Towards Making Unlabeled Data Never Hurt

Yu-Feng Li LIYF@LAMDA.NJU.EDU.CN
Zhi-Hua Zhou ZHOUZH@LAMDA.NJU.EDU.CN
National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China

Abstract

It is usually expected that, when labeled data are limited, the learning performance can be improved by exploiting unlabeled data. In many cases, however, the performances of current semi-supervised learning approaches may be even worse than purely using the limited labeled data. It is desired to have *safe* semi-supervised learning approaches which never degenerate learning performance by using unlabeled data. In this paper, we focus on semi-supervised support vector machines (S3VMs) and propose S4VMs, i.e., safe S3VMs. Unlike S3VMs which typically aim at approaching an optimal low-density separator, S4VMs try to exploit the candidate low-density separators simultaneously to reduce the risk of identifying a poor separator with unlabeled data. We describe two implementations of S4VMs, and our comprehensive experiments show that the overall performance of S4VMs are highly competitive to S3VMs, while in contrast to S3VMs which degenerate performance in many cases, S4VMs are never significantly inferior to inductive SVMs.

1. Introduction

During the past decade, many effective semi-supervised learning approaches have been developed (Chapelle et al., 2006b; Zhu, 2006; Zhou & Li, 2010). It is expected that, when labeled data are limited, the use of unlabeled data will help improve the performance. However, it has been found that the performances of current semi-supervised learning approaches may be even worse than purely using labeled data in many cases (Nigam et al., 2000; Cozman et al., 2003; Grandvalet & Bengio, 2005). It is very desired to have *safe* semi-supervised learning approaches which never degenerate performance by using unlabeled data.

Among popular semi-supervised learning approaches, S3VMs (Vapnik, 1998; Bennett & Demiriz, 1999; Joachims, 1999) are based on the low-density assumption and try to learn a low-density separator which favors the decision boundary going across low-density regions in the feature space (Chapelle & Zien, 2005). These approaches have already been applied to diverse applications such as text classification (Joachims, 1999), image retrieval (Wang et al., 2003), bioinformatics (Kasabov & Pang, 2004), natural language processing (Goutte et al., 2002), etc. Similar to other semi-supervise approaches, however, it has been found that S3VMs may degenerate the performance by using unlabeled data (Zhang & Oles, 2000; Wang et al., 2003; Chapelle et al., 2006b; 2008).

To address this problem, in this paper we present the S4VMs (safe S3VMs). In contrast to common S3VMs which typically focus on approaching one optimal low-density separator, S4VMs try to exploit multiple candidate low-density separators. Our motivation lies in the observation that, given a few labeled data and abundant unlabeled data, there usually exist more than one large-margin low-density separators (see Figure 1), while it is hard to decide which one is the best based on the limited labeled data. Though these low-density separators all coincide with the limited labeled data well, they are often diverse and therefore, a wrong selection may cause a large loss and result in a degenerated performance. Furthermore, the optimal objective value may deviate from the ground-truth because of the limited training data. Thus, selecting one optimal low-density separator according to the objective value may not be really optimal, and instead, we will try to consider all the candidate low-density separators.

Specifically, focusing on transductive setting, we construct S4VMs by optimizing the label assignment for unlabeled instances in the worse case. Theoretical analysis discloses that if the ground-truth label assignment can be realized by a low-density separator, as assumed by current S3VMs, our S4VMs will never degenerate performance. We present two implementations of S4VMs; one tries to find diverse large-margin low-density separators based on global simulated annealing search, while the other is based on a sim-

Appearing in *Proceedings of the 28^{th} International Conference on Machine Learning*, Bellevue, WA, USA, 2011.

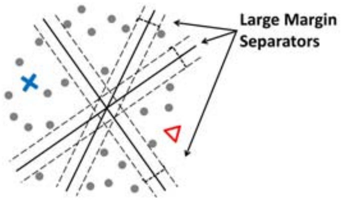


Figure 1. There are usually multiple large-margin low-density separators coincide well with labeled data (cross and triangle)

pler and efficient sampling strategy. Comprehensive experiments show that the overall performance of S4VMs are highly competitive with S3VMs, while contrasting to S3VMs which degenerate performances in many cases, our S4VMs are never significantly worse than inductive SVMs (i.e., SVMs considering only labeled data).

The rest of this paper is organized as follows. We briefly introduce S3VMs in Section 2, and then present our S4VMs in Section 3. Experimental results are reported in Section 4, and finally we conclude the paper in Section 5.

2. S3VMs

Inspired by the success of large margin principle, S3VMs are extensions of supervised SVMs to semi-supervised learning by simultaneously learning the optimal hyperplane and the labels for unlabeled instances. It was disclosed that S3VMs realize the low-density assumption (Chapelle & Zien, 2005) by favoring the decision boundary going across low-density regions.

Formally, considering binary classification, we are given a set of labeled data $\{\mathbf{x}_i, y_i\}_{i=1}^{l}$ and a set of unlabeled data $\{\hat{\mathbf{x}}_j\}_{j=1}^{u}$, where $\mathbf{x}, \hat{\mathbf{x}} \in \mathcal{X}$, $y \in \{\pm 1\}$, l and u are the number of labeled and unlabeled instances, respectively. The goal is to find a function $f : \mathcal{X} \to \{\pm 1\}$ and $\hat{\mathbf{y}} \in \{\pm 1\}^u$ such that the following functional is minimized:

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

where $\mathcal{B} = \{\hat{\mathbf{y}} \in \{\pm 1\}^u | -\beta \leq \frac{\sum_{j=1}^{u} \hat{y}_j}{u} - \frac{\sum_{i=1}^{l} y_i}{l} \leq \beta\}$ is induced by the balance constraint to avoid trivial solution (Joachims, 1999; Chapelle et al., 2008), $\mathcal{H}$ is the Reducing Kernel Hilbert Space (RKHS) induced by a kernel k and $\ell(y, f(\mathbf{x})) = \max\{0, yf(\mathbf{x}) - 1\}$ is the hinge loss, C_1 and C_2 are regularization parameters trading off the complexity and the empirical error on label and unlabeled data, respectively. As can be seen from Eq. 1, S3VMs enforce the decision boundary to lie in low-density regions, and otherwise a large loss will occur with respective to the objective function (Chapelle & Zien, 2005).

Unlike inductive SVMs with convex formulation, the formulation of S3VMs (i.e., Eq. 1) is non-convex and the optimal solution is intractable in general. Great efforts have been devoted to avoiding S3VMs getting stuck in poor local minima. Roughly, there are three categories of approaches. The first kind of approaches are based on global combinatorial optimization (e.g., branch-and-bound search), and achieve promising performances on small data sets (Bennett & Demiriz, 1999; Chapelle et al., 2007). The second kind of approaches are based on global heuristic search, which gradually increases the difficulty of solving the non-convex part in Eq. 1. Examples include the TSVM (Joachims, 1999) which gradually increases the value of C_2, the deterministic annealing approach (Sindhwani et al., 2006) which gradually increases the temperature of an entropy function, the continuation method (Chapelle et al., 2006a) which gradually decreases the smoothing of a surrogate function, etc. The third kind of approaches are based on convex relaxation, which transforms Eq. 1 into a relaxed convex problem. Examples include the semi-definite programming (SDP) relaxation (De Bie & Cristianini, 2004; Xu & Schuurmans, 2005), the minimax relaxation (Li et al., 2009b;a), etc.

