.5	**71.7 ± 6.6**	**71.7 ± 6.3**	71.6 ± 6.5	**72.4 ± 6.4**	**72.1 ± 6.3**	**71.9 ± 6.2**	71.4 ± 6.7	**72.1 ± 6.3**
house-votes	87.8 ± 3.3	**89.4 ± 4.5**	**88.7 ± 3.3**	**88.1 ± 3.1**	**88.5 ± 3.2**	**91.9 ± 3.8**	**90.3 ± 5.4**	88.9 ± 4.4	**89.0 ± 3.7**	**89.3 ± 3.9**
house	90.1 ± 3.7	**91.9 ± 3.2**	**90.6 ± 3.0**	**90.4 ± 3.4**	91.1 ± 2.6	**95.6 ± 2.9**	**92.6 ± 4.7**	90.2 ± 3.7	**91.8 ± 3.1**	**90.7 ± 4.1**
ionosphere	74.0 ± 5.7	74.5 ± 4.7	**74.8 ± 6.0**	73.8 ± 5.6	74.0 ± 5.6	**79.8 ± 4.5**	75.3 ± 5.2	75.6 ± 5.1	**75.6 ± 6.3**	**76.0 ± 5.6**
isolet	92.3 ± 3.3	**99.7 ± 0.1**	**97.2 ± 2.5**	**93.0 ± 3.0**	**93.0 ± 3.1**	**99.6 ± 0.1**	**99.5 ± 0.1**	**99.4 ± 0.1**	**97.7 ± 0.8**	**98.6 ± 2.7**
liverDisorders	54.3 ± 4.6	53.7 ± 4.9	54.0 ± 4.5	54.5 ± 4.5	54.3 ± 4.6	53.6 ± 4.3	53.2 ± 4.5	52.2 ± 6.3	54.4 ± 4.5	53.5 ± 4.3
optdigits	95.4 ± 2.3	**99.8 ± 0.0**	**98.4 ± 1.5**	**95.8 ± 2.1**	**97.0 ± 1.2**	**99.7 ± 0.1**	**99.7 ± 0.1**	95.3 ± 6.9	-	**98.4 ± 1.9**
vehicle	78.6 ± 6.6	**84.5 ± 9.2**	**82.0 ± 7.3**	**79.9 ± 6.2**	**81.6 ± 7.0**	**84.5 ± 6.6**	**83.2 ± 8.0**	**82.4 ± 8.0**	**81.2 ± 7.2**	**82.4 ± 7.7**
wdbc	85.2 ± 5.7	**91.1 ± 2.8**	**88.3 ± 4.7**	**85.7 ± 5.5**	85.5 ± 5.3	**89.5 ± 5.4**	**89.3 ± 5.4**	**89.1 ± 5.4**	**86.2 ± 5.8**	**89.2 ± 5.5**
digit1	76.4 ± 5.4	**84.3 ± 1.7**	**80.4 ± 6.7**	**78.0 ± 4.8**	**76.6 ± 5.4**	**83.2 ± 2.8**	**81.2 ± 4.3**	69.8 ± 5.8	-	76.4 ± 5.4
usps	78.2 ± 4.9	74.5 ± 5.9	78.2 ± 4.9	78.6 ± 4.6	**79.1 ± 4.2**	**82.7 ± 1.7**	74.7 ± 6.3	77.6 ± 2.4	-	78.6 ± 4.1
coil	58.1 ± 6.1	57.5 ± 5.5	57.9 ± 5.4	57.8 ± 5.9	**58.2 ± 6.1**	**66.9 ± 4.8**	58.8 ± 6.7	56.2 ± 7.0	-	57.9 ± 6.2
bci	54.2 ± 5.6	52.2 ± 3.7	52.8 ± 4. 4	53.9 ± 5.5	54.0 ± 5.5	54.9 ± 4.3	51.8 ± 4.1	51.3 ± 4.5	-	53.5 ± 5.9
g241c	60.0 ± 2.8	**83.7 ± 1.3**	**69.3 ± 4.5**	**62.2 ± 2.5**	**61.2 ± 3.0**	**65.2 ± 3.5**	**65.0 ± 3.9**	49.6 ± 4.5	-	**60.4 ± 2.9**
adult-a	74.8 ± 2.9	74.4 ± 3.1	-	-	-	75.2 ± 3.2	74.3 ± 3.1	74.3 ± 3.1	-	74.7 ± 3.1
mnist1vs7	95.0 ± 2.4	94.9 ± 2.4	-	-	-	**96.5 ± 1.9**	**96.2 ± 2.0**	**96.2 ± 2.0**	-	**96.4 ± 2.3**
mnist3vs8	81.1 ± 6.8	**82.4 ± 6.6**	-	-	-	**84.8 ± 7.0**	**84.2 ± 7.3**	**84.2 ± 7.3**	-	**84.0 ± 7.1**
mnist4vs9	73.9 ± 5.6	**74.7 ± 5.4**	-	-	-	**76.7 ± 6.7**	**75.8 ± 6.9**	**75.8 ± 6.9**	-	**75.8 ± 6.7**
mnist7vs9	79.2 ± 5.9	**80.5 ± 6.2**	-	-	-	**83.5 ± 7.5**	**82.9 ± 7.7**	**82.9 ± 7.7**	-	**82.6 ± 7.5**
real-sim	73.5 ± 2.8	74.0 ± 4.1	-	-	-	**75.5 ± 4.4**	**75.3 ± 4.5**	**75.3 ± 4.5**	-	**75.6 ± 4.1**
rcv1	69.5 ± 5.1	**71.4 ± 4.9**	-	-	-	**73.6 ± 5.7**	**73.5 ± 5.8**	**73.5 ± 5.8**	-	**73.3 ± 5.7**
Win/Tie/Loss against SVM		12 / 12 / 3	11 / 8 / 1	11 / 8 / 1	**9 / 11 / 0**	**22 / 5 / 0**	16 / 10 / 1	9 / 14 / 4	**7 / 4 / 0**	**16 / 11 / 0**

