Semi-Supervised Learning Using Label Mean

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

James T. Kwok JAMESK@CSE.UST.HK
Dept. Computer Science & Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong

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

Abstract

Semi-Supervised Support Vector Machines (S3VMs) typically directly estimate the label assignments for the unlabeled instances. This is often inefficient even with recent advances in the efficient training of the (supervised) SVM. In this paper, we show that S3VMs, with knowledge of the means of the class labels of the unlabeled data, is closely related to the supervised SVM with known labels on all the unlabeled data. This motivates us to first estimate the label means of the unlabeled data. Two versions of the *meanS3VM*, which work by maximizing the margin between the label means, are proposed. The first one is based on multiple kernel learning, while the second one is based on alternating optimization. Experiments show that both of the proposed algorithms achieve highly competitive and sometimes even the best performance as compared to the state-of-the-art semi-supervised learners. Moreover, they are more efficient than existing S3VMs.

1. Introduction

During the past decade, many semi-supervised learning algorithms have been proposed, among which a very popular type of algorithms is the semi-supervised support vector machines (S3VMs). Examples include the semi-supervised SVM (Bennett & Demiriz, 1999), the transductive SVM (TSVM) (Joachims, 1999), and the Laplacian SVM (Belkin et al., 2006). Bennett & Demiriz's S3VM and the TSVM are built upon the cluster assumption and use the unlabeled data to regularize the decision boundary. Specifically, these methods prefer the decision boundary to pass through low-density regions (Chapelle & Zien, 2005). The Laplacian SVM is a S3VM that exploits the data's manifold structure via the graph Laplacian. It encodes both the labeled and unlabeled data by a connected graph, where each instance is represented as a vertex and two vertices are connected by an edge if they have large similarity. The goal is to find class labels for the unlabeled data such that their inconsistencies with both the supervised data and the underlying graph structure are minimized.

However, while many efficient SVM softwares have been developed for supervised learning, S3VMs still suffer from inefficiency issues. In particular, the optimization problem of Bennett and Demiriz's S3VM is formulated as a mixed-integer programming problem and so is computationally intractable in general. TSVM, on the other hand, iteratively solves standard supervised SVM problems. However, the number of iterations required may be large since the TSVM is based on a local combinatorial search that is guided by a label switching procedure. Unlike the TSVM, the Laplacian SVM only needs to solve one small SVM with the labeled data only. However, it needs to calculate the inverse of an $n \times n$ matrix, where n is the size of the whole data set. This costs $O(n^3)$ time and $O(n^2)$ memory.

In this paper, we propose a new approach which may lead to efficient designs of semi-supervised learning algorithms. In contrast to directly estimating the labels of the unlabeled examples, we propose a new S3VM which first estimates the label means of the unlabeled data. We show that this S3VM, referred to as the *meanS3VM*, is nearly the same as the supervised SVM that is provided with all the labels of the unlabeled examples. Roughly speaking, when the data set is separable, the meanS3VM is equivalent to the supervised SVM; otherwise, the loss function of the meanS3VM is no more than twice of that of the supervised SVM. So, instead of estimating the label of ev-

Appearing in *Proceedings of the 26^{th} International Conference on Machine Learning*, Montreal, Canada, 2009. Copyright 2009 by the author(s)/owner(s).

ery unlabeled example, we estimate the label means of the whole unlabeled data. This will be more efficient for semi-supervised learning. Based on this observation, we propose two efficient algorithms by maximizing the margin between the label means of the unlabeled data. Experimental comparisons with state-of-the-art semi-supervised learning methods show that our proposed algorithms achieve highly competitive or even the best performance on a broad range of data sets, including the semi-supervised learning benchmark data sets, UCI data sets and the newsgroup data sets. In particular, on relatively large data (e.g., those with more than 1,000 instances), our proposed algorithms are one hundred times faster than the TSVM and ten times faster than the Laplacian SVM.

The rest of this paper is organized as follows. Section 2 briefly introduces the semi-supervised support vector machines. Section 3 studies the usefulness of the label mean of the unlabeled data. Section 4 proposes two algorithms which use the label mean for the S3VM. Experimental results are reported in Section 5. The last section concludes this paper.

2. Semi-Supervised SVM (S3VM)

In semi-supervised learning, we are given a set of labeled data $\{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_l, y_l)\}$ and a set of unlabeled data $\{\mathbf{x}_{l+1}, \cdots, \mathbf{x}_{l+u}\}$, where l and u are the sizes of the labeled data and unlabeled data, respectively, and $y_i \in \{\pm 1\}$. Let $\mathcal{I}_l = \{1, 2, \ldots, l\}$ be the set containing indices of the labeled data, and $\mathcal{I}_u = \{l+1, l+2, \ldots, l+u\}$ be the set for the unlabeled data. The goal is to find f that minimizes

$$\min_f \frac{1}{2}\|f\|_{\mathcal{H}}^2 + C_1 \sum_{i \in \mathcal{I}_l} \ell(y_i, f(\mathbf{x}_i)) + C_2 \sum_{i \in \mathcal{I}_u} \ell_{sym}(f(\mathbf{x}_i)), \tag{1}$$

where $\mathcal{H}$ is a reproducing kernel Hilbert space (RKHS) induced by the kernel k, $\ell(y, f(\mathbf{x})) = \max\{0, 1 - yf(\mathbf{x})\}$ is the hinge loss, $\ell_{sym}(f(\mathbf{x})) = \max\{0, 1 - |f(\mathbf{x})|\}$ is the symmetric hinge loss, and C_1, C_2 are regularization parameters that balance model complexity with the empirical risks on the labeled and unlabeled data, respectively.

Let the decision function be $f(\mathbf{x}) = \mathbf{w}'\phi(\mathbf{x}) + b$, where ϕ is the feature map induced by the kernel k. By introducing slack variables, (1) can be reformulated as

$$\begin{aligned} \min_{\mathbf{w}, b, \boldsymbol{\xi}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C_1 \sum_{i \in \mathcal{I}_l} \xi_i + C_2 \sum_{i \in \mathcal{I}_u} \xi_i \\ \text{s.t.} \quad & y_i(\mathbf{w}'\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \;\; i \in \mathcal{I}_l, \\ & |\mathbf{w}'\phi(\mathbf{x}_i) + b| \geq 1 - \xi_i, \;\; i \in \mathcal{I}_u, \\ & \xi_i \geq 0, \quad i \in \mathcal{I}_i \cup \mathcal{I}_u, \\ & \sum_{i \in \mathcal{I}_u} \operatorname{sgn}(\mathbf{w}'\phi(\mathbf{x}_i) + b) = r, \end{aligned} \tag{2}$$

where $\boldsymbol{\xi} = [\xi_1, \ldots, \xi_{l+u}]$. The last constraint (with user-defined parameter r) is a balance constraint that avoids the trivial solution of assigning all patterns to the same class and thus achieves "infinite" margin (Chapelle et al., 2008). It is worth noting that the effect of the objective in (2) has been well studied and many algorithms have been developed (Chapelle et al., 2008).