Avoiding inappropriate local minima when approaching the optimal solution of Eq. 1 can be regarded as a strategy towards safe S3VMs; however, this is quite challenging. To the best of our knowledge, there was no proposal of safe S3VMs in literature.

3. S4VMs

As mentioned, given limited labeled data and abundant unlabeled data, there usually exist multiple large-margin low-density separators coincide well with the labeled data. Without further prior information for distinguishing these separators, it might be risky to select any one of them. So, we suggest to consider all these candidate separators.

In the following, we first introduce how to construct S4VMs given a number of diverse large-margin separators, by optimizing the label assignment for unlabeled instances such that the worst-case performance improvement over inductive SVM is maximized; then, we present two S4VM implementations which search for diverse large-margin separators by a global simulated annealing search and an efficient sampling strategy, respectively.

3.1. Constructing S4VMs

Given the predictors of multiple low-density separators $\{\hat{\mathbf{y}}_t\}_{t=1}^{T}$, suppose that $\mathbf{y}^*$ is the ground-truth label assignment and let $\mathbf{y}^{svm}$ denote the predictions of the inductive SVM on unlabeled data. For any label assignment $\mathbf{y} \in \{\pm 1\}^u$, denote $earn(\mathbf{y}, \mathbf{y}^*, \mathbf{y}^{svm})$ and $lose(\mathbf{y}, \mathbf{y}^*, \mathbf{y}^{svm})$ as the increased and decreased accuracies compared to the inductive SVM, respectively. Our goal is to learn $\mathbf{y}$ such that the improved performance over the inductive SVM is maximized; this can be cast as the optimization problem:

$$\max_{\mathbf{y}\in\{\pm1\}^u} earn(\mathbf{y},\mathbf{y}^*,\mathbf{y}^{svm}) - \lambda\, lose(\mathbf{y},\mathbf{y}^*,\mathbf{y}^{svm}), \quad (2)$$

where λ is a parameter for trading-off how much risk the user would like to undertake. For simplification of notation, we denote $J(\mathbf{y},\hat{\mathbf{y}},\mathbf{y}^{svm})$ as $earn(\mathbf{y},\hat{\mathbf{y}},\mathbf{y}^{svm}) - \lambda\, lose(\mathbf{y},\hat{\mathbf{y}},\mathbf{y}^{svm})$ in the sequel.

Note that the difficulty for solving Eq. 2 lies in the fact that the ground-truth $\mathbf{y}^*$ is unknown; otherwise it is trivial to get the solution. Similar to existing S3VMs, here we assume that the ground-truth boundary $\mathbf{y}^*$ can be realized by a low-density separator in $\{\hat{\mathbf{y}}_t\}_{t=1}^T$, i.e., $\mathbf{y}^* \in \mathcal{M} = \{\hat{\mathbf{y}}_t\}_{t=1}^T$. We consider optimizing the worst-case improvement over the inductive SVM, that is,

$$\bar{\mathbf{y}} = \arg\max_{\mathbf{y}\in\{\pm1\}^u} \min_{\hat{\mathbf{y}}\in\mathcal{M}} J(\mathbf{y},\hat{\mathbf{y}},\mathbf{y}^{svm}). \quad (3)$$

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. $\bar{\mathbf{y}}$ is the optimal solution and $\forall\hat{\mathbf{y}}$, $J(\mathbf{y}^{svm},\hat{\mathbf{y}},\mathbf{y}^{svm})$ is always zero, and thus, we have

$$\min_{\hat{\mathbf{y}}\in\mathcal{M}} J(\bar{\mathbf{y}},\hat{\mathbf{y}},\mathbf{y}^{svm}) \geq \min_{\hat{\mathbf{y}}\in\mathcal{M}} J(\mathbf{y}^{svm},\hat{\mathbf{y}},\mathbf{y}^{svm}) = 0. \quad (4)$$

Since $\mathbf{y}^* \in \mathcal{M}$, we have

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

According to Eqs. 4 and 5, we have $J(\bar{\mathbf{y}},\mathbf{y}^*,\mathbf{y}^{svm}) \geq 0$, i.e., $earn(\bar{\mathbf{y}},\mathbf{y}^*,\mathbf{y}^{svm}) \geq \lambda\, lose(\bar{\mathbf{y}},\mathbf{y}^*,\mathbf{y}^{svm})$. Recall that $\lambda \geq 1$, we have $earn(\bar{\mathbf{y}},\mathbf{y}^*,\mathbf{y}^{svm}) \geq lose(\bar{\mathbf{y}},\mathbf{y}^*,\mathbf{y}^{svm})$, and the theorem is proved. □

Theorem 1 shows that the S4VM is never worse than the inductive SVM. 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 $\mathbf{y}^{svm}$.*

To solve Eq. 3, note that the following are linear functions of $\mathbf{y}$:

$$\begin{aligned} earn(\mathbf{y},\mathbf{y}^*,\mathbf{y}^{svm}) &= \sum\nolimits_{j=1}^{u} I(y_j = y_j^*) I(y_j^* \neq y_j^{svm}) \\ &= \sum\nolimits_{j=1}^{u} \frac{1+y_j y_j^*}{2} \frac{1-y_j^{svm} y_j^*}{2}, \\ lose(\mathbf{y},\mathbf{y}^*,\mathbf{y}^{svm}) &= \sum\nolimits_{j=1}^{u} I(y_j \neq y_j^*) I(y_j^* = y_j^{svm}) \\ &= \sum\nolimits_{j=1}^{u} \frac{1-y_j y_j^*}{2} \frac{1+y_j^{svm} y_j^*}{2}. \end{aligned}$$

Without lose of generality, let $J(\mathbf{y},\hat{\mathbf{y}}_t,\mathbf{y}^{svm}) = \mathbf{c}_t'\mathbf{y} + d_t$. Eq. 3 can be cast as

$$\max_{\mathbf{y}\in\{\pm1\}^u} \theta \quad \text{s.t.} \quad \theta \leq \mathbf{c}_t'\mathbf{y} + d_t, \forall t = 1,\ldots,T. \quad (6)$$

Though Eq. 6 is an integer linear programming, according to Proposition 1, we do not need to get the optimal solution for achieving our goal, and thus we employ a simple heuristic method to solve Eq. 6. Specifically, we first solve a convex linear programming by relaxing the integer constraint of $\mathbf{y}$ in Eq. 6 to $[-1,1]^u$ and then project it back to integer solution with minimum distance. If the function value of the resulting integer solution is smaller than that of $\mathbf{y}^{svm}$, $\mathbf{y}^{svm}$ is output as the final solution instead. It is evident that our final solution satisfies Proposition 1.