RBF	SVM	S3VM	S3VM-c	S3VM-p	S3VM-us	S3VM_s^{best}	S3VM_s^{min}	S3VM_s^{com}	S4VM_a	S4VM_s
austra	69.2 ± 7.1	70.4 ± 11.9	69.0 ± 8.5	69.1 ± 7.2	69.1 ± 7.5	**76.3 ± 10.1**	70.8 ± 12.0	70.1 ± 12.3	69.2 ± 10.5	**70.6 ± 8.8**
australian	71.4 ± 6.8	**77.7 ± 10.5**	**72.8 ± 7.9**	**71.9 ± 6.7**	71.2 ± 7.2	**80.5 ± 6.7**	71.1 ± 14.4	71.3 ± 10.6	-	71.2 ± 7.1
breastw	95.0 ± 2.4	93.2 ± 0.4	94.9 ± 2.1	95.0 ± 2.4	95.0 ± 2.4	**96.5 ± 0.4**	**96.4 ± 0.4**	**96.3 ± 0.7**	**95.8 ± 1.1**	**95.9 ± 1.5**
clean1	64.3 ± 4.9	60.8 ± 6.9	63.8 ± 5.2	63.9 ± 4.7	64.7 ± 5.0	65.4 ± 4.5	57.9 ± 5.3	60.3 ± 5.9	-	64.4 ± 4.4
diabetes	66.1 ± 4.4	65.1 ± 7.0	66.3 ± 4.2	**66.4 ± 4.3**	**66.4 ± 4.4**	66.0 ± 5.7	65.2 ± 5.5	64.8 ± 5.4	65.8 ± 4.2	65.5 ± 5.5
haberman	65.8 ± 5.4	61.0 ± 3.7	65.8 ± 5.2	65.9 ± 5.3	65.7 ± 5.4	65.0 ± 3.1	62.5 ± 3.3	65.4 ± 3.6	-	66.0 ± 4.2
heart	72.2 ± 5.5	**73.9 ± 5.1**	**72.9 ± 5.5**	72.6 ± 5.3	72.4 ± 5.9	**75.0 ± 5.1**	**73.4 ± 5.8**	**73.4 ± 6.1**	72.9 ± 5.6	**73.5 ± 5.6**
house-votes	87.9 ± 2.4	**89.1 ± 2.0**	**88.4 ± 2.2**	**88.1 ± 2.3**	**88.5 ± 2.2**	**89.4 ± 2.2**	88.5 ± 2.0	88.5 ± 2.4	**89.0 ± 2.3**	88.6 ± 2.2
house	89.3 ± 2.3	**90.4 ± 1.8**	**89.7 ± 2.1**	**89.4 ± 2.2**	**89.8 ± 2.1**	**90.6 ± 2.5**	89.2 ± 2.4	89.5 ± 2.7	**89.7 ± 2.5**	**89.8 ± 2.4**
ionosphere	79.7 ± 5.6	**83.4 ± 5.6**	**80.4 ± 5.4**	79.9 ± 5.6	**80.0 ± 5.7**	**87.2 ± 6.5**	**82.8 ± 6.5**	**82.0 ± 6.4**	**83.0 ± 6.0**	**84.3 ± 6.6**
isolet	91.9 ± 3.1	**99.7 ± 0.1**	**96.8 ± 2.6**	**92.6 ± 2.8**	**92.6 ± 2.8**	**99.2 ± 0.3**	**98.5 ± 0.7**	**98.6 ± 0.5**	**97.1 ± 1.5**	**98.6 ± 0.6**
liverDisorders	55.5 ± 4.7	54.1 ± 4.7	54.8 ± 4.5	55.6 ± 4.7	55.4 ± 4.6	55.6 ± 4.7	55.4 ± 4.7	55.1 ± 4.7	55.6 ± 4.7	55.4 ± 4.7
optdigits	94.6 ± 3.2	**99.7 ± 0.1**	**97.3 ± 2.5**	**95.1 ± 2.8**	**96.6 ± 1.5**	**99.8 ± 0.1**	**99.6 ± 0.9**	**97.5 ± 2.2**	-	**98.0 ± 2.0**
vehicle	80.3 ± 6.2	**84.8 ± 11.5**	**83.2 ± 8.1**	**81.1 ± 6.2**	**82.7 ± 7.2**	**91.1 ± 5.7**	**87.5 ± 8.4**	**84.6 ± 8.7**	**84.5 ± 8.9**	**85.0 ± 7.5**
wdbc	85.3 ± 5.1	**90.7 ± 2.1**	**88.2 ± 4.6**	**85.9 ± 4.9**	85.6 ± 4.9	**91.9 ± 3.7**	**91.2 ± 3.6**	**90.8 ± 3.7**	**89.0 ± 4.0**	**90.7 ± 4.1**
digit1	75.4 ± 8.0	**90.1 ± 3.2**	**80.7 ± 9.2**	**77.1 ± 7.1**	**75.9 ± 8.0**	**91.8 ± 2.0**	**88.5 ± 1.5**	**88.5 ± 3.8**	-	**79.1 ± 5.1**
usps	80.0 ± 0.0	67.9 ± 5.9	80.0 ± 0.0	80.0 ± 0.0	80.0 ± 0.0	77.9 ± 4.7	65.9 ± 0.4	78.2 ± 3.9	-	80.0 ± 0.0
coil	62.0 ± 6.4	61.6 ± 6.1	62.5 ± 6.8	61.2 ± 6.4	**62.1 ± 6.3**	**72.5 ± 7.9**	64.4 ± 9.8	59.9 ± 8.2	-	61.9 ± 6.4
bci	51.5 ± 2.5	50.0 ± 2.0	50.2 ± 2.4	51.4 ± 2.4	51.4 ± 2.4	52.1 ± 2.1	49.8 ± 1.7	48.9 ± 3.0	-	50.8 ± 2.6
g241c	59.8 ± 2.7	**60.8 ± 2.8**	**60.5 ± 2.9**	60.0 ± 2.8	59.9 ± 2.7	**63.7 ± 2.6**	**62.2 ± 3.5**	52.1 ± 4.7	-	**60.2 ± 2.8**
Win/Tie/Loss against SVM		11 / 3 / 6	10 / 8 / 2	9 / 9 / 2	10 / 9 / 1	**14 / 6 / 0**	9 / 6 / 5	8 / 8 / 4	**7 / 4 / 0**	**11 / 9 / 0**

us. S3VM_s^{min} and S3VM_s^{com} still show significantly reduced performance in many cases. The Wilcoxon sign tests at 95% significance level validate S4VM_s and S3VM-us to be significantly better than inductive SVM with both linear and RBF kernels, but other semi-supervised methods, such as S3VM, S3VM-c, S3VM-p, S3VM_s^{min} and S3VM_s^{com}, do not obtain significance.

Although S3VM-us is found to be safer than S3VM, it employs a conservative strategy and its improvement is often much smaller than that of S3VM. In contrast, S4VM_s takes the improvement in performance into account and performs much better. Specifically, in terms of average performance, S4VM_s is superior to S3VM-us. It reaches 75.91% vs S3VM-us's 74.97% on the 40 cases of S3VM-us reported in Table 2. The paired t-tests at 95% significance level show that S4VM_s performed significantly better than S3VM-us. These comparisons confirm that S4VM_s is better than S3VM-us.

The condition of Theorem 1 is already weaker than the traditional low-density assumption in S3VMs, the theorem may not always hold in practice. That is, the ground-truth may not reside among the low-density separators (cf. the performance of S3VM_s^{best}). Even in such cases, S4VMs still work well. That might be because i) Theorem 1 only presents a sufficient rather than necessary condition for safeness, and ii) the analysis of the diversity among low-density separators [39], provides an explanation to S4VMs' superiority to single separator.

TABLE 3
Comparison of accuracy (mean±std.) with out-of-sample extension.