3. Usefulness of the Label Mean

Consider the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{p}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C_1 \sum_{i \in \mathcal{I}_l} \xi_i \\ & + C_2 \sum_{i \in \mathcal{I}_u} (\xi_i + p_{i-l} - |f(\mathbf{x}_i)|) \\ \text{s.t.} \quad & y_i(\mathbf{w}'\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \;\; i \in \mathcal{I}_l, \\ & \mathbf{w}'\phi(\mathbf{x}_i) + b \leq p_{i-l}, \;\; -\mathbf{w}'\phi(\mathbf{x}_i) - b \leq p_{i-l}, \\ & p_{i-l} \geq 1 - \xi_i, \;\; i \in \mathcal{I}_u; \;\; \xi_i \geq 0, \; i \in \mathcal{I}_l \cup \mathcal{I}_u, \\ & \sum_{i \in \mathcal{I}_u} \operatorname{sgn}(\mathbf{w}'\phi(\mathbf{x}_i) + b) = r. \end{aligned} \tag{3}$$

Lemma 1. *Let $(\mathbf{w}^*, b^*, \boldsymbol{\xi}^*, \boldsymbol{p}^*)$ be the optimal solution of (3). Then, for $i \in \mathcal{I}_u$,*

$$\xi_i^* + p_{i-l}^* = \begin{cases} 1 & |f(\mathbf{x}_i)| \leq 1, \\ |f(\mathbf{x}_i)| & \textit{otherwise}. \end{cases}$$

Proof. When $|f(\mathbf{x}_i)| \leq 1$, it is easy to verify that setting $p_{i-l}^* = |f(\mathbf{x}_i)| + \epsilon$ and $\xi_i^* = 1 - |f(\mathbf{x}_i)| - \epsilon$ (where $0 \leq \epsilon \leq 1 - |f(\mathbf{x}_i)|$) satisfies all the constraints in (3). When $|f(\mathbf{x}_i)| > 1$, given $p_{i-l}^* \geq |f(x_i)| \geq 1$, then the second condition $p_{i-l}^* \geq 1 - \xi_i$ is satisfied for any $\xi_i \geq 0$, and $\xi_i^* = 0$ is the choice that minimizes Eq.3. □

Proposition 1. *Problem (3) is equivalent to problem (2).*

Proof. The key is to show that the loss functions in (2) and (3) are the same. Obviously, this is the case for the labeled data. For the unlabeled data ($i \in I_u$), let $\ell'_{sym}(f(\mathbf{x}_i)) = \xi_i + p_{i-l} - |f(\mathbf{x}_i)|$. From Lemma 1,

$$\xi_i + p_{i-l} - |f(\mathbf{x}_i)| = \begin{cases} 1 - |f(\mathbf{x}_i)| & |f(\mathbf{x}_i)| \leq 1, \\ 0 & \text{otherwise}. \end{cases}$$

Note that the RHS is the same as the symmetric hinge loss $\ell_{sym}(f(\mathbf{x}_i))$, while the LHS is the same as $\ell'_{sym}(f(\mathbf{x}_i))$. Therefore, $\ell'_{sym}(f(\mathbf{x}_i)) = \ell_{sym}(f(\mathbf{x}_i))$. □

From the balance constraint, the numbers of positive and negative instances in the unlabeled data can be obtained as $u_+ = \frac{r+u}{2}$ and $u_- = \frac{-r+u}{2}$, respectively. Let the true label of the unlabeled pattern i be y_i^*. The (true) means for the positive and negative classes are then $\mathbf{m}_+ =$

$\frac{1}{u_+}\sum_{y_i^*=1}\phi(\mathbf{x}_i)$ and $\mathbf{m}_- = \frac{1}{u_-}\sum_{y_i^*=-1}\phi(\mathbf{x}_i)$, respectively. Now,

$$\begin{aligned}&\sum_{i\in\mathcal{I}_u}|f(\mathbf{x}_i)| - (u_- - u_+)b\\ &= \mathbf{w}'\left(\sum_{i\in\mathcal{I}_u, f(\mathbf{x}_i)\geq 0}\phi(\mathbf{x}_i) - \sum_{i\in\mathcal{I}_u, f(\mathbf{x}_i)<0}\phi(\mathbf{x}_i)\right)\\ &= u_+\mathbf{w}'\hat{\mathbf{m}}_+ - u_-\mathbf{w}'\hat{\mathbf{m}}_-, \qquad (4)\end{aligned}$$

where $\hat{\mathbf{m}}_+ = \frac{1}{u_+}\sum_{i\in\mathcal{I}_u, f(\mathbf{x}_i)\geq 0}\phi(\mathbf{x}_i)$ and $\hat{\mathbf{m}}_- = \frac{1}{u_-}\sum_{i\in\mathcal{I}_u, f(\mathbf{x}_i)<0}\phi(\mathbf{x}_i)$ are estimates of $\mathbf{m}_+$ and $\mathbf{m}_-$. Using (4), the objective of (3) can be rewritten as

$$\begin{aligned}&\frac{1}{2}\|\mathbf{w}\|^2 + C_1\sum_{i\in\mathcal{I}_l}\xi_i + C_2\sum_{i\in\mathcal{I}_u}(\xi_i + p_{i-l})\\ &-C_2(u_+\mathbf{w}'\hat{\mathbf{m}}_+ - u_-\mathbf{w}'\hat{\mathbf{m}}_- + (u_+ - u_-)b).\end{aligned}$$