Note that prior knowledge on low-density separators can be easily incorporated into our framework. Specifically, by introducing dual variables $\boldsymbol{\alpha}$ for constraints in Eq. 6, according to KKT condition, Eq. 6 can be reformulated as

$$\max_{\mathbf{y}\in\{\pm1\}^u} \min_{\boldsymbol{\alpha}'\mathbf{1}=1,\boldsymbol{\alpha}\geq0} \sum\nolimits_{t=1}^{T} \alpha_t(\mathbf{c}_t'\mathbf{y} + d_t). \quad (7)$$

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 learn the optimal $\mathbf{y}$ with respect to the target in Eq. 7 using the known $\boldsymbol{\alpha}$.

It is worth mentioning that, by considering all candidate large-margin low-density separators, S4VMs are relevant to ensemble methods (Zhou, 2009), and the spirit may also be extended to other semi-supervised learning approaches.

3.2. Two Implementations

Now we consider how to find diverse large-margin low-density separators. Let $h(f,\hat{\mathbf{y}})$ denote the functional to be minimized by the objective function of S3VMs (i.e., Eq. 1):

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

Our goal is to find multiple large-margin low-density separators $\{f_t\}_{t=1}^T$ and the corresponding label assignments $\{\hat{\mathbf{y}}_t\}_{t=1}^T$ such that the following functional is minimized:

$$\min_{\{f_t,\hat{\mathbf{y}}_t\in\mathcal{B}\}_{t=1}^T} \sum\nolimits_{t=1}^{T} h(f_t,\hat{\mathbf{y}}_t) + M\Omega(\{\hat{\mathbf{y}}_t\}_{t=1}^T), \quad (8)$$

where T is the number of separators, Ω is a quantity of penalty about the diversity of separators, and M is a large constant (e.g., 10^5 in our experiments) enforcing large diversity. It is evident that minimizing Eq. 8 favors the separators with large-margin as well as large diversity.

In this paper, we consider $\Omega(\{\hat{\mathbf{y}}_t\}_{t=1}^T)$ as sum of pairwise terms, i.e., $\Omega(\{\hat{\mathbf{y}}_t\}_{t=1}^T) = \sum_{1\leq t\neq\tilde{t}\leq T} \mathbf{I}(\frac{\hat{\mathbf{y}}_t'\hat{\mathbf{y}}_{\tilde{t}}}{u} \geq 1-\epsilon)$

where $\mathbf{I}$ is the identity function and $\epsilon \in [0, 1]$ is a constant, but note that other penalty quantities are also applicable.

Without loss of generality, suppose that f is a linear model, i.e., $f(\mathbf{x}) = \mathbf{w}'\phi(\mathbf{x}) + b$ where $\phi(\mathbf{x})$ is a feature mapping induced by the kernel k. Thus, Eq. 8 is cast as:

$$\begin{aligned}\min_{\{\mathbf{w}_t, b_t, \hat{\mathbf{y}}_t \in \mathcal{B}\}_{t=1}^T} \quad & \sum_{t=1}^{T} \Big(\frac{1}{2}\|\mathbf{w}_t\|^2 + C_1 \sum_{i=1}^{l} \xi_i + C_2 \sum_{j=1}^{u} \hat{\xi}_j \Big) \\ & + M \sum\nolimits_{1 \le t \ne \tilde{t} \le T} I\Big(\frac{\hat{\mathbf{y}}_t' \hat{\mathbf{y}}_{\tilde{t}}}{u} \ge 1 - \epsilon\Big) \qquad (9) \\ \text{s.t.} \quad & y_i \mathbf{w}_t' \phi(\mathbf{x}_i) \ge 1 - \xi_i, \quad \forall i = 1, \ldots, l, \\ & \hat{y}_{t,j} \mathbf{w}_t' \phi(\hat{\mathbf{x}}_j) \ge 1 - \hat{\xi}_j, \ \forall j = 1, \ldots, u, \\ & \qquad\qquad\qquad\qquad \forall t = 1, \ldots, T, \end{aligned}$$

where $\hat{y}_{t,j}$ refers to the jth entry of $\hat{\mathbf{y}}_t$. Eq. 9 is non-convex and in the following we will present two solutions. It is evident that this can also be implemented by other solutions, especially those based on efficient S3VMs.

3.2.1. Global Simulated Annealing Search

Our first implementation is based on global search, e.g., simulated annealing (SA) search (Kirkpatrick, 1984; Černŷ, 1985). SA is a probabilistic method for approaching global solutions of objective functions suffering from multiple local minima. Specifically, at each step, SA replaces current solution by a random nearby solution with a probability depending on the value difference between their corresponding function targets as well as a global parameter, i.e., the temperature P, which gradually decreases during the process. When P is large, current solution almost changes randomly; while as P goes to zero, the changes are increasingly "downhill". In theory, according to the convergence analysis of Markov Process, the prob ability that SA converges to the global solution approaches to 1 as SA procedure is extended (Laarhoven & Aarts, 1987).

To alleviate the low convergence rate of pure SA, inspired by (Sindhwani et al., 2006), a deterministic local search scheme is used. Specifically, once $\{\hat{\mathbf{y}}_t\}_{t=1}^T$ are fixed, $\{\mathbf{w}_t, b_t\}_{t=1}^T$ are solved via multiple individual SVM subroutines; once $\{\mathbf{w}_t, b_t\}_{t=1}^T$ are fixed, $\{\hat{\mathbf{y}}_t\}_{t=1}^T$ are updated based on local binary search, iteratively until convergence.

Algorithm 1 presents the pseudo-code of simulated annealing approach for Eq. 9, where the local search subroutine is given in Algorithm 2.

3.2.2. Representative Sampling

To further alleviate the computational complexity, our second implementation is based on heuristic sampling search. Recall that the goal of Eq. 8 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, by searching for multiple large-margin low-density separators at first and then selecting the representative separators.

Algorithm 1 Solving Eq. 9 by Simulated Annealing Search

```
Input: $\{\mathbf{x}_i, y_i\}_{i=1}^l$, $\{\hat{\mathbf{x}}_j\}_{j=1}^u$, $T$
Output: $\{\hat{\mathbf{y}}_t^{best}\}_{t=1}^T$
Initialize $P \leftarrow 1$, $e \leftarrow 1$, $\{\hat{\mathbf{y}}_t\}_{t=1}^T$
$(\{\hat{\mathbf{y}}_t\}_{t=1}^T, o) \leftarrow$ 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
        $o \leftarrow o^{new}$;  $\hat{\mathbf{y}}_t^{best} \leftarrow \hat{\mathbf{y}}_t \leftarrow \hat{\mathbf{y}}_t^{new}, \forall t = 1, \ldots, T$
    else if random() $<$ exp$(-(o^{new} - o)/P)$ 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
        $P \leftarrow$ cooling($P$); $e \leftarrow 1$
    end if
end while
```

Algorithm 2 Localsearch

```
Input: $\{\hat{\mathbf{y}}_t\}_{t=1}^T$;  (Denote $[m] = \{1, \ldots, m\}$)
Output: $(\{\hat{\mathbf{y}}_t\}_{t=1}^T, obj)$
while not converged do
    Fix $\{\hat{\mathbf{y}}_t\}_{t=1}^T$, solve $\{\mathbf{w}_t, b_t\}_{t=1}^T$ via multiple SVMs
    while not converged do
        cyclically random pick $j \in [u]$, $t \in [T]$
        optimize $\hat{y}_{t,j} \in \{\pm 1\}$ according to Eq. 9
    end while
end while
Output $\{\hat{\mathbf{y}}_t\}_{t=1}^T$ and corresponding objective value $obj$
```

Algorithm 3 shows the pseudo-code of our second implementation. As can be seen, multiple candidate large-margin low-density separators are first obtained via local search similar to that of Algorithm 2. A clustering algorithm is then applied to identify the representative separators. This approach is simple, and experiments in Section 4 show that it is efficient and effective.