Data	SVM Linear/RBF	S3VM Linear/RBF	$S3VM_s^{min}$ Linear/RBF	$S3VM_s^{com}$ Linear/RBF	$S4VM_s$ Linear/RBF
austra	69.9 ± 8.6/69.3 ± 8.0	68.2 ± 11.4/70.3 ± 12.4	70.3 ± 10.4/70.1 ± 13.3	69.4 ± 10.3/69.6 ± 12.0	69.6 ± 10.1/70.3 ± 10.1
australian	74.6 ± 9.4/70.4 ± 7.8	77.0 ± 9.5/**76.7 ± 11.6**	74.9 ± 11.9/71.3 ± 14.4	73.7 ± 10.0/**73.4 ± 10.7**	74.9 ± 9.3/70.5 ± 8.0
breastw	93.9 ± 2.4/95.0 ± 3.0	93.2 ± 2.0/93.5 ± 1.9	**95.7 ± 2.3/96.6 ± 1.6**	93.5 ± 4.5/**96.5 ± 1.8**	**94.8 ± 2.7/96.2 ± 2.2**
clean1	58.7 ± 6.7/64.8 ± 5.2	56.9 ± 6.0/59.6 ± 7.4	57.9 ± 5.5/60.8 ± 6.4	58.8 ± 7.0/62.4 ± 6.0	58.8 ± 6.7/64.8 ± 5.3
diabetes	66.1 ± 6.0/66.1 ± 5.0	65.1 ± 8.8/65.8 ± 8.6	66.0 ± 7.9/65.6 ± 5.6	66.1 ± 7.8/65.7 ± 5.3	66.3 ± 7.8/65.8 ± 5.2
haberman	63.2 ± 8.4/63.9 ± 7.0	60.0 ± 5.3/59.3 ± 6.3	61.7 ± 4.9/63.3 ± 5.9	62.5 ± 7.7/**65.0 ± 6.1**	62.9 ± 7.8/64.3 ± 6.3
heart	71.2 ± 7.0/72.4 ± 6.1	73.3 ± 6.6/73.4 ± 5.9	71.3 ± 6.7/73.1 ± 6.0	71.2 ± 6.7/72.8 ± 5.8	71.2 ± 6.7/73.3 ± 5.6
house-votes	87.5 ± 4.5/87.9 ± 3.9	88.3 ± 5.8/**89.4 ± 2.9**	89.0 ± 5.8/88.5 ± 3.6	**88.9 ± 4.9**/88.4 ± 3.5	**89.0 ± 4.7**/88.6 ± 3.5
house	90.5 ± 4.5/88.7 ± 4.1	91.1 ± 4.5/**90.6 ± 3.8**	92.4 ± 5.4/89.0 ± 4.6	91.6 ± 3.9/89.2 ± 4.4	**91.8 ± 4.5/89.5 ± 4.1**
ionosphere	75.7 ± 6.2/81.5 ± 5.9	75.3 ± 6.3/83.7 ± 5.7	76.3 ± 7.7/83.5 ± 7.0	**78.1 ± 6.9**/82.9 ± 6.1	**77.5 ± 7.0/85.1 ± 6.1**
isolet	92.3 ± 3.8/91.7 ± 3.4	**98.8 ± 1.1/99.4 ± 0.6**	**99.3 ± 0.6/98.3 ± 1.1**	**98.2 ± 5.7/98.3 ± 1.0**	**98.8 ± 1.8/98.5 ± 0.9**
liverDisorders	54.8 ± 7.1/56.5 ± 6.6	53.3 ± 7.5/55.7 ± 6.8	53.8 ± 7.7/56.6 ± 6.7	52.8 ± 9.7/56.6 ± 6.6	54.2 ± 6.8/56.6 ± 6.7
optdigits	95.3 ± 3.0/94.8 ± 3.3	**99.2 ± 0.7/99.7 ± 0.5**	**99.7 ± 0.3/99.2 ± 1.7**	95.3 ± 8.6/**97.1 ± 2.7**	**98.4 ± 1.8/98.3 ± 1.8**
vehicle	79.3 ± 5.3/81.3 ± 4.7	**83.7 ± 9.2/84.6 ± 10.9**	**83.1 ± 6.1/84.8 ± 9.0**	**82.0 ± 6.5/84.4 ± 6.9**	**81.9 ± 6.1/84.7 ± 6.3**
wdbc	84.5 ± 7.4/84.9 ± 6.7	**89.7 ± 4.1/90.1 ± 3.5**	**86.7 ± 6.6/89.1 ± 4.7**	**86.5 ± 6.6/89.2 ± 4.7**	**86.6 ± 6.7/88.9 ± 4.9**
digit1	75.4 ± 5.3/74.1 ± 7.9	**83.8 ± 2.7/89.4 ± 4.0**	**81.7 ± 4.3/88.2 ± 2.7**	74.0 ± 6.0/**88.0 ± 5.2**	75.4 ± 5.2/**77.7 ± 6.5**
usps	77.9 ± 5.1/79.8 ± 2.4	74.4 ± 5.4/68.2 ± 7.0	76.1 ± 6.3/68.8 ± 2.6	78.5 ± 5.0/78.7 ± 4.6	78.1 ± 4.5/79.8 ± 2.4
coil	58.2 ± 7.0/62.0 ± 6.2	57.2 ± 4.8/61.4 ± 6.6	56.4 ± 4.7/63.1 ± 10.7	56.3 ± 7.9/60.9 ± 9.0	58.2 ± 7.1/62.0 ± 6.2
bci	53.5 ± 8.6/51.7 ± 5.6	50.5 ± 8.1/51.4 ± 4.4	51.5 ± 7.5/50.7 ± 5.6	50.7 ± 7.0/51.2 ± 5.5	52.8 ± 8.0/51.7 ± 5.8
g241c	58.7 ± 4.0/58.7 ± 3.9	**79.9 ± 1.9/63.2 ± 4.4**	**65.4 ± 4.9**/49.5 ± 2.3	54.1 ± 5.1/45.8 ± 14.6	58.7 ± 4.0/58.7 ± 3.9
Win/Tie/Loss against SVM:		15/19/6	13/23/4	13/23/4	**16/24/0**

5.2 Out-of-Sample Extension

Table 3 shows the performance of S4VMs with out-of-sample extension on small and medium scale data sets. For each data set, 75% of instances are used for training, among which 10 are served as labeled data and required to be satisfied by the balance constraint. The remaining instances are used for testing. Experiment repeats for 30 times. The average performance and standard deviation are recorded.

As can be seen from Table 3 that $S4VM_s$ works quite well with out-of-sample extension. Specifically, in terms of *wins*, $S4VM_s$ performs the best in comparison with the other three S3VMs. More importantly, unlike the other S3VMs, such as S3VM, $S3VM_s^{min}$ and $S3VM_s^{com}$, which show significant performance reductions in many cases, $S4VM_s$ is never inferior to inductive SVM. The Wilcoxon sign tests at 95% significance level confirm that $S4VM_s$ is significantly better than inductive SVM with both linear and RBF kernels, and the other three S3VMs do not achieve significance.

5.3 Influence of the Number of Labeled Data

Table 4 shows the performance of $S4VM_s$ under different numbers of labeled examples. As can be seen from Table 4 that $S4VM_s$ is found to be highly competitive with S3VM for each number of labeled examples. Specifically, in terms of *wins*, S3VM obtains significance on 19/20/20 of the 40 cases for 20, 50 and 100 labeled examples, respectively; while $S4VM_s$ outperforms on 20/20/17 cases accordingly. In terms of pairwise comparison (suppose *win*, *tie* and *loss* stand for scores of 1, 0 and -1 for each data set), $S4VM_s$ outscores S3VM on 7 data sets, scores the same as S3VM on 7 data sets, and lower on 6 data sets.

