Probabilistic labeled Semi-supervised SVM

Mingjie Qian, Feiping Nie, Changshui Zhang
State Key Laboratory of Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology(TNList)
Department of Automation, Tsinghua University, Beijing 100084,P. R. China
{qian.mingjie,feipingnie}@gmail.com, zcs@mail.tsinghua.edu.cn

***Abstract*—Semi-supervised learning has been paid increasing attention and is widely used in many fields such as data mining, information retrieval and knowledge management as it can utilize both labeled and unlabeled data. Laplacian SVM (LapSVM) is a very classical method whose effectiveness has been validated by large number of experiments. However, LapSVM is sensitive to labeled data and it exposes to cubic computation complexity which limit its application in large scale scenario. In this paper, we propose a multi-class method called Probabilistic labeled Semi-supervised SVM (PLSVM) in which the optimal decision surface is taught by probabilistic labels of all the training data including the labeled and unlabeled data. Then we propose a kernel version dual coordinate descent method to efficiently solve the dual problems of our Probabilistic labeled Semi-supervised SVM and decrease its requirement of memory. Synthetic data and several benchmark real world datasets show that PLSVM is less sensitive to labeling and has better performance over traditional methods like SVM, LapSVM (LapSVM) and Transductive SVM (TSVM).**

***Keywords*-Semi-supervised Learning; Probabilistic Label; Multi-class Classification; Dual Coordinate Descent Algorithm;**

I. Introduction

In many real applications, labeled samples are often time consuming or expensive to obtain. In this scenario, traditional supervised method like SVM [1] is hard to use due to lack of labeled samples. However, in many situations, large numbers of unlabeled data are much easier to collect. For example, in photo genre classification, one could have a very easy access to a large database of pictures, whereas only a small portion of them are labeled manually.

Semi-supervised learning [2] has received increasing attention during recent years since it can utilize both labeled and unlabeled data. In the literature of this field, some semi-supervised methods including inductive methods [3] and transductive methods [4] have been proposed during recent years. LapSVM [3] is a classical method which is based on a form of regularization that exploits the geometry of marginal distribution. The manifold regularization term enables LapSVM to utilize unlabeled data effectively in some situations. However, there are several limitations for LapSVM. First, it is sensitive to data labeling. Given not good labeled data, LapSVM performs badly. Second, it has a cubic computation complexity which limit its use in large scale problems. Third, it is only a binary-class method. In real practical problems, we usually face multi-class problems. Although binary LapSVM can deal with multiple classification using one versus rest or one versus one schemes, it is rather expensive especially with large number of classes. Transductive SVM (TSVM) [4] is also a very famous semi-supervised method which considers the concurrence of components of data. TSVM is suitable for text categorization since words in natural language occur in strong co-occurrence pattern in the field of information retrieval. However, TSVM exposes to several drawbacks. First, TSVM is susceptible to local minima. Second, TSVM solves multiple quadratic programs in the size of the training set with an unknown number of iterations which needs a large amount of time to converge. Third, for non text data, TSVM doesn't perform very well, because for non-text data, the components of data doesn't presents strong co-occurrence pattern apparently. Fourth, TSVM doesn't consider the structure information among data, thus it can't make a rich use of unlabeled data. Besides, TSVM can't deal with multi-class classification directly.

In this paper, we propose a novel method called Probabilistic Labeled Semi-supervised SVM (PLSVM). Unlike the traditional SVM and LapSVM which constrain on the hinge loss on labeled data, we optimize the probabilistic labels of all the training data and utilize the obtained probabilistic labels to punish the hinge loss on the whole training data including labeled data and unlabeled data. By utilizing the probabilistic labels, PLSVM can make a better use of the geometrical structure information of unlabeled data and is less sensitive to data labeling and more robust even under extremely bad labeling. Additionally, in order to efficiently solve the dual problems of PLSVM and decrease its memory requirement, we propose a kernel version dual coordinate descent algorithm. Theoretically, PLSVM as a whole has quadratic complexity. However, in linear kernel, PLSVM has a linear complexity.

Notations Throughout this paper, we assume that there are l labeled data denoted by$\boldsymbol{X^L} = \{(\boldsymbol{x_i}, y_i)\}_{i=1}^{l}$ and u unlabeled data $\boldsymbol{X^U} = \{\boldsymbol{x_j}\}_{j=l+1}^{l+u}$. For binary classification, $y_i \in \{-1, 1\}$, for multi-class classification, $y_i \in \{1, 2, \ldots, C\}$, where C is the number of classes.