4. Experiments

We evaluate S4VMs on a broad range of tasks including seven SSL benchmark data sets[1], i.e., *digit1, USPS, BCI, g241c, g241n, COIL, Text*, and sixteen UCI data sets[2]. Table 1 summarizes the statistics of the data sets.

Both linear and RBF kernels are used in our experiments. As for benchmark data sets, the archival includes two sets of twelve data splits, one with 10 while the other with 100

[1] http://www.kyb.tuebingen.mpg.de/ssl-book/

[2] http://archive.ics.uci.edu/ml/datasets.html

Algorithm 3 Solving Eq. 9 by Representative Sampling

```
Input: $\{\mathbf{x}_i, y_i\}_{i=1}^{l}$, $\{\hat{\mathbf{x}}_j\}_{j=1}^{u}$, $T$;
Output: $\{\hat{\mathbf{y}}_t^{best}\}_{t=1}^{T}$
Randomly sampling $N$ number of $\hat{\mathbf{y}}$'s, i.e., $\mathcal{S} = \{\hat{\mathbf{y}}_n\}_{n=1}^{N}$
for $n = 1 : N$ do
    while not converged do
        Fix $\hat{\mathbf{y}}_n$, solve $\{\mathbf{w}_n, b_n\}$ via SVM solver
        Fix $\{\mathbf{w}_n, b_n\}$, update $\hat{\mathbf{y}}_n$ w.r.t S3VM's objective function via sorting (Zhang et al., 2007)
    end while
end for
Perform clustering (e.g., $k$-means) for $\mathcal{S}$ where $k = T$
Output $\hat{\mathbf{y}}$'s with the minimum objective value within each cluster
```

Table 1. Experimental data sets.

ID	Data	# Inst	# Feat	ID	Data	# Inst	# Feat
1	BCI	400	117	13	vehicle	435	16
2	g241c	1500	241	14	house-votes	435	16
3	g241d	1500	241	15	clean1	476	166
4	COIL	1500	241	16	wdbc	569	14
5	Digit1	1500	241	17	isolet	600	51
6	USPS	1500	241	18	breastw	683	9
7	Text	1500	11960	19	austra	690	42
8	house	232	16	20	australian	690	15
9	heart	270	9	21	diabetes	768	8
10	haberman	306	14	22	german	1000	59
11	liverDisorders	345	6	23	optdigits	1143	42
12	ionosphere	351	33				

labeled examples. As for UCI data sets, we randomly select 10 and 100 examples to be used as labeled examples, and use the remaining data as unlabeled data. The experiments are repeated for 30 times and the average accuracy and standard deviations are recorded.

Inductive SVM[3] and TSVM[4] (Joachims, 1999) are evaluated as baselines. Linear programming is conducted by linprog function in MATLAB. The regularization parameters C_1, C_2 and β in balance constraint are fixed as 100, 0.1 and 0.1 for all S3VMs. We call our S4VM which uses simulated annealing as S4VM$_a$, and the one which uses sampling as S4VM$_s$. For S4VM$_a$, ϵ and T are simply fixed as 0.05 and 3, respectively. For S4VM$_s$, the sampling size N and the number of separators T are simply fixed as 100 and 10, respectively. λ is fixed as 3 for S4VMs. For 10 labeled examples, the width of RBF kernel is set as δ, i.e., the average distance between instances; for 100 labeled examples, the width of RBF kernel is selected by 5-fold cross validation from the set of $\{0.25\delta, 0.5\delta, \delta, 2\delta, 4\delta\}$.

4.1. Results of S4VM$_a$

In addition to inductive SVM and TSVM, we also compare S4VM$_a$ with three variants using multiple low-density separators. S3VM$_a^{best}$ presents the best performance among the multiple candidate separators (note that this method is impractical), S3VM$_a^{min}$ selects the low-density separator with minimum objective value, and S3VM$_a^{com}$ combines the candidate separators using uniform weights. Though simulated annealing was used to improve the efficiency of S3VMs (Sindhwani et al., 2006), note that it is still with high computational load, and therefore Table 2 only reports the performances on UCI data sets with RBF kernels.

Table 2 shows that the overall performance of S4VM$_a$ is highly competitive with TSVM. In terms of pairwise accuracy comparison, S4VM$_a$ is better than TSVM on 6 out of 12 data sets for 10 labeled examples, while this number rises to 11 out of 12 data sets for 100 labeled examples. In terms of average accuracy, S4VM$_a$ is slightly worse (better) than TSVM for 10 (100) labeled examples. S3VM$_a^{min}$ and S3VM$_a^{com}$ do not perform as well as S4VM$_a$.

More importantly, unlike TSVM which is significantly worse than inductive SVM on 4 out of 12 data sets for 10 labeled data, and 7 out of 12 for 100 labeled data, S4VM$_a$ never degenerates the performance significantly. Both S3VM$_a^{min}$ and S3VM$_a^{com}$ are capable to reduce the chance of significantly degenerating performance compared with TSVM, however, they still degenerate performance significantly in many cases.

Though the condition of Theorem 1 is relaxed than traditional assumption of S3VMs, the theorem does not always hold owing to many factors, e.g., the ground-truth is not among the low-density separators. Even in such cases, however, S4VM may still work. Note that Theorem 1 presents a sufficient rather than necessary condition for S4VMs, and the relevance to ensemble methods provides an explanation to S4VMs' superiority to single separators.

4.2. Results of S4VM$_s$

Similar to S4VM$_a$, three variants, i.e., S3VM$_s^{best}$, S3VM$_s^{min}$ and S3VM$_s^{com}$ are compared with S4VM$_s$ in Table 3, in addition to inductive SVM and TSVM.

Table 3 shows that the overall performance of S4VM$_s$ is highly competitive with TSVM, though the ground-truth is seldom realized by a low-density separator (see the performance of S3VM$_s^{best}$). In terms of pairwise accuracy comparison, S4VM$_s$ outperforms TSVM on 15/13 and 13/18 out of the 23 data sets with linear/RBF kernels for 10 and 100 labeled examples, respectively. In terms of average accuracy, S4VM$_s$ is slightly worse (better) than TSVM for 10 (100) labeled examples. Except for the case of S3VM$_s^{min}$ on 100 label examples, S3VM$_s^{min}$ and S3VM$_s^{com}$ do not perform as well as S4VM$_s$.