More importantly, in contrast to S3VM that significantly reduces performance on 17 cases, $S4VM_s$ only shows decreased performance on 3 cases which all happen on liverDiscorders with linear kernel. The might be because, in that setting, even the $S3VM_s^{best}$ approach (which always selects the best candidate separator) cannot achieve a comparable performance against the inductive SVM (the accuracies of $S3VM_s^{best}$ are 56.9, 61.2 and 64.5 for 20, 50 and 100 labeled examples, which are all significantly inferior to the inductive SVM). The Wilcoxon sign tests at 95% significance level confirm that $S4VM_s$ is significantly better than the inductive SVM on each number of label examples, whereas S3VM does not show significance.

5.4 Influence of the Number of Unlabeled Data

Table 5 shows the performance of $S4VM_s$ with different numbers of unlabeled instances. As can be seen, similar to the cases in Section 5.3, $S4VM_s$ still performs highly competitive with S3VM, both in terms of the *wins* as well as the pairwise comparison. Furthermore, unlike S3VM which significantly hurts performance on 23 cases, $S4VM_s$ never shows decreased performance. The Wilcoxon sign tests at 95% significance level still conform that $S4VM_s$ is significantly better than inductive SVM on each number of unlabeled instances, and S3VM does not show such a significance.

5.5 Influence of the Balance Constraint

One piece of prior knowledge of S3VMs is the *balance constraint*. Although the balance constraint is often a mild assumption, it might still be violated in some cases. To study the influence of the balance constraint, 10 labeled examples whose class proportion is substantially different from that of remaining unlabeled data, are randomly selected, and the balance constraint is still required for S3VMs and S4VM. Experiments are repeated for 30 times. The average

TABLE 4
Accuracy of SVM and accuracy improvements of S4VM$_s$ and S3VM against SVM on different numbers of labeled data. The accuracy improvement of $algo$ against SVM is calculated by $(acc_{algo} - acc_{svm})$. 'lin' stands for the linear kernel.

Data	20 labeled			50 labeled			100 labeled			Win/Tie/Loss	
	SVM lin/RBF	S3VM lin/RBF	S4VM lin/RBF	SVM lin/RBF	S3VM lin/RBF	S4VM lin/RBF	SVM lin/RBF	S3VM lin/RBF	S4VM lin/RBF	S3VM	S4VM
austra	73.1/75.0	**2.3**/1.5	0.6/**2.8**	79.9/77.3	0.4/**1.8**	0.8/**0.7**	83.3/78.2	-0.5/**0.8**	-0.4/**0.4**	3/3/0	3/3/0
australian	76.6/77.9	0.6/**2.8**	0.4/**0.8**	75.2/79.8	-0.1/**1.0**	**1.6**/**0.9**	79.9/80.8	**0.6**/**1.2**	**1.6**/**0.7**	4/2/0	5/1/0
breastw	95.2/96.0	**1.2**/0.4	**0.5**/**0.5**	94.0/95.2	**1.3**/0.4	**0.5**/**0.3**	94.6/95.0	**0.5**/**0.4**	**0.3**/**0.1**	4/2/0	6/0/0
clean1	64.1/69.0	-1.0/-2.8	0.9/0.0	69.6/76.3	0.3/-0.4	0.1/-0.3	72.8/83.2	0.4/0.1	0.2/0.1	0/5/1	0/6/0
diabetes	68.7/67.3	0.4/1.1	0.2/0.2	72.8/68.9	-0.5/**0.8**	0.1/0.2	74.9/69.5	-0.9/0.1	-0.2/0.9	1/4/1	0/6/0
haberman	66.0/65.6	0.0/-1.2	0.3/-0.2	71.3/67.0	-3.9/-1.5	-0.5/-0.2	72.9/68.6	-2.4/-1.0	0.2/-0.4	0/2/4	0/6/0
heart	72.5/73.1	0.9/**1.3**	1.1/**1.0**	78.9/75.8	-0.6/0.7	-0.2/0.2	80.8/76.5	0.3/0.2	-0.5/0.0	1/5/0	1/5/0
house-votes	88.7/90.0	1.1/**0.8**	0.7/**0.5**	89.6/91.5	**1.0**/**0.6**	**0.4**/0.0	91.0/92.8	**0.7**/0.3	0.2/0.1	4/2/0	2/4/0
house	92.5/90.3	0.0/**1.0**	**0.8**/0.3	95.1/94.0	-0.6/**0.4**	0.1/0.1	92.7/94.2	**0.7**/**0.4**	0.1/0.1	4/1/1	1/5/0
ionosphere	79.4/87.4	1.3/**2.1**	**1.7**/**3.0**	81.7/90.3	-1.5/-0.4	-0.6/**0.5**	84.2/91.6	-1.8/0.0	-0.2/**0.3**	1/3/2	4/2/0
isolet	96.5/96.5	**3.2**/**3.1**	**3.1**/**2.4**	98.7/98.7	**1.0**/**1.0**	**0.9**/**0.4**	99.2/99.4	**0.5**/**0.4**	**0.4**/**0.1**	6/0/0	6/0/0
liverDisorders	59.0/59.7	-2.0/-0.2	-2.4/-0.7	63.1/64.3	-1.6/0.0	-1.9/-0.7	66.4/67.1	-0.7/-0.3	-1.9/0.4	0/4/2	0/3/3
optdigits	97.3/97.3	**2.5**/**2.4**	**1.8**/**1.8**	98.6/98.8	**1.1**/**0.9**	**0.9**/**0.6**	99.2/99.5	**0.5**/**0.2**	**0.4**/**0.2**	6/0/0	6/0/0
vehicle	84.9/88.3	**4.3**/**5.1**	**1.6**/**3.6**	90.4/94.6	**1.3**/**2.4**	**0.3**/**1.0**	93.5/97.8	**0.6**/**0.7**	-0.2/**0.1**	6/0/0	5/1/0
wdbc	89.8/89.8	**4.3**/**3.7**	**0.5**/**1.3**	91.8/91.6	**1.0**/**1.4**	-0.2/**0.4**	95.3/93.8	0.4/**0.7**	-0.4/0.0	5/1/0	3/3/0
digit1	83.4/84.0	**2.9**/**7.1**	0.1/**4.5**	88.7/91.2	**1.2**/**2.9**	**0.3**/**0.9**	90.9/94.5	**2.0**/**0.9**	**0.6**/**0.4**	6/0/0	5/1/0
usps	82.3/80.1	-3.4/-2.2	0.0/0.1	85.4/80.7	-1.1/**6.3**	**0.4**/**6.4**	86.9/83.3	-0.2/**8.3**	**0.5**/**7.4**	2/2/2	4/2/0
coil	66.1/68.8	0.8/-2.1	0.1/0.0	74.7/80.2	-0.1/-1.6	0.1/0.3	80.4/87.1	0.6/-0.6	0.2/0.0	0/6/0	0/6/0
bci	56.2/53.8	-1.1/-2.5	-1.1/-0.9	62.4/55.9	-1.9/-2.3	-0.6/0.4	68.5/61.6	0.0/-0.9	2.1/1.0	0/2/4	0/6/0
g241c	65.3/65.3	**18.0**/**1.2**	**0.3**/**1.2**	70.5/71.6	**11.6**/**1.4**	**0.3**/**1.5**	73.7/76.8	**6.6**/**0.8**	**0.4**/**0.9**	6/0/0	6/0/0
Win/Tie/Loss against SVM:		19/17/4	**20/19/1**	-	20/12/8	**20/19/1**	-	20/15/5	**17/22/1**	59/44/17	**57/60/3**

TABLE 5
Accuracy of SVM and accuracy improvements of S4VM$_s$ and S3VM on different numbers of unlabeled data.