We use $\mathcal{K}(\boldsymbol{x_i}, \boldsymbol{x_j})$ to denote the kernel function defined by $\mathcal{K}(\boldsymbol{x_i}, \boldsymbol{x_j}) = \phi(\boldsymbol{x_i})^T \phi(\boldsymbol{x_j})$, where ϕ is the map from data space $\mathcal{X}$ into the Hilbert space of functions mapping $\mathcal{X}$ into $\mathcal{R}$. We use $\boldsymbol{\Phi(X)}$ to denote the mapped data matrix as $\boldsymbol{\Phi(X)} = [\phi(\boldsymbol{x_1}), \phi(\boldsymbol{x_2}), \ldots, \phi(\boldsymbol{x_n})]$. In the semi-supervised situation, Graph Laplacian is widely used to describe the data structure. We define $G = (V, E)$ as the graph associated with the samples. V is the vertex set of graph, which is defined on the training set. E is the edge set containing the pairs of neighboring vertices $(\boldsymbol{x_i}, \boldsymbol{x_j})$. There are two typical methods to calculate the weight matrix or adjacency matrix W:

(1) Gaussian kernel of width σ:

$$W_{ij} = \left\{ \begin{array}{ll} exp\{-\frac{\|\boldsymbol{x_i}-\boldsymbol{x_j}\|^2}{2\sigma^2}\} & if(\boldsymbol{x_i}, \boldsymbol{x_j}) \in E \\ 0 & otherwise \end{array} \right. \tag{1}$$

(2) K-nearest neighbors:

$$W_{ij} = \left\{ \begin{array}{ll} 1 & if \;\; \boldsymbol{x_i} \in \mathcal{N}(\boldsymbol{x_j}) \; or \; \boldsymbol{x_j} \in \mathcal{N}(\boldsymbol{x_i}) \\ 0 & otherwise \end{array} \right. \tag{2}$$

Where $\mathcal{N}(\boldsymbol{x})$ denotes the set comprising K nearest neighbors of $\boldsymbol{x}$. Let L be the graph Laplacian given by $L = D - W$ where W_{ij} is the edge weight in the data adjacency matrix defined in (1) or (2) and D is the diagonal matrix given by $D_{ii} = \sum_{j=1}^{l+u} W_{ij}$. One often use the normalized Laplacian defined by $L' = D^{-1/2} L D^{-1/2}$. $\boldsymbol{K}$ is the gram matrix defined by $\boldsymbol{K = \Phi^T(X)\Phi(X)}$ and $\boldsymbol{Y}$ is a $(l+u) \times l$ matrix given by $\boldsymbol{Y}_{ij} = y_i$ if $i = j$ and $\boldsymbol{x_i}$ is a labeled example, and $\boldsymbol{Y}_{ij} = 0$ otherwise.

II. Probabilistic labeled Semi-supervised SVM

A. Formulation

The proposed Probabilistic labeled Semi-supervised SVM (PLSVM) is based on the following principle:

$$\begin{aligned} f^* \;\; & = \arg \min_{W_p, \xi, F} \frac{1}{2} \|W_p\|_F^2 + \gamma \sum_{i=1}^{n} \sum_{p=1}^{C} \sum_{q=1}^{C} F_{ip} \xi_{ipq} \\ & + \lambda tr\left(F^T \tilde{L} F\right) + tr\left((F-Y)^T U \tilde{D} (F-Y)\right) \\ s.t. \;\; & \langle w_p - w_q, \phi(x_i) \rangle \geq 1 - \xi_{ipq}, i = 1, ..., n \\ & \xi_{ipq} \geq 0, p = 1, ..., C, q = 1, ..., C, p \neq q \\ & F_{ip} \geq 0 \\ & \sum_{p=1}^{C+1} F_{ip} = 1 \end{aligned} \tag{3}$$