On replacing the estimates $\hat{\mathbf{m}}_+$ and $\hat{\mathbf{m}}_-$ by the ground truth values, we obtain the following optimization problem:

$$\begin{aligned}\min_{\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{p}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C_1\sum_{i\in\mathcal{I}_l}\xi_i + C_2\sum_{i\in\mathcal{I}_u}(\xi_i + p_{i-l}) \quad (5)\\ & -C_2(u_+\mathbf{w}'\mathbf{m}_+ - u_-\mathbf{w}'\mathbf{m}_- + (u_+ - u_-)b)\\ \text{s.t.} \quad & \text{constraints in (3)} .\end{aligned}$$

Proposition 2. *In (5), the loss for an unlabeled $\mathbf{x}_i$ is*

$$\tilde{\ell}(\mathbf{x}_i) = \begin{cases} -2y_i^* f(\mathbf{x}_i) & y_i^* f(\mathbf{x}_i) < 0, |f(\mathbf{x}_i)| \geq 1,\\ \ell(y_i^*, f(\mathbf{x}_i)) & \textit{otherwise}.\end{cases} \quad (6)$$

Proof. The objective in (5) can also be rewritten as $\frac{1}{2}\|\mathbf{w}\|^2 + C_1\sum_{i\in\mathcal{I}_l}\xi_i + C_2\sum_{i\in\mathcal{I}_u}(\xi_i + p_{i-l} - y_i^* f(\mathbf{x}_i))$. Hence, $\tilde{\ell}(\mathbf{x}_i) = \xi_i + p_{i-l} - y_i^* f(\mathbf{x}_i)$. From Lemma 1,

$$\tilde{\ell}(\mathbf{x}_i) = \begin{cases} 1 - y_i^* f(\mathbf{x}_i) & |f(\mathbf{x}_i)| \leq 1,\\ 0 & |f(\mathbf{x}_i)| \geq 1, y_i^* f(\mathbf{x}_i) \geq 0,\\ -2y_i^* f(\mathbf{x}_i) & |f(\mathbf{x}_i)| \geq 1, y_i^* f(\mathbf{x}_i) < 0,\end{cases}$$

which is equivalent to (6). □

Corollary 1. *(Separable case) If the data is separable, the loss in (6) is the same as the hinge loss for sample $\mathbf{x}_i$ w.r.t. its true label.*

Proof. When the data is separable, (6) becomes $\tilde{\ell}(\mathbf{x}_i) = \ell(y_i^*, f(\mathbf{x}_i))$ which is the same as the hinge loss in the standard SVM. □

Corollary 2. *(Non-separable case) If the data is non-separable, the loss in (6) is no more than twice of that of the hinge loss w.r.t. the true label.*

Proof. (5) is upper-bounded by $\max\{-2y_i^* f(\mathbf{x}_i), 1 - y_i^* f(\mathbf{x}_i)\}$, while the hinge loss is $\ell(y_i^*, f(\mathbf{x}_i)) = 1 - y_i^* f(\mathbf{x}_i)$. Since $-2y_i^* f(\mathbf{x}_i) < 2(1 - y_i^* f(\mathbf{x}_i))$, hence $\tilde{\ell}(\mathbf{x}_i) < 2\ell(y_i^*, f(\mathbf{x}_i))$. □

Figure 1 compares the loss in (6) with the hinge loss.

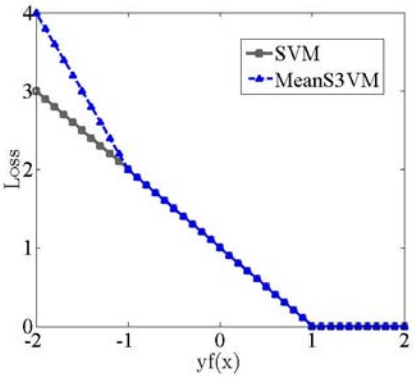

Figure 1. Plots of $yf(\mathbf{x})$, for the loss in (6) and the hinge loss.

4. S3VM Training via the Label Mean

Results in Section 3 suggests that the label mean, being a simple statistic, can be useful in building a semi-supervised learner. To find the label means of the unlabeled data, we propose in this section a margin-based framework such that the margin between the means is maximized. Mathematically, it is formulated as:

$$\begin{aligned}\min_{\mathbf{d}\in\Delta}\min_{\mathbf{w},b,\rho,\boldsymbol{\xi}} \quad & \frac{1}{2}\|\mathbf{w}\|_2^2 + C_1\sum_{i=1}^{l}\xi_i - C_2\rho \qquad (7)\\ \text{s.t.} \quad & y_i(\mathbf{w}'\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i,\ i = 1, \ldots, l,\\ & \frac{1}{u_+}\left(\mathbf{w}'\sum_{j=l+1}^{l+u} d_{j-l}\phi(\mathbf{x}_j)\right) + b \geq \rho,\\ & \frac{1}{u_-}\left(\mathbf{w}'\sum_{j=l+1}^{l+u}(1 - d_{j-l})\phi(\mathbf{x}_j)\right) + b \leq -\rho.\end{aligned}$$

Here, $\Delta = \{\mathbf{d} \mid d_i \in \{0, 1\}, \sum_{i=1}^{u} d_i = u_+\}$. Note that the balance constraint is incorporated into Δ. Moreover, by focusing on the label means, we no longer need to have one constraint for each unlabeled sample. Consequently, (7) has much fewer constraints than (3). As will be verified empirically in Section 5, this allows (7) to be trained much faster than existing S3VM implementations.

Besides, (7) can also be motivated from the Hilbert space embedding of distributions (Gretton et al., 2006). Here, we map the positive class to one point in the RKHS and the negative class to another point. Then, we try to separate the means with maximum margin. Intuitively, the further part these two class distributions are, the easier the classification problem becomes.

However, note that (7) is non-convex due to the bilinear constraint between $\mathbf{w}$ and $\mathbf{d}$. In the following, we propose two algorithms to solve (7). The first one is based on convex relaxation (Li et al., 2009), while the second one is based on alternating optimization.