Data	40% unlabeled			60% unlabeled			80% unlabeled			Win/Tie/Loss	
	SVM lin/RBF	S3VM lin/RBF	S4VM lin/RBF	SVM lin/RBF	S3VM lin/RBF	S4VM lin/RBF	SVM lin/RBF	S3VM lin/RBF	S4VM lin/RBF	S3VM	S4VM
austra	69.9/69.2	-0.7/**2.6**	0.7/**1.8**	70.2/69.3	-0.7/2.0	0.9/**1.5**	70.0/69.3	0.1/2.0	0.7/0.8	1/5/0	2/4/0
australian	75.0/70.6	**2.5**/**6.6**	0.5/**1.0**	75.3/71.3	**2.6**/**6.7**	0.2/0.4	75.3/71.5	**2.7**/**5.9**	0.1/0.5	6/0/0	1/5/0
breastw	94.3/95.0	-1.0/-1.7	**0.7**/**1.1**	94.3/95.0	-1.1/-1.8	**0.9**/**1.4**	94.3/95.0	-1.0/-1.7	**0.8**/**1.1**	0/0/6	6/0/0
clean1	59.7/64.2	-1.3/-2.8	0.0/-0.7	59.2/64.0	-1.4/-4.5	-0.2/-0.5	59.0/64.2	-1.7/-3.8	0.2/-0.5	0/2/4	0/6/0
diabetes	66.0/66.1	-1.4/-1.5	0.3/-0.6	65.4/65.9	-1.3/-0.7	0.2/-0.3	65.5/66.0	-1.1/-0.9	0.1/-0.4	0/5/1	0/6/0
haberman	63.6/66.0	-1.1/-4.7	-0.3/0.2	63.7/66.0	-1.9/-4.9	0.3/0.2	63.6/65.9	-2.0/-5.1	0.1/0.0	0/3/3	0/6/0
heart	71.9/73.0	1.0/-0.1	0.1/0.3	72.1/73.1	1.2/0.1	0.4/0.4	71.4/72.5	1.1/0.6	0.8/0.3	0/6/0	0/6/0
house-votes	88.2/88.0	**1.4**/1.1	**1.0**/**1.1**	87.9/87.9	**1.2**/**1.0**	**1.1**/**1.0**	87.9/87.9	1.0/**1.2**	**1.2**/**1.0**	4/2/0	6/0/0
house	89.4/88.7	0.3/**1.1**	**1.3**/**0.5**	89.8/88.9	0.8/0.4	**1.2**/0.6	90.0/89.1	**1.5**/**0.9**	**0.5**/**0.4**	3/3/0	5/1/0
ionosphere	74.9/80.2	-0.5/**2.5**	1.3/**2.7**	74.5/79.4	-0.6/**4.0**	1.4/**4.1**	73.9/79.4	0.7/**3.6**	**2.1**/**4.3**	3/3/0	4/2/0
isolet	92.1/91.9	**5.9**/**5.9**	**6.9**/**5.5**	92.2/91.9	**6.5**/**6.6**	**7.0**/**6.2**	92.3/91.9	**6.6**/**7.0**	**6.5**/**6.6**	6/0/0	6/0/0
liverDisorders	53.5/54.7	-1.7/-0.9	-0.3/-0.1	54.0/55.0	-1.1/-1.1	-0.6/-0.1	54.4/55.2	-1.2/-1.4	-0.6/-0.2	0/3/3	0/6/0
optdigits	95.4/94.6	**3.2**/**4.0**	**3.1**/**3.4**	95.3/94.6	**3.8**/**4.5**	**3.5**/**3.4**	95.3/94.6	**4.2**/**4.9**	**3.5**/**3.7**	6/0/0	6/0/0
vehicle	78.6/80.0	**5.2**/**3.9**	**2.0**/**3.4**	78.7/80.2	**5.6**/**5.3**	**3.0**/**4.5**	78.8/80.3	**6.3**/**4.4**	**3.4**/**5.0**	6/0/0	6/0/0
wdbc	85.1/85.4	**5.5**/**4.5**	**2.6**/**3.7**	85.1/85.4	**6.1**/**5.6**	**3.0**/**4.6**	85.1/85.3	**5.4**/**5.4**	**3.5**/**5.0**	6/0/0	6/0/0
digit1	76.1/75.3	**7.0**/**11.5**	0.1/**4.6**	76.5/75.6	**7.5**/**13.2**	0.3/**4.9**	76.7/75.7	**8.1**/**13.8**	0.2/**4.7**	6/0/0	3/3/0
usps	78.4/80.3	-4.0/-9.4	0.4/0.3	78.5/80.3	-3.8/-11.8	0.6/0.1	78.1/80.0	-3.6/-12.2	0.4/0.0	0/2/4	0/6/0
coil	57.9/61.9	0.1/-0.5	0.0/0.0	57.8/61.9	-0.1/0.0	0.2/-0.1	57.9/61.9	-0.3/-0.5	0.0/0.0	0/6/0	0/6/0
bci	54.0/51.5	-0.6/-1.1	0.0/-1.0	54.4/51.6	-1.1/-1.2	0.2/0.0	54.0/51.4	-1.5/-1.5	-0.3/-0.4	0/4/2	0/6/0
g241c	60.3/60.1	**17.0**/**0.8**	0.1/0.1	60.4/60.2	**20.7**/**0.7**	0.1/0.0	60.3/60.1	**22.9**/**0.9**	0.0/**0.4**	6/0/0	1/5/0
Win/Tie/Loss against SVM:		18/14/8	**18/22/0**	-	17/17/6	**16/24/0**	-	18/13/9	**18/22/0**	53/44/23	**52/68/0**

performance and standard deviation on UCI data sets with linear kernel are reported in Table 6.

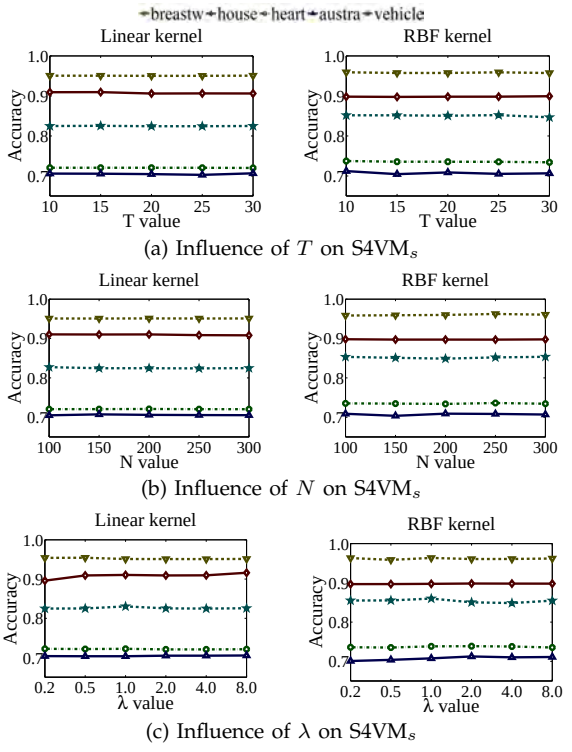
The results show that both the S4VM$_s$ and S3VM perform much worse than those without the violation of the balance constraint (cf. results in Table 2). Moreover, although S4VM$_s$ has already substantially improved the safeness of S3VM, it still shows significant decrease performance on 2 cases. This suggests that, in the cases in which the class proportion of unlabeled instances cannot be estimated using existing labeled examples, it is still challenging to have safe S3VMs.