Where W_p is the projection matrix. $F_{n \times (C+1)}$ is a probability matrix whose element F_{ip} represents the probability of x_i belonging to the p-th class. Let $\mathcal{X} = \{x_1, ..., x_l, x_{l+1}, ..., x_{l+u}\}$ be the training set, the first l data are labeled and the rest $(l+u)$ data are unlabeled. Assuming that there are C classes and let $\mathcal{C} = \{1, 2, ..., C, C+1\}$ be the class set. Note that the $(C+1)$-th class denotes the novel class which gives our method a mechanism to find the novel class. This is meaningful, because not every sample is meritorious. Some data maybe noise or not belong to labeled classes. $Y_{n \times (C+1)}$ is the label indicator matrix defined by $Y_{ip} = 1$ if $p = y_i$ and $Y_{ip} = 0$ if $p \neq y_i$ for labeled data x_i and $Y_{i(C+1)} = 1$ if x_i is unlabeled. $\tilde{L} = \tilde{D} - \tilde{W}$ is Laplacian matrix with normalized weights $\tilde{W}$ which is defined by $\tilde{W}_{ij} = W_{ij} / \sqrt{d_i d_j}$ and the normalized weight matrix can be written as $\tilde{W} = D^{-1/2} W D^{-1/2}$, where D is a diagonal matrix with entries $d_i = \sum_j W_{ij}$. $\tilde{D}$ is a diagonal matrix with entries $\tilde{d}_i = \sum_j \tilde{W}_{ij}$. U is a diagonal matrix with its i-th diagonal component μ_i. The third term $tr\left(F^T \tilde{L} F\right)$ is a regularization term, which measures the smoothness of the resulted labels on graph. The fourth term is a fitting term, which measures the differences between the resulted labels and the initial label assignments. The trade off between these two competing constraints is controlled by μ_i and $\tilde{d}_i$. Here $\mu_i > 0$ is a regularization parameter for the i-th data point x_i and $\tilde{d}_i = \sum_j \tilde{W}_{ij}$ is the degree of the i-th data point x_i. The decision function usually has a bias term b, one often deal with this term by appending each sample with an additional constant dimension:

$$\phi^T(\boldsymbol{x_i}) \leftarrow [\phi^T(\boldsymbol{x_i}), 1] \quad \boldsymbol{w_k^T} \leftarrow [\boldsymbol{w_k^T}, b_k] \tag{4}$$

Note that we use a more general nonlinear mapping ϕ in order to introduce the kernel technique.

Our motivation is intuitive and simple. Unlike traditional methods that the data belong to fixed classes, in our formulation, data are endowed to be able to belong to any class. We use F_{ip} to punish the relaxation variable ξ_{ipq} or the hinge loss $max(1 - \langle w_p - w_q, \phi(x_i) \rangle, 0)$ which means that the greater F_{ip} is, the decision value $w_p^T x_i$ which denotes the score of x_i labeled as the p-th class should be greater than the decision value $w_q^T x_i$ which is the score of x_i labeled as other classes. If x_i is noise or it doesn't belong to any class of interest, F_{ip} will be very small, thus the negative effects of noise or samples from novel class could be inhibited.

B. Solution to PLSVM

Because F and ξ are coupled mutually, it's hard to solve this problem directly. However, we can dissolve this whole formulation into two parts and optimize them respectively. Specifically, we can first obtain the optimal probability matrix F and then arrive at the optimal decision function. Besides, in order to solve the quadratic convex programming of PLSVM efficiently and reduce the memory requirement, we propose a kernel version dual coordinate descent algorithm and analyze the computation complexity and memory requirement of the proposed algorithm.

1) Solving the Probability Matrix: Ignoring the coupled term $\gamma \sum\limits_{i=1}^{n} \sum\limits_{p=1}^{C} \sum\limits_{q=1}^{C} F_{ip} \xi_{ipq}$, we first obtain the probability matrix by solving the following problem:

$$\min J\left(F\right) = tr\left(F^T \tilde{L} F\right) + tr\left(\left(F - Y\right)^T U \tilde{D} \left(F - Y\right)\right) \tag{5}$$

The optimal solution for the optimization problem can be easily solved by setting the derivative of $J(F)$ to zero:

$$\left.\frac{\partial J}{\partial F}\right|_{F = F^*} = 2\tilde{L}F^* + 2U\tilde{D}\left(F^* - Y\right) = 0$$