4.1. Convex Relaxation

4.1.1. Formulating as Kernel Learning

First, we replace the inner problem of (7) by its dual:

$$\min_{\mathbf{d}\in\Delta}\max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad \boldsymbol{\alpha}'\tilde{\mathbf{1}} - \tfrac{1}{2}(\boldsymbol{\alpha}\odot\tilde{\mathbf{y}})'\mathbf{K}^{\mathbf{d}}(\boldsymbol{\alpha}\odot\tilde{\mathbf{y}}), \tag{8}$$

where $\odot$ denotes the element-wise product of two matrices, $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_{l+2}]' \in \mathbb{R}^{l+2}$, $\tilde{\mathbf{y}} = [y_1, \ldots, y_l, 1, -1]' \in \mathbb{R}^{l+2}$, $\tilde{\mathbf{1}} = [\mathbf{1}_l', \mathbf{0}_2']' \in \mathbb{R}^{l+2}$,

$$\begin{aligned}\mathcal{A} \;=\; & \Big\{\boldsymbol{\alpha} \mid \sum_{i=1}^{l+2}\alpha_i\tilde{y}_i = 0, \sum_{i=l+1}^{l+2}\alpha_i = C_2, \\ & 0 \le \alpha_i \le C_1, \forall i = 1, \ldots, l, \\ & 0 \le \alpha_{l+1}, \alpha_{l+2} \le C_2\Big\},\end{aligned} \tag{9}$$

and $\mathbf{K}^{\mathbf{d}} \in \mathbb{R}^{(l+2)\times(l+2)}$ is the kernel matrix with elements $\mathbf{K}^{\mathbf{d}}_{ij} = (\boldsymbol{\phi}^{\mathbf{d}}_i)'(\boldsymbol{\phi}^{\mathbf{d}}_j)$, where

$$\boldsymbol{\phi}^{\mathbf{d}}_i = \begin{cases} \phi(\mathbf{x}_i) & i = 1, \ldots, l, \\ \sum_{j=l+1}^{l+u} d_{j-l}\phi(\mathbf{x}_j)/u_+ & i = l+1, \\ \sum_{j=l+1}^{l+u}(1 - d_{j-l})\phi(\mathbf{x}_j)/u_- & i = l+2. \end{cases} \tag{10}$$

Note that (8) is a mixed-integer programming problem, and so is computationally intractable in general.

Here, we consider a minimax convex relaxation of (8) (Li et al., 2009). Using the minimax inequality, (8) is lower-bounded by

$$\max_{\boldsymbol{\alpha}\in\mathcal{A}}\min_{\mathbf{d}\in\Delta} \quad \boldsymbol{\alpha}'\tilde{\mathbf{1}} - \frac{1}{2}(\boldsymbol{\alpha}\odot\tilde{\mathbf{y}})'\mathbf{K}^{\mathbf{d}}(\boldsymbol{\alpha}\odot\tilde{\mathbf{y}}). \tag{11}$$

The inner minimization (over $\mathbf{d}$) can also be written as

$$\begin{aligned} \max\nolimits_{\theta} \quad & \boldsymbol{\alpha}'\tilde{\mathbf{1}} - \theta \\ \text{s.t.} \quad & \theta \ge \frac{1}{2}(\boldsymbol{\alpha}\odot\tilde{\mathbf{y}})'\mathbf{K}^{\mathbf{d}_t}(\boldsymbol{\alpha}\odot\tilde{\mathbf{y}}), \forall \mathbf{d}_t \in \Delta. \end{aligned} \tag{12}$$

Let $\mu_t \ge 0$ be the dual variable for each constraint in (12). It can be shown that (11) can be rewritten as

$$\min_{\boldsymbol{\mu}\in\mathcal{M}}\max_{\boldsymbol{\alpha}\in\mathcal{A}} \boldsymbol{\alpha}'\tilde{\mathbf{1}} - \frac{1}{2}(\boldsymbol{\alpha}\odot\tilde{\mathbf{y}})'\Big(\sum_{t:\mathbf{d}_t\in\Delta}\mu_t\mathbf{K}^{\mathbf{d}_t}\Big)(\boldsymbol{\alpha}\odot\tilde{\mathbf{y}}). \tag{13}$$

Here, $\mathcal{M}$ is the simplex $\{\boldsymbol{\mu} \mid \sum \mu_t = 1, \mu_t \ge 0\}$.

Note that (13) is of the same form as the optimization problem in multiple kernel learning (MKL) (Lanckriet et al., 2004). However, note that (12) involves minimizing over all $\mathbf{d}_t \in \Delta$. There are an exponential number of feasible $\mathbf{d}_t$'s, and hence the set of base kernels is also exponential in size and so direct MKL is computationally intractable.

In this paper, we apply the cutting plane method (Kelly, 1960), which has been successfully used in many optimization problems in machine learning (Joachims et al., 2009; Tsochantaridis et al., 2005), to handle the exponential number of constraints. The algorithm is shown in Algorithm 1. First, we initialize $\boldsymbol{\alpha}$ and the set of cutting planes $\mathcal{C}$. Since the size of $\mathcal{C}$ is no longer exponential, we can perform MKL with the subset of kernel matrices in $\mathcal{C}$ and obtain $\boldsymbol{\alpha}$ from (13). The most violated $\mathbf{d}$ is then added to $\mathcal{C}$, and the process repeated until convergence.

Algorithm 1 Cutting plane algorithm for solving the convex relaxation in (11).

```
1: Initialize $\boldsymbol{\alpha} = \frac{C_2}{2}\mathbf{1}$, find the most violate $\mathbf{d}_0$, and set $\mathcal{C} = \{\mathbf{d}_0\}$
2: Run MKL for the subset of kernel matrices selected in $\mathcal{C}$ and obtain $\boldsymbol{\alpha}$ from dual problem of (7).
3: Find the most violated $\mathbf{d}$ and set $\mathcal{C} = \mathbf{d}\bigcup\mathcal{C}$.
4: Repeat steps 2-3 until convergence.
```

There are two important issues in the cutting plane algorithm. First, how to efficiently solve the MKL problem in step 2? Second, how to efficiently find the most violated $\mathbf{d}$ in step 3? These will be addressed in the next two sections.

4.1.2. Solving the MKL Problem in Step 2

In recent years, a number of MKL methods have been proposed in the literature (Bach et al., 2004; Lanckriet et al., 2004; Sonnenburg et al., 2006; Rakotomamonjy et al., 2008; Xu et al., 2009). In this paper, we use an adaptation of the SimpleMKL algorithm (Rakotomamonjy et al., 2008) to solve the MKL problem in Step 2 of Algorithm 1.