5.6 Influence of Parameters

S4VM$_s$ has four parameters, i.e., sampling size N, cluster number T, risk parameter λ and the kernel

TABLE 6
Comparison of accuracy (mean±std.) when the balance constraint is violated.

Data	SVM	S3VM	S4VM$_s$
austra	68.1 ± 9.1	64.7 ± 10.9	66.8 ± 10.4
australian	69.7 ± 10.2	70.5 ± 13.1	69.3 ± 9.9
breastw	94.8 ± 3.0	85.6 ± 8.1	91.1 ± 6.5
clean1	57.4 ± 5.3	57.1 ± 4.4	57.4 ± 5.2
diabetes	65.3 ± 6.3	64.4 ± 6.5	64.7 ± 6.3
haberman	63.2 ± 7.2	61.6 ± 5.5	63.8 ± 6.6
heart	71.9 ± 7.7	71.3 ± 8.0	72.1 ± 7.7
house-votes	86.7 ± 4.5	85.4 ± 5.0	87.1 ± 3.7
house	89.2 ± 4.7	85.2 ± 9.3	88.5 ± 7.8
ionosphere	70.9 ± 6.7	71.4 ± 10.0	71.3 ± 8.2
isolet	90.6 ± 4.9	87.3 ± 8.2	**92.7 ± 5.9**
liverDisorders	56.6 ± 5.3	54.6 ± 5.6	54.8 ± 5.7
optdigits	93.1 ± 4.5	88.2 ± 8.4	93.1 ± 7.8
vehicle	76.4 ± 8.2	**79.2 ± 8.9**	76.5 ± 9.7
wdbc	81.8 ± 7.5	**86.6 ± 5.0**	**84.2 ± 6.4**
S3VMs vs SVM: Win/Tie/Loss		2/7/6	**2/11/2**



(a) Influence of T on S4VM$_s$

(b) Influence of N on S4VM$_s$

(c) Influence of λ on S4VM$_s$

Fig. 2. Parameter influence with 10 labeled examples.

type to set. In previous empirical studies, N, T and λ are set as default values, i.e., 100, 10 and 3. Figure 2 further studies the influence of N, T and λ with linear and RBF kernels on five representative data sets (the results on other data sets are similar) with 10 labeled examples by fixing other parameters as default values.

It can be seen that, though the number of labeled examples is small, the performance of S4VM$_s$ is quite insensitive to the setting of the parameters. One possible reason is that, rather than simply picking one low-density separator, S4VMs optimize the assignment of labels in the worst cases. This property makes S4VM$_s$ even more attractive, especially when the number of labeled examples is too small to afford a reliable model selection. Moreover, paired t-tests at 95% significance level confirm that S4VM$_s$ does not reduce performance on all the cases in Figure 2(a)-(b) and Figure 2(c) when $\lambda \geq 1$.

5.7 Running Time

Following the setup in Section 5.2, Figure 3 gives the training and testing time of S3VM and S4VM$_s$ with linear kernel on UCI data sets. S4VM$_s$ runs approximately 10 times of S3VM. That is because S4VM$_s$ needs to generate T low-density separators, where T is usually a small constant (such as 10 in our experiments). It is notable that the implementation of S4VM$_s$ is inherently parallelizable, and thus S4VM$_s$ can be accelerated by parallel implementations or by using more efficient S3VM solvers.

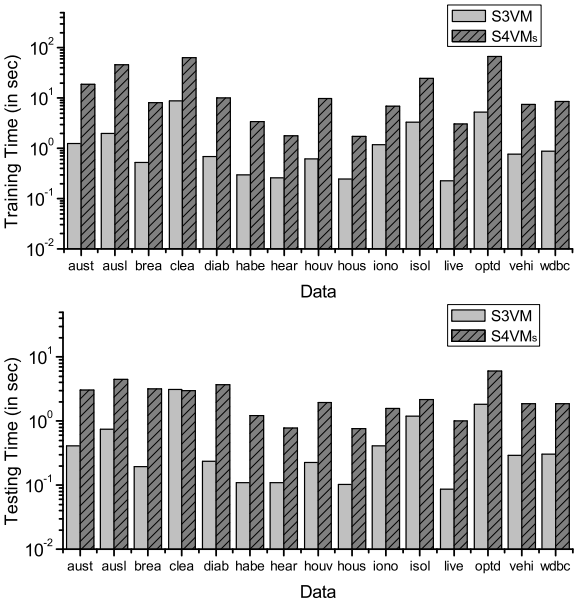


Fig. 3. Training and testing time (in seconds) of S3VM and S4VM$_s$ on UCI data sets with linear kernel.

TABLE 7
Accuracy of other S3VMs (mean±std.).

Data	SVM	LapSVM	LDS	S4VM$_s$
austra	69.2 ± 7.1	65.1 ± 5.4	73.7 ± 14.8	**70.6 ± 8.8**
australian	71.4 ± 6.8	73.0 ± 7.7	**76.7 ± 12.7**	71.2 ± 7.1
breastw	95.0 ± 2.4	93.2 ± 1.9	**96.2 ± 0.6**	**95.9 ± 1.5**
clean1	64.9 ± 4.1	51.5 ± 3.9	55.0 ± 6.1	64.4 ± 4.4
diabetes	66.1 ± 4.4	66.7 ± 4.5	64.2 ± 6.9	65.5 ± 5.5
haberman	65.8 ± 5.4	58.0 ± 8.4	64.3 ± 3.0	66.0 ± 4.2
heart	72.2 ± 5.5	**74.7 ± 5.5**	75.1 ± 7.3	**73.5 ± 5.6**
house-votes	87.9 ± 2.4	86.8 ± 3.2	**89.3 ± 0.8**	88.6 ± 2.2
house	89.3 ± 2.3	89.7 ± 1.4	**90.2 ± 1.8**	**89.8 ± 2.4**
ionosphere	79.7 ± 5.6	69.2 ± 5.3	**89.1 ± 5.5**	**84.3 ± 6.6**
isolet	91.9 ± 3.1	90.1 ± 6.8	**99.2 ± 0.1**	**98.6 ± 0.6**
liverDisorders	55.5 ± 4.7	53.0 ± 5.0	52.1 ± 3.9	55.4 ± 4.7
optdigits	94.6 ± 3.2	93.2 ± 4.5	**99.5 ± 0.0**	**98.0 ± 2.0**
vehicle	80.3 ± 6.2	78.4 ± 7.7	**90.4 ± 12.2**	**85.0 ± 7.5**
wdbc	85.3 ± 5.1	83.2 ± 6.2	**92.7 ± 0.5**	**90.7 ± 4.1**
S3VMs vs SVM: Win/Tie/Loss		1/6/8	9/4/2	9/6/0