Let us introduce a set of variables,

$$\alpha_i = \frac{1}{\left(1 + \mu_i\right)}, i = 1, 2, ..., n$$

and let $P = \tilde{D}^{-1}\tilde{W}$,then the solution can be written as

$$\begin{aligned} F^* &= \left(\tilde{L} + U\tilde{D}\right)^{-1} U\tilde{D}Y \\ &= \left(I - P + U\right)^{-1} UY \\ &= \left(I - I_\alpha P\right)^{-1} I_\beta Y \end{aligned} \tag{6}$$

where I is an $n \times n$ identity matrix, I_α is an $n \times n$ diagonal matrix with the i-th entry being α_i, and $I_\beta = I - I_\alpha$. We use α_l to denote those α_is if x_i is labeled and α_u to denote the ones for unlabeled data. When $\alpha_l = 0$, that means that $\mu_l \to \infty$ and $F_{iy_i} = 1$, so x_i is fixed to the y_i-th class whereas when $0 < \alpha_l \leq 1$, x_i could walk to the other classes. Setting $\alpha_l > 0$ can correct the erroneously labeled data. Likewise, when setting $\alpha_u < 1$, we could find novel class. Now we'll show that the optimal F shown in Eq (6) satisfies the corresponding constraints in Eq (3):

$$\begin{aligned} & I_\alpha P 1_n + I_\beta Y 1_{C+1} = 1_n \\ \Rightarrow\ & I_\beta Y 1_{C+1} = \left(I - I_\alpha P\right) 1_n \\ \Rightarrow\ & \left(I - I_\alpha P\right)^{-1} I_\beta Y 1_{C+1} = 1_n \\ \Rightarrow\ & F^* 1_{C+1} = 1_n \end{aligned}$$

2) solving the optimal projection matrix: Given the probability matrix F in Eq (6), we can solve the following problem:

$$\begin{aligned} W_p^* \ &= \arg\min_{W_p, \xi} \frac{1}{2} \left\|W_p\right\|_F^2 + \gamma \sum_{i=1}^{n} \sum_{p=1}^{C} \sum_{q=1}^{C} F_{ip}\xi_{ipq} \\ s.t. \ & \left(w_p - w_q\right)^T \phi(x_i) \geq 1 - \xi_{ipq} \\ & \xi_{ipq} \geq 0, i = 1, ..., n; p, q = 1, ..., C; p \neq q \end{aligned} \tag{7}$$

The problem in Eq (7) is a linear inequality constrained quadratic convex optimization problem. Using the standard lagrange multiplier technique, we obtain the dual problem as follows:

$$\begin{aligned} \min \ & g(\boldsymbol{\alpha}) = \frac{1}{2} vec^T V_\alpha^T \bar{Q} vec V_\alpha^T - \sum_{i=1}^{n} \sum_{p,q} \alpha_{ipq} \\ s.t. \ & 0 \leq \alpha_{ipq} \leq \gamma F_{ip}, p \neq q \\ & \alpha_{ipq} = 0, p = q \end{aligned} \tag{8}$$

Where $\bar{\boldsymbol{Q}}$ is a semi-definite positive matrix given by

$$\bar{\boldsymbol{Q}} = (V_t \otimes \boldsymbol{\Phi}^T)(V_t^T \otimes \boldsymbol{\Phi}) = \left[V_t V_t^T\right] \otimes \left[\boldsymbol{K}\right] \tag{9}$$

and

$$\boldsymbol{V_t} = \left[\ vec\left(\boldsymbol{t}^1\right) \quad vec\left(\boldsymbol{t}^2\right) \quad \cdots \quad vec\left(\boldsymbol{t}^C\right)\ \right] \tag{10}$$

$$t_{ipq}^k = \begin{cases} 1 & \text{if } k = p, k \neq q \\ -1 & \text{if } k = q, k \neq p \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

$$\boldsymbol{V_\alpha} = \left[\ vec\left(\boldsymbol{\alpha}_1\right) \quad vec\left(\boldsymbol{\alpha}_2\right) \quad \cdots \quad vec\left(\boldsymbol{\alpha}_n\right)\ \right] \tag{12}$$

This is a linear inequality constrained quadratic convex problem. Once the variables α_{ipq} are solved, the expansion coefficient matrix $\boldsymbol{\beta} = \boldsymbol{V_\alpha^T V_t}$ could be calculated and the kth decision function $f_k(\boldsymbol{x})$ could be represented by the kernel functions$\{g_i(\boldsymbol{x}) = \mathcal{K}(\boldsymbol{x_i}, \boldsymbol{x})\}_{i=1}^{l+u}$ as follows:

$$f_k(\boldsymbol{x}) = \sum_{i=1}^{l+u} \beta_i^k \left(\mathcal{K}(\boldsymbol{x_i}, \boldsymbol{x}) + 1\right) \tag{13}$$