More importantly, unlike TSVM which degenerates performance on 12 and 17 cases for 10 and 100 labeled ex-

[3] http://www.csie.ntu.edu.tw/ cjlin/libsvm/

[4] http://svmlight.joachims.org/

Table 2. Accuracy (mean±std.) of S4VM$_a$ and compared methods. 'SVM' denotes inductive SVM. For semi-supervised methods (TSVM, S3VM$_a^{best}$, S3VM$_a^{com}$ and S4VM$_a$), if the performance is significantly better/worse than SVM (paired t-tests at 95% significance level), the corresponding entries are bolded/underlined. The win/tie/loss counts are summarized in the last row, and the method with the smallest number of losses against SVM is bolded.

# Labeled	Data	SVM	TSVM	S3VM$_a^{best}$	S3VM$_a^{min}$	S3VM$_a^{com}$	S4VM$_a$
10	austra	66.6±7.5	67.8±13.2	69.4±12.2	67.4±12.5	67.2±12.5	68.0±10.4
	breastw	95.7±2.8	94.7±0.1	95.9±0.8	95.9±0.8	**96.6±0.6**	96.5±0.6
	diabetes	65.6±5.5	65.3±6.2	65.3±6.0	64.8±6.3	65.2±6.1	65.4±5.8
	haberman	65.9±8.1	62.4±6.2	64.3±6.0	63.3±6.5	65.1±7.5	65.8±7.6
	heart	71.8±6.8	72.5±7.6	73.1±6.9	71.9±7.5	72.3±7.3	72.2±7.0
	house	88.0±2.8	**89.8±1.8**	**89.9±2.1**	88.0±1.9	**89.1±2.0**	**88.5±2.4**
	house-votes	87.0±3.1	85.0±5.8	84.6±8.4	83.8±8.1	84.3±8.5	85.4±6.0
	ionosphere	74.7±9.1	77.4±8.6	75.3±9.7	73.7±10.0	74.0±9.1	74.7±9.6
	isolet	81.0±14.7	**86.7±9.9**	82.0±19.8	81.4±19.8	82.4±21.2	**83.3±18.4**
	liverDisorders	55.5±5.9	54.2±4.8	**56.4±5.1**	55.7±5.1	55.4±5.4	55.6±5.9
	vehicle	74.3±8.1	**78.7±8.7**	76.8±10.9	75.3±10.9	75.2±10.9	75.4±9.8
	wdbc	80.7±7.5	**84.4±6.3**	80.9±9.6	80.2±9.8	81.4±9.3	**81.9±8.6**
Average Accuracy		75.6	76.6	76.2	75.1	75.7	76.1
SVM vs. Semi-Supervised: W/T/L			4/4/4	1/9/2	2/10/0	0/10/2	**0/9/3**

# Labeled	Data	SVM	TSVM	S3VM$_a^{best}$	S3VM$_a^{min}$	S3VM$_a^{com}$	S4VM$_a$
100	austra	78.7±2.9	78.6±2.8	78.5±2.8	78.4±2.8	78.7±2.9	78.8±2.9
	breastw	95.4±1.0	**95.8±0.7**	95.2±1.0	95.2±1.0	95.3±1.0	95.4±1.0
	diabetes	70.3±2.1	70.0±2.1	70.0±2.0	69.8±1.9	70.1±2.0	70.3±2.1
	haberman	68.3±2.8	66.3±2.6	68.2±2.6	67.2±3.0	68.0±2.6	68.3±2.8
	heart	76.3±3.4	76.0±3.4	76.5±3.0	76.2±3.3	76.0±3.1	76.4±3.5
	house	94.9±1.7	92.4±3.3	93.9±2.7	93.4±2.8	94.5±2.0	**95.1±1.6**
	house-votes	92.5±1.7	90.9±2.4	91.9±1.8	91.6±2.0	92.3±1.6	92.5±1.7
	ionosphere	91.5±2.1	90.6±2.8	90.8±2.1	90.7±2.2	91.5±2.1	91.5±2.2
	isolet	99.2±0.5	96.4±3.4	98.2±1.6	98.2±1.7	98.8±0.7	99.2±0.5
	liverDisorders	66.5±2.6	66.1±2.3	66.9±2.5	66.4±2.7	66.6±2.6	66.6±2.6
	vehicle	97.7±1.0	96.0±2.1	97.6±1.0	97.4±1.3	97.6±0.8	97.6±1.0
	wdbc	93.6±1.7	92.4±2.6	92.8±2.2	92.7±2.3	93.2±1.8	93.5±1.8
Average Accuracy		85.4	84.3	85.1	84.8	85.2	85.4
SVM vs. Semi-Supervised: W/T/L			7/4/1	6/6/0	8/4/0	4/8/0	**0/11/1**

amples, respectively, S4VM$_s$ never degenerates the performance significantly. Both S3VM$_s^{min}$ and S3VM$_s^{com}$ are capable to reduce the chance of degenerating performance compared with TSVM, however, they still degenerate performance significantly in many cases.

Wilcoxon sign tests at 95% significant level disclose that S4VM$_s$ is significantly better than inductive SVM for both 10 and 100 labeled examples. The other three semi-supervised methods, however, do not obtain such a significance. These results validate the effectiveness of S4VM$_s$.

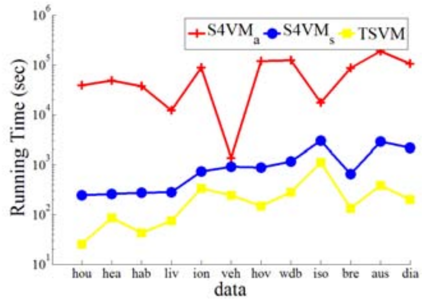


Figure 2. Running time (in seconds) of S4VM$_a$, S4VM$_s$ and TSVM with 10 labeled examples

4.3. Running Time

Figure 2 plots the running time on 12 UCI data sets with 10 labeled examples. As can be seen, S4VM$_a$ has the highest time cost, and S4VM$_s$ scales slightly worse than TSVM but much better than S4VM$_a$. Note that S4VM$_s$ is inherently parallel due to the consideration of multiple separators, and it can be speedup by parallel implementation or using efficient S3VM solutions.

4.4. Parameter Influence

S4VM$_s$ has four parameters, i.e., sampling size N, cluster number T, risk parameter λ and the kernel type. Figure 3 studies the influence of N, T and λ with linear/RBF kernels on five representative data sets with 10 labeled examples.

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 label assignments in the worse case. This property makes S4VM$_s$ even more attractive, since the performance of current S3VMs are usually sensitive to parameter settings, especially when the number of labeled examples is too few to afford a reliable model selection. Moreover, paired t-tests at 95% significant level

Table 3. Accuracy (mean ± std.) of S4VM$_s$ and compared methods (see title of Table 2 for more information).