Recall that the feature map induced by the base kernel matrix $\mathbf{K}^{\mathbf{d}_t}$ is given in (10). As in the derivation of the SimpleMKL algorithm, we consider the following MKL problem in (13).

$$\begin{aligned} \min_{\boldsymbol{\mu}\in\mathcal{M}}\min_{\mathbf{w},\boldsymbol{\xi}} \quad & \frac{1}{2}\sum_{t=1}^{T}\frac{\|\mathbf{w}_t\|^2}{\mu_t} + C_1\sum_{i\in\mathcal{I}_l}\xi_i - C_2\rho \\ \text{s.t. } & y_i\left(\sum_{t=1}^{T}\mathbf{w}_t'\phi(\mathbf{x}_i) + b\right) \ge 1 - \xi_i, \; i \in \mathcal{I}_l, \\ & \frac{1}{u_+}\left(\sum_{t=1}^{T}\mathbf{w}_t'\sum_{j\in\mathcal{I}_u}d^t_{j-l}\phi(\mathbf{x}_j)\right) + b \ge \rho, \\ & \frac{1}{u_-}\left(\sum_{t=1}^{T}\mathbf{w}_t'\sum_{j\in\mathcal{I}_u}(1 - d^t_{j-l})\phi(\mathbf{x}_j)\right) + b \le -\rho. \end{aligned} \tag{14}$$

It is easy to verify that its dual is (11). Following the SimpleMKL, we solve (13), or equivalently, (14), iteratively. First, we fix $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_T]'$ and solve the dual of the inner problem in (14):

$$\max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad \boldsymbol{\alpha}'\tilde{\mathbf{1}} - \tfrac{1}{2}(\boldsymbol{\alpha}\odot\tilde{\mathbf{y}})'\Big(\sum_{t=1}^{T}\mu_t\mathbf{K}^{\mathbf{d}_t}\Big)(\boldsymbol{\alpha}\odot\tilde{\mathbf{y}}).$$

Note that this is simply the SVM's dual, with kernel matrix $\sum_{t=1}^{T} \mu_t \mathbf{K}^{\mathbf{d}_t}$ and label vector $\tilde{\mathbf{y}}$. Hence, it can be solved efficiently with existing SVM solvers. Then, we fix $\boldsymbol{\alpha}$ and use the reduced gradient method to update $\boldsymbol{\mu}$. These two steps are iterated until convergence.

4.1.3. Finding the most violated $\mathbf{d}$ in Step 3

From (13), the most violated $\mathbf{d}$ maximizes $(\boldsymbol{\alpha}\odot\tilde{\mathbf{y}})'\mathbf{K}^{\mathbf{d}}(\boldsymbol{\alpha}\odot\tilde{\mathbf{y}}) = \sum_{i,j=1}^{l+2} \alpha_i\alpha_j\tilde{y}_i\tilde{y}_j(\boldsymbol{\phi}_i^{\mathbf{d}})'(\boldsymbol{\phi}_j^{\mathbf{d}})$. However, this is a concave QP and so cannot be solved efficiently. Note, however, that the cutting plane algorithm only requires the addition of a violated constraint at each iteration. Hence, we propose in the following a simple and efficient method for finding a good approximation of the most violated $\mathbf{d}$.

Using (10), it is easy to verify that

$$\begin{aligned} &\sum_{i,j=1}^{l+2} \alpha_i\alpha_j\tilde{y}_i\tilde{y}_j(\boldsymbol{\phi}_i^{\mathbf{d}})'(\boldsymbol{\phi}_j^{\mathbf{d}}) \qquad (15)\\ &= \|\alpha_i\tilde{y}_i\phi(\mathbf{x}_i)\|_2^2 + \alpha_{l+1}\sum_{i=1}^{l}\alpha_i\tilde{y}_i(\boldsymbol{\phi}_i^{\mathbf{d}})'(\boldsymbol{\phi}_{l+1}^{\mathbf{d}})\\ &\quad -\alpha_{l+2}\sum_{i=1}^{l}\alpha_i\tilde{y}_i(\boldsymbol{\phi}_i^{\mathbf{d}})'(\boldsymbol{\phi}_{l+2}^{\mathbf{d}}) + \|\alpha_{l+1}\boldsymbol{\phi}_{l+1}^{\mathbf{d}} - \alpha_{l+2}\boldsymbol{\phi}_{l+2}^{\mathbf{d}}\|_2^2. \end{aligned}$$

The first term is not related to $\mathbf{d}$ and so can be removed. Moreover, since labeled data are often more reliable than unlabeled data (Joachims, 1999), so $C_2 \ll C_1$. Consequently, from (9), $\alpha_{l+1}, \alpha_{l+2}$ are much smaller than the other α_i's and the last term in (15) can be dropped. Thus, as an approximation, we simply maximize

$$\alpha_{l+1}\sum_{i=1}^{l}\alpha_i\tilde{y}_i(\boldsymbol{\phi}_i^{\mathbf{d}})'(\boldsymbol{\phi}_{l+1}^{\mathbf{d}}) - \alpha_{l+2}\sum_{i=1}^{l}\alpha_i\tilde{y}_i(\boldsymbol{\phi}_i^{\mathbf{d}})'(\boldsymbol{\phi}_{l+2}^{\mathbf{d}}).$$

On using (10) again, this can be rewritten as

$$\begin{aligned} &\frac{\alpha_{l+1}}{u_+}\sum_{i=1}^{l}\alpha_i\tilde{y}_i\sum_{j=l+1}^{l+u} d_{j-l}k(\mathbf{x}_i,\mathbf{x}_j)\\ &-\frac{\alpha_{l+2}}{u_-}\sum_{i=1}^{l}\alpha_i\tilde{y}_i\sum_{j=l+1}^{l+u}(1-d_{j-l})k(\mathbf{x}_i,\mathbf{x}_j)\\ = \;&(\frac{\alpha_{l+1}}{u_+} - \frac{\alpha_{l+2}}{u_-})\sum_{j=l+1}^{l+u} d_{j-l}\sum_{i=1}^{l}\alpha_i\tilde{y}_i k(\mathbf{x}_i,\mathbf{x}_j)\\ &-\frac{\alpha_{l+2}}{u_-}\sum_{i=1}^{l}\alpha_i\tilde{y}_i\sum_{j=l+1}^{l+u} k(\mathbf{x}_i,\mathbf{x}_j). \qquad (16) \end{aligned}$$

Since the second term and $(\frac{\alpha_{l+1}}{u_+} - \frac{\alpha_{l+2}}{u_-})$ are not related to $\mathbf{d}$, the maximization of (16) can be rewritten as

$$\max_{\mathbf{d}\in\Delta}\sum_{j=l+1}^{l+u} d_{j-l}\sum_{i=1}^{l}\alpha_i\tilde{y}_i k(\mathbf{x}_i,\mathbf{x}_j).$$

This can be easily solved as follows. First, we sort the $\sum_{i=1}^{l}\alpha_i\tilde{y}_i k(\mathbf{x}_i,\mathbf{x}_j)$'s in descending order. Then, those d_j's with the corresponding j among the largest u_+ values are assigned 1, while all others are assigned 0.