The label of sample $\boldsymbol{x}$ could be predicted by

$$y = \arg\max_k f_k(\boldsymbol{x}) \tag{14}$$

C. Connection to SVM

Traditional Multi-class SVM uses deterministic labels, whereas PLSVM uses probabilistic labels. If $I_\alpha = 0$ i.e. $\mu_i \to \infty, i = 1, ..., n$, all the labels of labeled data are deterministic and all the unlabeled data are predicted to be novel class samples. In this case PLSVM degenerates to traditional multi-class SVM [5] and of course, when $C = 2$, SVM is just a special case of PLSVM.

III. Dual Coordinate Descent algorithm for PLSVM

Coordinate descent algorithm is a popular optimization approach which has been broadly used in machine learning [6][7][8][9][10][11]. Lin et al [8] propose a dual coordinate descent method to solve the dual quadratic convex programming of SVM efficiently. The idea behind Dual coordinate descent algorithm is updating one component at a time by minimizing a single variable sub-problem. The updating procedure iterates continuously until arrives at a designated precision. If the subproblem can be efficiently solved, then it can be a competitive optimization algorithm. We see that the objective $g(\boldsymbol{\alpha})$ in Eq (8) is twice differentiable, and each variable has a simple bound constraint $0 \leq \alpha_{ipq} \leq \gamma F_{ip}$. We can update one variable at a time by minimizing a single-variable sub-problem. The kernel version dual coordinate descent algorithm for PLSVM is listed in Algorithm 1. Theorem 1 ensures the linear convergence of our algorithm (for lack of apace, we omit the proof).

Theorem 1: $\boldsymbol{\alpha}$ generated by Algorithm 1 globally converges to an optimal solution $\boldsymbol{\alpha}^*$. The convergence rate is

Algorithm 1 A kernel version dual coordinate descer method for multi-class UCSVM

```
Require: $\boldsymbol{K} = \Phi^{\boldsymbol{T}}(\boldsymbol{X})\Phi(\boldsymbol{X})$ and probability matrix $\boldsymbol{F}$
  Start with $\boldsymbol{\alpha} = \mathbf{0}$ and $\boldsymbol{\beta} = \mathbf{0}$
  while 1 do
    for all $i = 1, \cdots, n; p, q = 1, \cdots, C; p \neq q$ do
      1. $G = \boldsymbol{K}(i,:)(\boldsymbol{\beta}^p - \boldsymbol{\beta}^q) - 1$
      2. $PG = \begin{cases} G & \text{if } 0 < \alpha_{ipq} < \gamma F_{ip} \\ \min(0, G) & \text{if } \alpha_{ipq} = 0 \\ \max(0, G) & \text{if } \alpha_{ipq} = \gamma F_{ip} \end{cases}$
      3. If $|PG| \neq 0$,
         $\alpha^*_{ipq} \leftarrow \min\left(\max\left(\alpha_{ipq} - \frac{G}{2\boldsymbol{K}_{ii}}, 0\right), \gamma F_{ip}\right)$
         $\boldsymbol{\beta}^p_i \leftarrow \boldsymbol{\beta}^p_i + \left(\alpha^*_{ipq} - \alpha_{ipq}\right)$
         $\boldsymbol{\beta}^q_i \leftarrow \boldsymbol{\beta}^q_i - \left(\alpha^*_{ipq} - \alpha_{ipq}\right)$
    end for
    if $\|\boldsymbol{\alpha}^* - \boldsymbol{\alpha}\| < \epsilon$ then
      Break
    end if
    $\boldsymbol{\alpha} = \boldsymbol{\alpha}^*$
  end while
```

at least linear: there are $0 < \mu < 1$ and an iteration k_0 such that

$$g(\boldsymbol{\alpha}^{k+1}) - g(\boldsymbol{\alpha}^*) \leq \mu \left(g(\boldsymbol{\alpha}^k) - g(\boldsymbol{\alpha}^*)\right), \forall k \geq k_0 \quad (15)$$