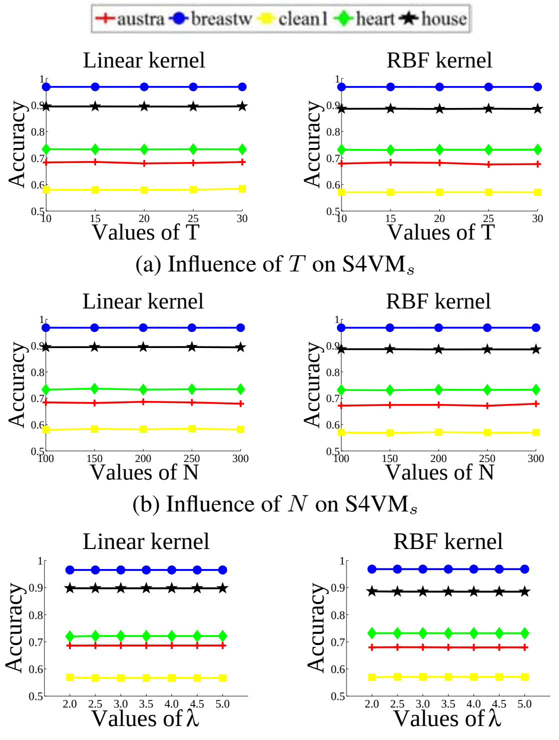
10 Lab	SVM	TSVM	S3VM$_s^{best}$	S3VM$_s^{min}$	S3VM$_s^{com}$	S4VM$_s$
	(linear / rbf)	(linear / rbf)	(linear / rbf)	(linear / rbf)	(linear / rbf)	(linear / rbf)
aust	67.2±8.0 / 66.6±7.5	67.8±12.1 / 67.8±13.2	**69.4±10.0** / **73.1±11.8**	69.0±10.0 / 68.2±12.6	69.0±10.0 / 67.4±11.4	68.7±9.8 / **68.1±9.8**
ausl	77.3±7.1 / 72.9±6.1	**81.1±6.1** / **77.0±8.2**	**82.2±5.2** / **82.3±4.0**	79.1±8.2 / 73.3±10.6	74.3±8.9 / **75.1±7.9**	77.8±7.5 / 73.3±6.4
brea	94.4±3.5 / 95.7±2.8	94.5±0.4 / 94.7±0.1	**96.9±0.3** / **96.8±0.2**	**96.8±0.4** / **96.8±0.2**	**96.5±0.7** / **96.7±0.3**	**96.5±0.6** / **96.8±0.4**
clea	56.6±5.0 / 57.1±8.1	56.7±5.7 / 55.8±7.8	**63.2±4.7** / **61.9±5.4**	55.4±6.3 / 54.7±7.1	54.0±5.1 / 54.9±8.0	56.7±5.1 / 56.9±8.2
diab	65.7±5.4 / 65.6±5.5	65.7±5.4 / 65.3±6.2	66.7±5.2 / 65.8±6.2	66.4±5.3 / 65.2±6.6	65.6±5.6 / 65.2±6.3	66.4±5.2 / 65.5±6.2
germ	63.1±8.0 / 64.8±12.1	62.2±5.5 / 61.6±5.1	**66.0±4.5** / 65.9±4.9	61.0±6.7 / 61.6±6.2	60.7±9.0 / 61.1±7.7	63.3±7.9 / 64.8±11.8
habe	64.6±7.0 / 65.9±8.1	62.1±7.9 / 62.4±6.2	65.2±6.1 / 65.7±5.1	64.1±6.6 / 63.4±6.7	63.7±6.5 / 63.9±7.8	65.0±6.5 / 65.9±7.9
hear	72.2±6.7 / 71.8±6.8	71.9±6.9 / 72.5±7.6	72.2±6.8 / **74.4±6.2**	71.9±6.9 / 72.8±7.1	71.7±6.8 / 72.9±6.8	72.2±6.7 / 72.7±7.0
hous	89.0±3.0 / 88.0±2.8	**90.6±3.3** / **89.8±1.8**	**93.5±4.0** / **90.1±2.1**	90.1±4.6 / 88.4±2.1	88.5±4.3 / 88.5±2.3	**89.6±3.1** / 88.5±2.3
houv	88.0±2.1 / 87.0±3.1	85.4±6.4 / 85.0±5.8	88.9±5.6 / 87.0±4.6	87.3±6.4 / 86.1±4.7	86.5±5.2 / 86.2±4.5	87.9±4.4 / 86.8±3.9
iono	72.6±6.8 / 74.7±9.1	72.1±9.4 / 77.4±8.6	**78.2±5.8** / **80.8±7.4**	74.6±6.8 / 74.3±13.2	**74.4±6.0** / 76.1±8.7	73.7±6.7 / 75.2±10.0
isol	89.6±6.8 / 81.0±14.7	86.6±10.0 / **86.7±9.9**	90.0±10.3 / **89.1±12.1**	88.6±12.6 / **87.3±13.4**	87.9±13.5 / **87.2±13.2**	**91.4±9.0** / **85.2±17.2**
live	55.3±5.4 / 55.5±5.9	53.3±5.0 / 54.2±4.8	55.2±6.0 / 55.6±5.8	54.9±6.1 / 55.4±6.0	54.2±5.5 / 55.2±5.8	54.8±5.7 / 55.2±6.0
optd	92.2±4.4 / 85.8±9.8	87.1±8.7 / 87.2±8.7	91.1±7.4 / **90.9±7.9**	89.7±9.8 / **88.9±10.6**	88.2±9.7 / 87.7±10.1	**93.5±5.6** / **87.9±11.3**
vehi	74.0±7.9 / 74.3±8.1	76.2±9.2 / **78.7±8.7**	**76.1±8.8** / **80.2±10.8**	74.6±8.6 / 76.8±10.0	73.8±8.4 / 75.9±9.1	74.7±8.5 / 76.2±9.8
wdbc	81.3±7.4 / 80.7±7.5	**84.3±6.6** / **84.4±6.3**	82.1±7.1 / **84.2±7.5**	82.0±7.1 / **83.5±7.5**	81.9±7.1 / **83.4±7.4**	82.1±7.2 / **82.5±9.1**
digi	76.2±7.1 / 57.2±12.1	**80.7±3.9** / **84.2±4.4**	77.4±4.6 / **82.5±6.2**	74.3±6.4 / **79.4±6.7**	61.4±8.9 / **75.8±9.4**	76.0±7.1 / **63.6±12.6**
USPS	78.5±2.9 / 80.0±0.1	72.2±3.7 / 71.6±4.9	81.6±4.4 / 77.7±3.1	72.0±5.1 / 64.6±4.6	77.6±2.9 / 75.9±4.5	78.7±2.7 / **80.1±0.2**
COIL	56.5±5.4 / 56.9±4.2	54.9±6.2 / 57.2±3.9	**65.6±5.1** / **67.5±5.9**	56.8±6.4 / 59.6±6.5	54.7±6.2 / 56.9±7.1	56.8±5.4 / 57.0±4.2
BCI	52.6±2.4 / 51.3±2.0	50.9±2.8 / 51.2±2.1	53.9±2.9 / **53.8±1.8**	52.0±2.3 / 51.3±2.5	50.4±3.0 / 50.1±2.9	51.8±2.3 / 51.3±2.2
g241c	54.5±4.2 / 52.3±4.4	**78.3±4.5** / **59.5±3.4**	**60.6±3.2** / **59.0±3.0**	56.5±6.8 / **56.4±4.1**	49.3±2.7 / 49.8±3.1	54.6±4.5 / **52.8±4.4**
g241n	56.4±5.2 / 52.5±5.3	53.3±7.3 / 52.8±5.2	**60.3±3.1** / 56.3±2.9	51.4±3.0 / 52.4±5.9	48.4±4.6 / 49.5±5.6	56.3±5.1 / 52.7±5.1
Text	52.2±3.0 / 52.6±4.0	**64.8±7.9** / 52.4±6.1	**56.9±2.4** / **55.3±1.7**	**55.6±3.9** / 51.5±4.3	43.0±4.5 / 46.4±6.4	52.1±2.9 / 52.6±4.0
Avg.Acc.	71.3 / 69.7	72.2 / 71.3	74.0 / 74.1	71.1 / 70.6	69.1 / 70.2	71.8 / 70.6
SVM vs. Semi-Supervised: W/T/L		12/19/13	1/15/28	6/30/8	15/22/7	**0/32/12**