Algorithm 2 Alternating optimization.

```
1: Run the SVM with the labeled data only.
2: Set the $\mathbf{d}$ entries of those unlabeled patterns with the
   largest $u_+$ predictions to 1, otherwise set to 0.
3: Fix $\mathbf{d}$, and optimize the QP in (8).
4: Repeat steps 2-3 until convergence.
```

4.1.4. Final Assignment of $\mathbf{d}$

After training, the prediction of $\mathbf{x}$ is given by $f(\mathbf{x}) = \sum_{t=1}^{T}\mathbf{w}_t\phi(\mathbf{x}) + b$. Unlabeled patterns whose predictions are among the u_+ largest will have their corresponding d_j entries assigned 1, while the others will be assigned 0.

4.2. Alternating Optimization

Another natural way to solve (7) is based on alternating optimization. Note that for a fixed $\mathbf{d}$, the inner maximization of (8) w.r.t. $\boldsymbol{\alpha}$ can be solved efficiently with standard QP solvers. On the other hand, when $\boldsymbol{\alpha}$ is fixed, $\mathbf{w}$ and b can be determined from the KKT conditions, and (7) reduces to

$$\begin{aligned} \max_{\mathbf{d}\in\Delta,\rho} \quad & \rho\\ \text{s.t.} \quad & \frac{1}{u_+}\left(\mathbf{w}'\sum_{j=l+1}^{l+u} d_{j-l}\phi(\mathbf{x}_i)\right) + b \geq \rho,\\ & \frac{1}{u_-}\left(\mathbf{w}'\sum_{j=l+1}^{l+u}(1-d_{j-l})\phi(\mathbf{x}_i)\right) + b \leq -\rho. \end{aligned}$$

Proposition 3. *At optimality,* $d_{i-l} \geq d_{j-l}$ *if* $f(\mathbf{x}_i) > f(\mathbf{x}_j)$, $\forall i,j \in \mathcal{I}_u$

Proof. In order to maximize ρ, we want to increase $\sum_{q=l+1}^{l+u} d_{q-l}f(\mathbf{x}_q)$ and decrease $\sum_{q=l+1}^{l+u}(1 - d_{q-l})f(\mathbf{x}_q)$ as much as possible. Assume, to the contrary, that the optimal $\mathbf{d}$ does not have the same sorted order as $\{f(\mathbf{x}_j)\}_{j=l+1}^{l+u}$. Then, there are two entries of $\mathbf{d}$, say, d_{i-l} and d_{j-l} (both $\geq$ 0), with $f(\mathbf{x}_i) \geq f(\mathbf{x}_j)$ but $d_{i-l} < d_{j-l}$. Then $\sum_{q=l+1}^{l+u} d_{q-l}f(\mathbf{x}_q) = \sum_{q\neq i,j} d_{q-l}f(\mathbf{x}_q) + d_{i-l}f(\mathbf{x}_i) + d_{j-l}f(\mathbf{x}_j) < \sum_{q\neq i,j} d_{q-l}f(\mathbf{x}_q) + d_{j-l}f(\mathbf{x}_i) + d_{i-l}f(\mathbf{x}_j)$. A similar result holds for $\sum_{q=l+1}^{l+u}(1 - d_{q-l})f(\mathbf{x}_q)$. Thus, interchanging the values of d_{i-l} and d_{j-l} will increase $\sum_{q=l+1}^{l+u} d_{q-l}f(\mathbf{x}_q)$ and decrease $\sum_{q=l+1}^{l+u}(1-d_{q-l})f(\mathbf{x}_q)$, and so $\mathbf{d}$ is not optimal, a contradiction. □

Using this proposition, unlabeled patterns whose predictions are among the u_+ largest will have their corresponding entries in $\mathbf{d}$ set to one; otherwise they will be set to zero (Algorithm 2). Note that, unlike convex relaxation, alternating optimization may get stuck in a local minimum. On the other hand, its computation is very simple and, empirically, much faster.

Table 1. Accuracy (%) on benchmark data sets. The number in parentheses shows the relative rank of the algorithm (performance-wise) on the corresponding data set. The smaller the rank, the better the relative performance. The best performance on each data set is bolded.

#labeled	Method	g241c	g241d	Digit1	USPS	BCI	Text	Total rank
10	1-NN	52.12(8)	53.28(7)	86.35(3)	**83.34**(1)	51.00(5)	61.82(7)	31
	SVM	52.66(7)	53.34(6)	69.40(9)	79.97(6)	50.15(9)	54.63(9)	46
	TSVM	**75.29**(1)	49.92(8)	82.23(7)	74.80(9)	50.85(6)	68.79(2)	33
	Cluster-Kernel	51.72(9)	57.95(2)	81.27(8)	80.59(5)	51.69(3)	57.28(8)	35
	LDS	71.15(3)	49.37(9)	84.37(4)	82.43(2)	50.73(8)	63.85(5)	31
	Laplacian RLS	56.05(5)	54.32(5)	**94.56**(1)	81.01(3)	51.03(4)	66.32(4)	22
	Laplacian SVM	53.79(6)	54.85(4)	91.03(2)	80.95(4)	50.75(7)	62.72(6)	29
	meanS3vm-*iter*	72.22(2)	57.00(3)	82.98(6)	76.34(8)	51.88(2)	**69.57**(1)	22
	meanS3vm-*mkl*	65.48(4)	**58.94**(1)	83.00(5)	77.84(7)	**52.07**(1)	66.91(3)	21
100	1-NN	56.07(9)	57.55(9)	96.11(5)	94.19(4)	51.33(9)	69.89(9)	45
	SVM	76.89(6)	75.36(6)	94.47(8)	90.25(8)	65.69(6)	73.55(8)	42
	TSVM	81.54(3)	77.58(2)	93.85(9)	90.23(9)	66.75(5)	75.48(7)	35
	Cluster-Kernel	**86.51**(1)	**95.05**(1)	96.21(4)	90.32(7)	64.83(7)	75.62(6)	26
	LDS	81.96(2)	76.26(5)	96.54(3)	95.04(3)	56.03(8)	**76.85**(1)	22
	Laplacian RLS	75.64(8)	73.54(8)	**97.08**(1)	**95.32**(1)	68.64(3)	76.43(4)	25
	Laplacian SVM	76.18(7)	73.64(7)	96.87(2)	95.30(2)	67.61(4)	76.14(5)	27
	meanS3vm-*iter*	80.00(5)	77.52(4)	95.68(7)	93.83(5)	71.31(2)	76.74(2)	25
	meanS3vm-*mkl*	80.25(4)	77.58(2)	95.91(6)	93.17(6)	**71.44**(1)	76.60(3)	22