IV. Computation Complexity and Memory Requirement

The training of PLSVM incorporates two steps. The first step is to calculate the probability matrix F. The second step is to solve the quadratic convex programming. The first step costs $O(n(C+1))$ and requires a memory of $O(n^2)$. The second step costs $O(n_{\boldsymbol{\alpha}} n)$, where $n_{\boldsymbol{\alpha}} = nC(C-1)$ is the number of lagrange multipliers in $\boldsymbol{\alpha}$. Thus the gross computation complexity is $O(n^2C(C-1))$. Note that in linear kernel, we only need to store the projection matrix and obtain the gradient by calculating the inner product of projection vector and training data, in this case PLSVM takes $O(n_{\boldsymbol{\alpha}}\bar{m})$ operations, where $\bar{m}$ is the number of the nonzero elements of training sample. As to the memory, our algorithm needs $O(n^2 + nC(C-1))$. By comparison, using traditional algorithm such as inner point method to directly solve the quadratic programming needs a memory of $O(n^2C^2(C-1)^2 + n^2)$ and a computation complexity of $O(n^2C^2(C-1)^2)$. When the number of classes are very large, our algorithm is more applicable than the traditional method.

V. Experiments

In this section, we will first present a toy problem to show the effectiveness of PLSVM under extreme labeling. Then, we'll compare PLSVM with several baselines including SVM, TSVM and LapSVM on several benchmark real-world datasets. For all experiments, we use 6-neighbor graph. All the parameters are the best setting through grid search. We use a desktop computer with a 2.40GHz Intel(R) Core(TM)2 Quad CPU and 3.25GB memory.

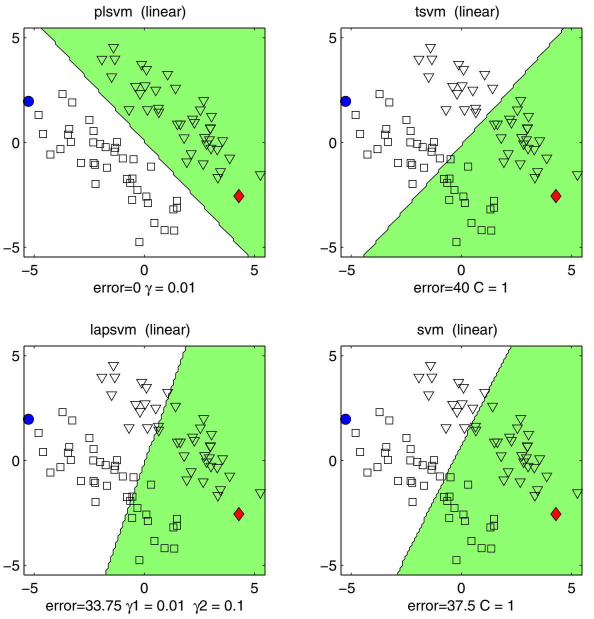

Figure 1. Two-Gaussian data. The top two figures show the best decision surfaces of PLSVM and TSVM, while the two bottom figures plot the best decision surfaces of LapSVM and SVM. Linear kernel is used.

A. Synthetic Data

In Figure 1, the unlabeled data are shown in black triangle and black square with each shape corresponding to a class. Only one sample is labeled for each class, shown in red diamond and blue circle respectively. We use gaussian kernel with $\sigma = 1$ to calculate the similarity matrix in Eq (1). We set $\alpha_u = 0.9999, \alpha_l = 0.0001, \gamma = 0.01$. From Figure 1, we can see that when labeling is not good enough, 1. LapSVM fails to find the ideal decision surface. This is because LapSVM just find a trade-off between the maximal margin of labeled data and the manifold smoothness. Although LapSVM introduces the manifold regularizer, it does not make full use of the unlabeled information. 2. TSVM is susceptible to local minima, and can not find the ideal decision function. 3. PLSVM presents an ideal decision surface because it can utilize more unlabeled information than LapSVM and TSVM by introducing probabilistic labels.

B. USPS: Digit Recognition

For USPS[1] dataset, 45 pairwise classification problems of handwritten digits are considered. The train set and test set are distributed as Table I shows. Two samples for each digit class are randomly labeled. We use the binary similarity matrix. Following the settings of [3], we use polynomial kernel of degree 3 to train the classifiers and set $C = 10$ for SVM and TSVM, and $\gamma_1 = 1/9, \gamma_2 = 200/9$ for LapSVM. For PLSVM, $r = 300, \alpha_l = 0.0001, \alpha_u = 0.9999$.