5.8 Comparison with Other S3VMs

Table 7 shows the accuracy of other S3VM implementations. Specifically, Laplacian SVM (LapSVM) [2][7] which incorporates manifold assumption into S3VMs, and Low Density Separation (LDS) [12][8] which first introduces a graph-based distance for instances and then optimizes the objective of S3VM with the gradient descent method, are compared with inductive SVM. The parameters γ_A, γ_I of LapSVM are set to the same as the parameters C_1 and C_2 in S3VMs and S4VMs (i.e., 100 and 0.1). The ρ in LDS is set to 4 which achieves the best performance reported in the paper. Since LDS is based on RBF kernel, RBF kernel is used for inductive SVM and LapSVM. The other parameters are with the default settings recommended by the paper. As shown in Table 7, similar to TSVM [23], other S3VM implementations

7. http://manifold.cs.uchicago.edu/manifold_regularization/software
8. http://olivier.chapelle.cc/lds/

like LapSVM and LDS also decrease the performance significantly in some cases.

6 Conclusion

The purpose of this paper is to develop safe semi-supervised support vector machines (S3VMs) which never perform significantly inferior to inductive SVMs that only use labeled data. Based on our preliminary works in [31], [32], this paper first proposes the S3VM-us approach. This approach uses only the unlabeled instances that are very likely to be helpful, and thus avoids the use of highly risky unlabeled instances. Our empirical studies show that this approach improves safeness but only improves the performance slightly, usually much less than S3VMs. To develop a safe and well-performing approach, we re-examine the fundamental assumption of S3VMs, i.e., low-density separation. Based on the observation that multiple low-density separators can be identified from training data, S4VMs (Safe S3VMs) approach, the main contribution of this paper, is proposed. This approach attempts to avoid the risk of using a poor separator. Under the low-density assumption used by S3VMs, S4VMs are found to be provably safe and to achieve the maximum improvement in performance. An out-of-sample extension of S4VMs is also presented so that S4VMs can make predictions on unseen instances. Our empirical studies on a broad range of data sets show that the overall performance of S4VMs is highly competitive with S3VMs, but unlike S3VMs which show significant reduced performance in many cases, S4VMs are rarely inferior to inductive SVMs.

Our empirical studies in Table 2 reveal that even when low-density assumption does not hold, S4VMs still work well. We conjecture that this is because S4VMs exploit multiple separators rather than relying on a single separator. In this way, its robustness benefits from an inherent ensemble learning mechanism [49]. Further study on this issue is an interesting future work. It is also possible to combine the advantages of S3VM-us and S4VMs to develop approaches that are even stronger than the current S4VMs. Moreover, extending the spirit of S4VMs to graph-based semi-supervised methods [2], [33], [45], [53], as well as connecting the safeness to the generalization are worth studying in the future.

Acknowledgments

The authors want to thank the associate editor and reviewers for helpful comments and suggestions. We thank Teng Zhang for help in some experiments. We thank Jianxin Wu and Cam-Tu Nguyen for proofreading the paper. This research was partially supported by the National Fundamental Research Program of China (2014CB340501), the National Science Foundation of China (61333014, 61021062) and the program for outstanding PhD candidate of Nanjing University. Z.-H. Zhou is the corresponding author of this paper.

References

[1] M. F. Balcan and A. Blum. A discriminative model for semi-supervised learning. *Journal of the ACM*, 57(3), 2010.

[2] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

[3] S. Ben-David, T. Lu, and D. Pál. Does unlabeled data provably help? worst-case analysis of the sample complexity of semi-supervised learning. In *Proceedings of the 21st Annual Conference on Learning Theory*, pages 33–44, Helsinki, Finland, 2008.

[4] K. Bennett and A. Demiriz. Semi-supervised support vector machines. In *Advances in Neural Information Processing Systems 11*, pages 368–374. MIT Press, 1999.

[5] J. C. Bezdek and R. J. Hathaway. Convergence of alternating optimization. *Neural, Parallel & Scientific Computations*, 11(4):351–368, 2003.

[6] T. De Bie and N. Cristianini. Convex methods for transduction. In *Advances in Neural Information Processing Systems 16*, pages 73–80. MIT Press, 2004.

[7] A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *Proceedings of the 8th International Conference on Machine Learning*, pages 19–26, Williamstown, MA, 2001.

[8] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the 7th Annual Conference on Computational Learning Theory*, Madison, WI, 1998.

[9] O. Chapelle, M. Chi, and A. Zien. A continuation method for semi-supervised SVMs. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 185–192, Pittsburgh, PA, 2006.

[10] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, 2006.

[11] O. Chapelle, V. Sindhwani, and S. S. Keerthi. Optimization techniques for semi-supervised support vector machines. *Journal of Machine Learning Research*, 9:203–233, 2008.

[12] O. Chapelle and A. Zien. Semi-supervised classification by low density separation. In *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, pages 57–64, Savannah Hotel, Barbados, 2005.

[13] N. Chawla and G. Karakoulas. Learning from labeled and unlabeled data: An empirical study across techniques and domains. *Journal of Artificial Intelligence Research*, 23:331–366, 2005.

[14] K. Chen and S. Wang. Semi-supervised learning via regularized boosting working on multiple semi-supervised assumptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):129–143, 2011.

[15] R. Collobert, F. Sinz, J. Weston, and L. Bottou. Large scale transductive SVMs. *Journal of Machine Learning Research*, 7:1687–1712, 2006.

[16] F. G. Cozman, I. Cohen, and M. C. Cirelo. Semi-supervised learning of mixture models. In *Proceedings of the 20th International Conference on Machine Learning*, pages 99–106, Washington, DC, 2003.

[17] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society - B*, pages 1–38, 1977.

[18] R. E. Fan, P. H. Chen, and C. J. Lin. Working set selection using second order information for training support vector machines. *Journal of Machine Learning Research*, 6:1889–1918, 2005.

[19] C. Goutte, H. Déjean, E. Gaussier, N. Cancedda, and J. M. Renders. Combining labelled and unlabelled data: A case study on fisher kernels and transductive inference for biological entity recognition. In *Proceedings of the 6th Conference on Natural Language Learning*, pages 1–7, Stroudsburg, PA, 2002.

[20] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems 17*. MIT Press.

[21] C. J. Hsieh, K. W. Chang, C. J. Lin, S. S. Keerthi, and S. Sundararajan. A dual coordinate descent method for large-scale linear SVM. In *Proceedings of the 25th International Conference on Machine Learning*, pages 408–415, Helsinki, Finland, 2008.

[22] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, NJ., 1988.

[23] T. Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the 16th International Conference on Machine Learning*, pages 200–209, Bled, Slovenia, 1999.

[24] N. Kasabov and S. Pang. Transductive support vector machines and applications in bioinformatics for promoter recognition. In *Proceedings of the International Conference on Neural Networks and Signal Processing*, pages 1–6, Nanjing, China, 2003.