100 Lab	SVM	TSVM	S3VM$_s^{best}$	S3VM$_s^{min}$	S3VM$_s^{com}$	S4VM$_s$
	(linear / rbf)	(linear / rbf)	(linear / rbf)	(linear / rbf)	(linear / rbf)	(linear / rbf)
aust	83.7±1.8 / 78.7±2.9	83.1±2.1 / 78.6±2.8	83.7±1.6 / 78.9±2.8	83.5±1.6 / 78.7±2.9	82.6±2.5 / 78.6±3.0	83.7±1.6 / 78.8±3.0
ausl	79.5±2.8 / 80.6±2.3	79.0±2.9 / 81.1±2.5	**81.5±2.1** / **81.8±1.9**	**80.9±2.2** / **81.2±2.0**	**80.7±2.2** / **81.2±2.1**	**81.1±2.1** / **81.3±2.0**
brea	95.0±1.3 / 95.4±1.0	**95.7±0.7** / **95.8±0.7**	**95.2±1.4** / **95.5±1.0**	95.1±1.3 / **95.5±1.0**	95.0±1.4 / 95.4±1.0	95.2±1.3 / 95.4±1.0
clea	73.3±3.1 / 83.1±2.0	73.6±2.9 / 83.2±2.2	**74.1±3.0** / **84.5±1.8**	**73.6±3.2** / **83.7±2.2**	73.6±3.0 / 83.5±2.1	73.5±2.9 / 83.3±2.1
diab	74.6±1.6 / 70.3±2.1	73.7±1.9 / 70.0±2.1	74.4±1.9 / 70.7±2.3	74.1±2.0 / 70.6±2.3	73.0±3.4 / 70.2±2.4	74.3±1.9 / 70.6±2.4
germ	65.3±3.0 / 70.9±1.0	**66.1±2.2** / 68.8±2.3	**66.2±3.0** / 70.4±1.7	65.8±2.9 / 69.4±1.8	65.5±3.5 / 68.8±2.0	**65.8±2.9** / 71.0±1.1
habe	71.7±3.1 / 68.3±2.8	69.1±3.0 / 66.3±2.6	71.6±3.6 / 67.7±2.6	69.2±4.2 / 67.5±2.8	69.3±4.1 / 67.2±2.9	71.9±3.2 / 68.1±2.6
hear	82.2±2.5 / 76.3±3.4	82.2±2.9 / 76.0±3.4	82.6±2.1 / 76.5±3.6	82.4±2.1 / 76.4±3.7	82.1±2.9 / 76.2±3.6	82.4±2.5 / 76.5±3.7
hous	94.0±1.8 / 94.9±1.7	92.1±3.2 / 92.4±3.3	**94.5±1.4** / **95.3±1.7**	94.2±1.8 / 94.9±1.9	94.0±1.5 / 94.6±1.7	94.2±1.6 / 94.9±1.7
houv	91.1±1.4 / 92.5±1.7	89.9±2.2 / 90.9±2.4	91.2±1.3 / 92.6±1.8	91.1±1.3 / 92.3±1.7	91.0±1.3 / 92.1±1.8	91.2±1.3 / 92.6±1.6
iono	83.9±2.7 / 91.5±2.1	81.8±3.0 / 90.6±2.8	83.6±2.6 / **91.9±2.0**	83.4±2.7 / 91.6±2.0	83.0±2.8 / 91.5±2.1	83.4±2.6 / 91.6±2.1
isol	99.0±0.4 / 99.2±0.5	96.3±3.3 / 96.4±3.4	98.9±1.4 / 99.0±1.5	98.9±1.4 / 98.9±1.4	98.9±1.3 / 98.8±1.3	98.9±1.2 / **99.3±0.5**
live	64.3±3.6 / 66.5±2.6	64.8±3.1 / 66.1±2.3	63.2±4.2 / 67.0±3.4	62.9±4.2 / 66.9±3.4	62.2±4.7 / 66.2±3.8	63.1±4.2 / 66.8±3.3
optd	99.2±0.3 / 99.5±0.2	96.2±3.3 / 96.2±3.3	99.0±1.7 / 99.1±1.8	99.0±1.7 / 99.1±1.8	98.9±1.8 / 99.1±1.7	99.1±1.2 / 99.6±0.4
vehi	93.6±1.9 / 97.7±1.0	93.2±2.0 / 96.0±2.1	93.7±2.0 / **98.1±0.8**	93.7±1.9 / **98.0±0.8**	93.6±1.9 / 97.8±1.0	93.7±2.0 / **97.9±0.8**
wdbc	95.2±1.4 / 93.6±1.7	93.4±2.6 / 92.4±2.7	94.6±1.7 / 93.6±1.7	94.6±1.7 / 93.5±1.7	94.6±1.7 / 93.4±1.7	94.7±1.7 / 93.6±1.7
digi	90.8±0.7 / 94.2±1.5	**92.0±1.6** / 94.5±1.9	**93.7±1.4** / **96.2±1.1**	**91.8±1.6** / **95.2±1.9**	**91.6±1.5** / **95.2±1.3**	**91.5±0.8** / **94.9±1.4**
USPS	87.2±1.0 / 83.1±1.9	86.7±1.5 / **91.7±2.5**	**89.1±0.6** / **92.5±2.0**	88.0±1.1 / **91.9±2.3**	**87.6±0.8** / **91.6±2.2**	**87.7±1.0** / **91.0±2.4**
COIL	80.2±2.2 / 87.1±2.0	80.8±2.7 / 87.0±1.5	**83.2±2.2** / **88.9±2.4**	**81.6±2.5** / 87.1±2.6	**80.9±2.0** / 87.2±2.5	**80.8±2.3** / 87.2±2.1
BCI	70.4±3.4 / 66.2±2.9	70.5±4.1 / 65.4±2.8	**70.7±3.5** / 66.2±2.7	70.6±3.4 / 66.1±2.8	70.4±3.4 / 65.7±2.8	70.5±3.4 / 66.1±2.9
g241c	74.5±2.0 / 70.1±8.5	**80.0±1.4** / **77.8±1.6**	**78.5±2.7** / **79.3±2.7**	**77.8±3.7** / **77.9±4.3**	74.4±2.0 / **76.5±3.7**	**75.3±1.8** / **74.8±4.1**
g241n	71.8±2.7 / 59.4±9.4	**75.4±4.5** / 65.0±13.8	**74.8±2.7** / **74.5±4.3**	**74.2±3.6** / **74.5±4.3**	71.0±3.0 / 53.5±2.5	**72.2±2.8** / **60.9±8.4**
Text	69.8±2.1 / 54.4±4.9	**74.2±1.4** / **58.8±4.9**	**71.3±1.0** / 57.6±3.8	**71.3±1.0** / 53.3±4.5	50.5±5.5 / 49.1±11.0	69.9±2.0 / 54.1±4.1
Avg.Acc.	82.4 / 81.8	82.3 / 82.0	83.2 / 83.6	82.7 / 83.0	81.3 / 81.8	82.5 / 82.5
SVM vs. Semi-Supervised: W/T/L		17/17/10	1/20/23	3/26/15	12/24/8	**0/30/14**