Table I
CLASS DISTRIBUTION IN USPS DATASET.

Digit	0	1	2	3	4
Train	1194	1005	731	658	652
Test	359	264	198	166	200
Digit	5	6	7	8	9
Train	556	664	645	542	644
Test	160	170	147	166	177

C. Umist: Face Recognition

For Umist face dataset[2], we resize each image to 28×23, normalize the gray scale to the interval of 0 to 1 and vectorize it as the feature vector. we choose the first 10 subjects in this experiment. For each subject, we select all images to build the train set with 2 of them randomly labeled(We don't build the test set because there are so few images for each subject). We consider 45 binary face recognition problems corresponding to 45 combinations for every two subjects. Error rates were averaged over 10 random choices of labeled examples with the linear kernel. Gaussian kernel is used with $\sigma = 1$ to calculate the similarity matrix. For SVM, we set $C = 0.06$. For transductive SVM, we set $C = 0.03$. For Laplacian SVM we set $\gamma_1 = 0.1, \gamma_2 = 0.1$. For PLSVM, $r = 300, \alpha_l = 0, \alpha_u = 0.9999$.

D. Coil20: Object Recognition

For Coil-20 dataset[3], we resize each image to 32×32, normalize the gray scale to the range of 0 and 1 and vectorize each image as its feature vector. Without loss of generality, we choose the first 10 objects in this experiment. For each object, we select all the images to construct the train set with 2 of them randomly labeled. We test our method on 45 binary object classification problems corresponding to 45 combinations for every two objects. We choose the linear kernel for the RKHS norm. We use gaussian kernel with $\sigma = 1$ to calculate the similarity matrix. For SVM, we set $C = 0.02$. For TSVM, we set $C = 0.03$. For LapSVM we set $\gamma_1 = 0.01, \gamma_2 = 0.01$. For PLSVM, $r = 300, \alpha_l = 0, \alpha_u = 0.9999$.

Table II
THE AVERAGE ERROR RATES(%)(STD%)

Method	USPS		20Newsgroup	
	Train	Test	Train	Test
svm	23.2(9.7)	21.6(12.7)	17.1(2.7)	17.5(3.0)
tsvm	15.6(5.4)	36.8(12.0)	**10.0(0.8)**	21.2(1.6)
lapsvm	7.4(6.3)	20.8(7.8)	13.5(2.8)	17.2(3.0)
plsvm	**2.4(2.2)**	**8.6(2.5)**	11.6(2.3)	**13.5(3.4)**
Method	Isolet		Umist	Coil20
	Train	Test	Train	Train
svm	20.1(0)	25.9(0)	37.2(5.4)	13.6(7.7)
tsvm	28.9(0)	31.0(0)	13.2(7.9)	8.5(4.8)
lapsvm	16.9(0)	24.0(0)	6.7(6.0)	6.0(5.9)
plsvm	**14.5(0)**	**21.8(0)**	**1.1(0.8)**	**0.2(0.3)**

Table III
THE AVERAGE TIME (SEC)(STD(SEC))

	svm	tsvm	lapsvm	plsvm
USPS	0.33(0.00)	0.33(0.00)	2.29(0.03)	45.4(1.32)
Umist	0.01(0.00)	0.001(0.00)	0.02(0.00)	0.60(0.42)
Coil20	0.01(0.00)	0.002(0.00)	0.06(0.00)	0.68(0.35)
20NG	0.17(0.02)	26.0(3.60)	6.56(0.03)	19.5(4.03)
Isolet	0.71(-)	28.1(-)	3.02(-)	38.1(-)

E. Isolet: Speech Recognition

Isolet dataset [12] contains the letters spoken by 150 subjects. We use the first 30 speakers(Isolet 1) for training and Isolet 5 for testing. We classify the first 13 letters of the English alphabet from the last 13. We consider 30 binary classification problems corresponding to 30 splits of the training data where all samples of one speaker are labeled and the rest are unlabeled. Following the settings of [3], we choose RBF kernel of width $\sigma = 10$. Binary similarity matrix is calculated. For SVM and TSVM, we set $C = 10$, for LapSVM, we set $\gamma_1 = 1, \gamma_2 = 200$. For PLSVM, we set $r = 200, \alpha_l = 0, \alpha_u = 0.6$.