[25] S. Kirkpatrick. Optimization by simulated annealing: Quantitative studies. *Journal of Statistical Physics*, 34(5):975–986, 1984.

[26] P. J. M. Laarhoven and E. H. L. Aarts. *Simulated Annealing: Theory and Applications*. Springer, 1987.

[27] M. Li and Z.-H. Zhou. SETRED: Self-training with editing. In *Proceedings of the 9th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 611–621, Hanoi, Vietnam, 2005.

[28] Y.-F. Li, J. T. Kwok, and Z.-H. Zhou. Semi-supervised learning using label mean. In *Proceedings of the 26th International Conference on Machine learning*, pages 633–640, Montreal, Canada, 2009.

[29] Y.-F. Li, I. W. Tsang, J. T. Kwok, and Z.-H. Zhou. Tighter and convex maximum margin clustering. In *Proceeding of the 12th International Conference on Artificial Intelligence and Statistics*, pages 344–351, Clearwater Beach, FL, 2009.

[30] Y.-F. Li, I. W. Tsang, J. T. Kwok, and Z.-H. Zhou. Convex and scalable weakly labeled svms. *Journal of Machine Learning Research*, 14:2151–2188, 2013.

[31] Y.-F. Li and Z.-H. Zhou. Improving semi-supervised support vector machines through unlabeled instances selection. In *Proceedings of 25th AAAI Conference on Artificial Intelligence*, pages 386–391, San Francisco, CA, 2011.

[32] Y.-F. Li and Z.-H. Zhou. Towards making unlabeled data never hurt. In *Proceedings of the 28th International Conference on Machine learning*, pages 1081–1088, Bellevue, WA, 2011.

[33] W. Liu, J.-F. He, and S.-F. Chang. Large graph construction for scalable semi-supervised learning. In *Proceedings of the 27th International Conference on Machine Learning*, pages 679–686, Haifa, Israel, 2010.

[34] W. Liu, J. Wang, and S.-F. Chang. Robust and scalable graph-based semisupervised learning. *Proceedings of the IEEE*, 100(9):2624–2638, 2012.

[35] D. J. Miller and H. S. Uyar. A mixture of experts classifier with learning based on both labelled and unlabelled data. In *Advances in Neural Information Processing Systems 9*, pages 571–577. MIT Press, 1997.

[36] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine learning*, 39(2):103–134, 2000.

[37] V. Sindhwani, S. S. Keerthi, and O. Chapelle. Deterministic annealing for semi-supervised kernel machines. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 841–848, Pittsburgh, PA, 2006.

[38] A. Singh, R. Nowak, and X. Zhu. Unlabeled data: Now it helps, now it doesn't. In *Advances in Neural Information Processing Systems 21*, pages 1513–1520. MIT Press, 2009.

[39] E. K. Tang, P. N. Suganthan, and X. Yao. An analysis of diversity measures. *Machine Learning*, 65(1):247–271, 2006.

[40] V. N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[41] J. Wang, T. Jebara, and S.-F. Chang. Graph transduction via alternating minimization. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1144–1151, Helsinki, Finland, 2008.

[42] L. Wang, K. Chan, and Z. Zhang. Bootstrapping SVM active learning by incorporating unlabelled images for image retrieval. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 629–634, Madison, WI, 2003.

[43] L. Xu and D. Schuurmans. Unsupervised and semi-supervised multi-class support vector machines. In *Proceedings of 20th National Conference on Artificial Intelligence*, pages 904–910, Pittsburgh, PA, 2005.

[44] A.L. Yuille and A. Rangarajan. The concave-convex procedure. *Neural Computation*, 15(4):915–936, 2003.

[45] K. Zhang, J. T. Kwok, and B. Parvin. Prototype vector machine for large scale semi-supervised learning. In *Proceedings of the 26th International Conference on Machine Learning*, pages 1233–1240, Montreal, Canada, 2009.

[46] K. Zhang, I. W. Tsang, and J. T. Kwok. Maximum margin clustering made practical. In *Proceedings of the 24th International Conference on Machine Learning*, pages 1119–1126, Corvallis, OR, 2007.

[47] T. Zhang and F. Oles. The value of unlabeled data for classification problems. In *Proceedings of the 17th International Conference on Machine Learning*, pages 1191–1198, Stanford, CA, 2000.

[48] D. Zhou, O. Bousquet, T. Navin Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, pages 595–602. MIT Press, 2004.

[49] Z.-H. Zhou. *Ensemble Methods: Foundations and Algorithms*. Chapman & Hall, Boca Raton: FL, 2012.

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

[51] Z.-H. Zhou and M. Li. Semi-supervised learning by disagreement. *Knowledge and Information Systems*, 24(3):415–439, 2010.

[52] X. Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison, 2007.

[53] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *Proceedings of the 20th International Conference on Machine learning*, pages 912–919, Washington, DC, 2003.

Yu-Feng Li received the BSc and PhD degrees in computer science from Nanjing University, China, in 2006 and 2013, respectively. He joined the Department of Computer Science & Technology at Nanjing University as an assistant researcher in 2013. His main research interests include machine learning and data mining. He has won the Microsoft Fellowship Award (2009). He has been a Program Committee member of several conferences including IJCAI'11, ICDM'11, AAAI'12 (special track on AI and the web), IJCNN'13 and ICML'14.

Zhi-Hua Zhou (S'00-M'01-SM'06) received the BSc, MSc and PhD degrees in computer science from Nanjing University, China, in 1996, 1998 and 2000, respectively, all with the highest honors. He joined the Department of Computer Science & Technology at Nanjing University as an assistant professor in 2001, and is currently professor and Director of the LAMDA group. His research interests are mainly in artificial intelligence, machine learning, data mining, pattern recognition and multimedia information retrieval. In these areas he has published more than 100 papers in leading international journals or conference proceedings, and holds 12 patents. He has won various awards/honors including the IEEE CIS Outstanding Early Career Award, the National Science & Technology Award for Young Scholars of China, the Fok Ying Tung Young Professorship Award, the Microsoft Young Professorship Award and nine international journals/conferences paper or competition awards. He is an Associate Editor-in-Chief of the *Chinese Science Bulletin*, Associate Editor of the *ACM Transactions on Intelligent Systems and Technology* and on the editorial boards of various other journals. He is the founder and Steering Committee Chair of ACML, and Steering Committee member of PAKDD and PRICAI. He serves/ed as General Chair/Co-chair of ACML'12, ADMA'12 and PCM'13, Program Chair/Co-Chair for PAKDD'07, PRICAI'08, ACML'09, SDM'13, etc., Workshop Chair of KDD'12, Program Vice Chair or Area Chair of various conferences, and chaired many domestic conferences in China. He is the Chair of the Machine Learning Technical Committee of the Chinese Association of Artificial Intelligence, Chair of the Artificial Intelligence & Pattern Recognition Technical Committee of the China Computer Federation, Vice Chair of the Data Mining Technical Committee of IEEE Computational Intelligence Society and the Chair of the IEEE Computer Society Nanjing Chapter. He is a fellow of the IAPR, the IEEE, and the IET/IEE.