4.3. MeanS3VM

The obtained $\mathbf{d}$ vector can be used to assign labels for the unlabeled data. We take these labels, together with the labels of the labeled data, to train a final SVM. The final SVMs, obtained with the $\mathbf{d}$ from Algorithms 1 and 2, will be denoted means3vm-*mkl* and means3vm-*iter*, respectively. It is worth noting that the MeanS3VM bears some resemblance with (Grandvalet et al., 2009), where one does not estimate the labels of some examples.

5. Experiments

The proposed algorithms are evaluated on the benchmark data sets used in (Chapelle et al., 2006) (Section 5.1), a number of UCI data sets (Section 5.2) and a text data set (Section 5.3). We also empirically study the computational efficiency in Section 5.4.

5.1. Benchmark Data Sets

We first perform experiments on the benchmark data sets (http://www.kyb.tuebingen.mpg.de/ssl-book/) in (Chapelle et al., 2006). These include g241c, g241d, Digit1, USPS, BCI and Text. There are two settings, one using 10 labeled examples and the other using 100 labeled examples. For each data set and each setting, there are twelve subsets. The average accuracy on the unlabeled data will be reported.

The settings of the proposed methods are the same as that of the TSVM in (Chapelle et al., 2006). Specifically, C_1 is fixed to 100, both the linear and Gaussian kernels are used and then have the better results reported. Besides, C_2 is fixed to 0.1. Following (Chapelle et al., 2008), μ_+, μ_- are set to the ground truth. When there are only 10 labeled examples, we simply set the Gaussian kernel width to the average distance between patterns. When there are 100 labeled examples, the kernel parameter is selected via ten-fold cross-validation.

Experimental results are shown in Table 1. We also include four other SVM-type methods from (Chapelle et al., 2006), including (1) SVM using labeled data only, (2) TSVM (Joachims, 1999), (3) SVM with the cluster kernel (Weston et al., 2005), and (4) Laplacian RLS/SVM (Belkin et al., 2006). In addition, we also compare with (5) the classic 1-nearest neighbor classifier, and (6) low-density separation (LDS) (Chapelle & Zien, 2005). Since the setup is the same as in (Chapelle et al., 2006), the results for these methods are simply taken from (Chapelle et al., 2006).

As can be seen, the proposed methods achieve highly competitive performance with the other state-of-the-art semi-supervised learning methods. In particular, means3vm-*mkl* is one of the most accurate algorithms.

5.2. UCI Data Sets

In this section, we evaluate the proposed algorithms on 9 UCI data sets. The experiment setup follows that in (Mallapragada et al., 2009). Each data set is split into two equal halves, with one for training and the other for testing. Each training data set contains ten labeled examples and the rest are used as unlabeled examples. The experiment is repeated 20 times and the average results reported. In addition to the SVM (using labeled data only), TSVM

Table 2. Accuracy (%) on UCI data. The numbers in parentheses show the number of instances (n) and dimensionality (d). The best performance on each data set is bolded.

Data set (n, d)	SVM	SB-SVM	LDS	TSVM	LapSVM	means3vm-*iter*	means3vm-*mkl*
house (232,16)	91.16	90.65	89.35	86.55	89.95	91.72	**91.90**
heart (270,9)	70.59	**79.00**	77.11	77.63	77.96	74.56	73.22
vehicle (435,26)	78.28	72.29	66.28	63.62	71.38	**82.47**	82.15
wdbc (569,14)	75.74	88.82	85.07	86.40	**91.07**	79.39	80.19
isolet (600,51)	89.58	95.12	92.07	90.38	93.93	98.75	**98.98**
austra (690,15)	65.64	71.36	66.00	73.38	**74.38**	68.12	67.59
optdigits (1143,42)	90.31	96.35	96.40	92.34	98.34	98.93	**99.09**
ethn (2630,30)	67.04	67.57	67.16	54.69	**74.60**	73.21	73.57
sat (3041,36)	99.13	87.71	94.20	98.26	99.12	**99.56**	**99.56**

Table 3. Accuracy (%) on text categorization tasks. The best performance on each data set is bolded.

	SB-			Lap-	means3vm	
Classes	SVM	TSVM	LDS	SVM	*-iter*	*-mkl*
(1,2)	70.74	75.44	55.10	68.23	**84.72**	84.27
(1,3)	74.83	89.34	58.88	71.34	90.54	**90.83**
(1,4)	78.47	88.71	61.72	74.67	88.33	**88.76**
(1,5)	82.64	**92.35**	66.45	78.01	91.10	91.14
(2,3)	64.06	66.05	50.76	61.68	66.48	**66.73**
(2,4)	74.85	81.50	50.32	70.95	**81.77**	81.71
(2,5)	80.12	**84.94**	53.94	74.79	77.13	77.37
(3,4)	75.26	81.98	50.08	71.45	**84.47**	84.12
(3,5)	78.31	77.38	53.83	74.91	**81.65**	80.36
(4,5)	68.07	67.54	52.39	65.05	66.45	**72.85**

and Laplacian SVM (LapSVM), we also compare with the Semi-Boost SVM (SB-SVM) (Mallapragada et al., 2009) and Inductive LDS (Chapelle & Zien, 2005). As in (Mallapragada et al., 2009), parameter C_1 is set to 1 and the linear kernel is used for all the SVMs. C_2 is fixed to 0.1.

As can be seen from Table 2, the proposed algorithms achieve highly competitive performance on all data sets. In particular, means3vm-*iter* is the best on 2 of the 9 tasks, while means3vm-*mkl* is the best on 4 tasks.