confirm that S4VM$_s$ does not degenerate performance on all the cases in Figure 3.

5. Conclusion

Semi-supervised learning tries to exploit unlabeled data to improve learning performance. Though semi-supervised learning approaches are promising, there are many cases where the performance of using unlabeled data is even worse than purely using the limited labeled data. In this paper, we focus on semi-supervised support vector machines (S3VMs) and propose the S4VMs. Unlike S3VMs which typically try to obtain one low-density separator, S4VMs attempt to exploit multiple candidate diverse large-margin low-density separators and optimize the label assignment for the worst case. We present two implementations, one uses a global simulated annealing search for the low-density separators, while the other uses a simpler and efficient sampling strategy. Comprehensive experiments validate the effectiveness of our S4VMs. It is particularly encouraging since the overall performance of S4VMs is highly competitive to TSVM, while contrasting to TSVM



(a) Influence of T on S4VM$_s$

(b) Influence of N on S4VM$_s$

(c) Influence of λ on S4VM$_s$

Figure 3. Parameter Influence with 10 labeled examples

which often degenerates performance, S4VMs are never significantly worse than inductive SVMs.

Acknowledgments

We want to thank anonymous reviewers for helpful comments and Teng Zhang for help in experiments. This research was supported by the NSFC (61073097), JiangsuSF (BK2008018) and 973 Program (2010CB327903).

References

Bennett, K. and Demiriz, A. Semi-supervised support vector machines. In *NIPS 11*, pp. 368–374. 1999.

Černỳ, V. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *J. Optimization Theory and Application*, 45(1):41–51, 1985.

Chapelle, O. and Zien, A. Semi-supervised learning by low density separation. In *AISTATS*, pp. 57–64, 2005.

Chapelle, O., Chi, M., and Zien, A. A continuation method for semi-supervised SVMs. In *ICML*, pp. 185–192, 2006a.

Chapelle, O., Schölkopf, B., and Zien, A. (eds.). *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006b.

Chapelle, O., Tubingen, G., Sindhwani, V., and Keerthi, S. S. Branch and bound for semi-supervised support vector machines. In *NIPS 20*, pp. 217–224. 2007.

Chapelle, O., Sindhwani, V., and Keerthi, S. S. Optimization techniques for semi-supervised support vector machines. *J. Mach. Learn. Res.*, 9:203–233, 2008.

Cozman, F., Cohen, I., and Cirelo, M. Semi-supervised learning of mixture models. In *ICML*, pp. 99–106, 2003.

De Bie, T. and Cristianini, N. Convex methods for transduction. In *NIPS 16*, pp. 73–80. 2004.

Goutte, C., Déjean, H., Gaussier, E., Cancedda, N., and Renders, J.M. Combining labelled and unlabelled data: A case study on fisher kernels and transductive inference for biological entity recognition. In *CoNLL*, pp. 1–7, 2002.

Grandvalet, Y. and Bengio, Y. Semi-supervised learning by entropy minimization. In *NIPS 17*, pp. 529–536. 2005.

Joachims, T. Transductive inference for text classification using support vector machines. In *ICML*, pp. 200–209, 1999.

Kasabov, N. and Pang, S. Transductive support vector machines and applications in bioinformatics for promoter recognition. In *ICNNSP*, pp. 1–6, 2004.

Kirkpatrick, S. Optimization by simulated annealing: Quantitative studies. *J. Stat. Phys.*, 34(5):975–986, 1984.

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

Li, Y.-F., Kwok, J. T., and Zhou, Z.-H. Semi-supervised learning using label mean. In *ICML*, pp. 633–640, 2009a.

Li, Y.-F., Tsang, I. W., Kwok, J. T., and Zhou, Z.-H. Tighter and convex maximum margin clustering. In *AISTATS*, pp. 344–351, 2009b.

Nigam, K., McCallum, A. K., Thrun, S., and Mitchell, T. Text classification from labeled and unlabeled documents using EM. *Mach. Learn.*, 39(2-3):103–134, 2000.

Sindhwani, V., Keerthi, S.S., and Chapelle, O. Deterministic annealing for semi-supervised kernel machines. In *ICML*, pp. 841–848, 2006.

Vapnik, V. N. *Statistical Learning Theory*. Wiley, New York, 1998.

Wang, L., Chan, K., and Zhang, Z. Bootstrapping SVM active learning by incorporating unlabelled images for image retrieval. In *CVPR*, pp. 629–634, 2003.

Xu, L. and Schuurmans, D. Unsupervised and semi-supervised multi-class support vector machines. In *AAAI*, pp. 904–910, 2005.

Zhang, K., Tsang, I.W., and Kwok, J.T. Maximum margin clustering made practical. In *ICML*, pp. 1119–1126, 2007.

Zhang, T. and Oles, F. The value of unlabeled data for classification problems. In *ICML*, pp. 1191–1198, 2000.

Zhou, Z.-H. Ensemble learning. In Li, S. Z. (ed.), *Encyclopedia of Biometrics*, pp. 270–273. Springer, 2009.

Zhou, Z.-H. and Li, M. Semi-supervised learning by disagreement. *Knowl. Infor. Syst.*, 24(3):415–439, 2010.

Zhu, X. Semi-supervised learning literature survey. Technical Report 1530, Dept. Comp. Sci., Univ. Wisconsin-Madison, 2006.