5.3. Text Categorization

In this section, we evaluate the various methods using the popular 20-newsgroups data set (http://people.csail.mit.edu/jrennie/20Newsgroups/). Following Mallapragada et al. (2009), we use five of the twenty classes, and generate ten tasks from these five classes by the one-vs-one strategy. The experimental setup is as same as that in Section 5.2, except that each training data set now has only two labeled examples per class.

Table 3 shows the results. As can be seen, the proposed algorithms achieve highly competitive or sometimes even the best performance on these data sets.

Table 4. Wall clock time (in seconds). The smallest time cost on each data set is bolded.

			means3vm	
Data set	TSVM	LapSVM	*-iter*	*-mkl*
BCI	73.88	**0.19**	0.27	2.45
Text	6181.12	17.27	**0.55**	14.12
g241d	596.23	5.88	**0.53**	0.94
g241c	552.19	7.08	**1.77**	2.17
Digit1	1222.90	6.54	**0.50**	0.83
USPS	560.05	7.48	**0.58**	1.25
house	3.19	**0.09**	**0.09**	0.77
heart	13.12	**0.06**	0.09	0.52
vehicle	34.46	0.20	**0.11**	0.65
wdbc	123.02	**0.29**	0.50	0.56
isolet	62.10	0.55	**0.19**	0.97
austra	44.37	0.40	**0.26**	0.80
optdigits	114.93	1.53	**0.39**	0.94
ethn	355.30	11.70	**1.09**	2.16
sat	494.38	18.78	**1.08**	1.86
(1,2)	2176.65	13.46	**0.81**	3.33
(1,3)	2151.67	13.48	**0.75**	3.09

5.4. Speed

In this section, we study the computational efficiency of the various methods[1]. All the experiments are performed on a PC with 2GHz Intel Xeon(R)2-Duo running Windows XP with 4GB main memory.

As can be seen from Table 4, TSVM is always slower than the Laplacian SVM and the proposed algorithms. The Laplacian SVM is slightly faster than the means3vm-*mkl* when the data set is small. However, on large data sets (with more than 1,000 instances), the Laplacian SVM is much slower than means3vm-*mkl*. Moreover, means3vm-*iter* is consistently faster than means3vm-*mkl* and is almost always the fastest among all methods. In particular, on the large data sets, means3vm-*iter* is about 100 times faster than TSVM and 10 times faster than Laplacian SVM.

[1] The TSVM and Laplacian SVM implementations are downloaded from http://svmlight.joachims.org/ and http://manifold.cs.uchicago.edu/manifoldregularization/software.html, respectively.

6. Conclusion

In contrast to previous semi-supervised support vector machines that directly estimate the label of every unlabeled example, we propose in this paper a new approach which works by first estimating the label means of the unlabeled data. We show that the semi-supervised SVM with known label means of unlabeled data is closely related to the supervised SVM that has access to all the labels of the unlabeled examples. Based on this observation, we propose two algorithms that maximize the margin between the label means of the unlabeled data. Experiments on a broad range of data sets show that, in comparison with state-of-the-art semi-supervised learning methods, both of our proposed algorithms achieve highly competitive or sometimes even the best performance, and are much faster to train.

The idea of using the label means can be applied to many other learning scenarios, and other approaches for estimating the label means will also be investigated in the future.

Acknowledgments

Thanks to anonymous reviews for their helpful suggestions. This work was supported by the NSFC (60635030, 60721002), JiangsuSF (BK2008018), Jiangsu 333 Program and Hong Kong RGC (614508).

References

Bach, R. R., Lanckriet, G. R. G., & Jordan, M. I. (2004). Multiple kernel learning, conic duality, and the SMO algorithm. *Proceeding of the 21st International Conference on Machine Learning* (pp. 41–48).

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

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

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

Chapelle, O., Sindhwani, V., & Keerthi, S. S. (2008). Optimization techniques for semi-supervised support vector machines. *Journal of Machine Learning Research*, *9*, 203–233.

Chapelle, O., & Zien, A. (2005). Semi-supervised learning by low density separation. *Proceeding of the 8th International Conference on Artificial Intelligence and Statistics* (pp. 57–64).

Grandvalet, Y., Rakotomamonjy, A., Keshet, J., & Canu, S. (2009). Support vector machines with a reject option. In *Advances in Neural Information Processing Systems 21*, 537–544.

Gretton, A., Borgwardt, K. M., Rasch, M., Schölkopf, B., & Smola, A. J. (2006). A kernel method for the two-sample-problem. In *Advances in Neural Information Processing Systems 19*, 513–520.

Joachims, T. (1999). Transductive inference for text classification using support vector machines. *Proceeding of the 16th International Conference on Machine Learning* (pp. 200–209).

Joachims, T., Finley, T., & Yu, C.-N. (2009). Cutting-plane training of structural SVMs. *Machine Learning*, to appear.

Kelly, J. E. (1960). The cutting plane method for solving convex programs. *Journal of Society for Industrial and Applied Mathematics*, *8*, 703–712.

Lanckriet, G. R. G., Cristianini, N., Bartlett, P., Ghaoui, L. E., & Jordan, M. I. (2004). Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, *5*, 27–72.

Li, Y.-F., Tsang, I. W., Kwok, J. T., & Zhou, Z.-H. (2009). Tighter and convex maximum margin clustering. *Proceeding of the 12th International Conference on Artificial Intelligence and Statistics* (pp. 344–351).

Mallapragada, P., Jin, R., Jain, A., & Liu, Y. (2009). Semi-Boost: Boosting for semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, to appear.

Rakotomamonjy, A., Bach, F. R., Canu, S., & Grandvalet, Y. (2008). SimpleMKL. *Journal of Machine Learning Research*, *9*, 2491–2521.

Sonnenburg, S., Rätsch, G., Schäfer, C., & Schölkopf, B. (2006). Large scale multiple kernel learning. *Journal of Machine Learning Research*, *7*, 1531–1565.

Tsochantaridis, I., Joachims, T., Hofmann, T., & Altun, Y. (2005). Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, *6*, 1453–1484.

Weston, J., Leslie, C., Ie, E., Zhou, D., Elisseeff, A., & Noble, W. S. (2005). Semi-supervised protein classification using cluster kernels. *Bioinformatics*, *21*, 3241–3247.

Xu, Z., Jin, R., King, I., & Lyu, M. R. (2009). An extended level method for efficient multiple kernel learning. In *Advances in Neural Information Processing Systems 21*, 1825–1832.