F. 20Newsgroup: Text Categorization

For 20 Newsgroups dataset[4], we use the first five groups to construct our training and testing set. Within each group, we choose the first 630 documents as the training samples and 10 of them were randomly labeled and the rest documents as the test sets. We take normalized $tfidf$ [4] as the feature vector. We consider 45 binary text categorization problems corresponding to 45 combinations for each two kind of news. Error rates were averaged over 10 random choices of labeled examples. We calculate the similarity matrix by inner product and use the linear kernel. For standard SVM, we set $C = 1.2$. For TSVM, we set $C = 0.1$. For LapSVM we set $\gamma_1 = 1, \gamma_2 = 10$. For PLSVM, we set $r = 400, \alpha_l = 0, \alpha_u = 0.9$.

G. Performance and Speed Analysis

The average error rates for the five real-world datasets are shown in Table II. From the results, we can conclude that PLSVM overwhelms all the other three baselines on digit

[1] http://www-stat-class.stanford.edu/~tibs/ElemStatLearn/index.html

[2] http://www.cs.toronto.edu/~roweis/data.html

[3] http://www1.cs.columbia.edu/CAVE/software/softlib/coil-20.php

[4] http://people.csail.mit.edu/jrennie/20Newsgroups/

Table IV
THE AVERAGE ERROR RATES(%)(STD%)

USPS	mcsvm	lapsvm	laprls	plsvm
Train	32.5(7.8)	**8.1(6.8)**	**8.1(6.8)**	9.5(4.0)
Test	36.9(5.8)	32.4(6.1)	32.3(6.0)	**15.7(3.9)**
Umist	mcsvm	lapsvm	laprls	plsvm
Train	28.4(2.2)	8.5(1.6)	8.8(1.8)	**7.9(2.1)**
Test	56.9(6.5)	48.4(4.2)	47.1(4.3)	**45.1(3.6)**

recognition, face recognition, object recognition, speech recognition and text categorization. Although PLSVM performs a little worse than TSVM on training set, it performs far better than TSVM on the test set. As a whole, PLSVM has better generalization ability and is less sensitive to labeling (smaller standard deviation).

Table III presents the average times of PLSVM and the other three baselines. For SVM and LapSVM, we use the matlab quadprog function to solve the QP problem. For TSVM, we use the SVM-light software. From Table III, we can see that PLSVM needs more computation than the other three baselines. This is because PLSVM constrains all the training samples and has more unknown variables than SVM and LapSVM. However, the proposed algorithm requires less memory which makes PLSVM more applicable to real applications especially when the number of classes are very large.

H. Multi-class Performance

PLSVM can deal with multi-class classification problems directly. Here we compare PLSVM with several baselines including multi-class SVM (MCSVM), LapSVM using one versus rest scheme(LapSVM) Laplacian least squares regression (LapRLS) on USPS $(2, 3, 5, 8)$ and Umist datasets. For USPS dataset, we choose four digits $(2, 3, 5, 8)$, for each digit, two samples are randomly labeled. For Umist dataset, we choose all the 20 persons and use the first $\%80$ samples for each subject to build the training sets with the left $\%20$ samples for testing. For each person, two sample are randomly labeled. The experiment results are shown in Table IV. We can conclude that PLSVM has better generality ability and is less sensitive to labeling (smaller standard deviation).

VI. CONCLUSIONS

In this paper, we propose a novel semi-supervised method called probabilistic labeled Semi-supervised SVM (PLSVM). Unlike traditional SVM and LapSVM which utilize the deterministic label of each labeled data, PLSVM makes use of the probabilistic labels of all the training data including the labeled and unlabeled data. Experiments on both one synthetic data and five real world benchmark datasets strongly validate our motivation. By introducing the mechanism of probabilistic label, PLSVM can extract more geometrical structure information of unlabeled data, be more robust against bad labeling, weaken the negative effects of noise and learn a better classifier, although it needs more computation.

ACKNOWLEDGMENT

This work was supported by NSFC (Grant No. 60675009).

REFERENCES

[1] V. Vapnik, *Statistical Learning Theory*. Wiley-Interscience, 1998.

[2] O. Chapelle, B.Scholkopf, and A. Zien, *Semi-Supervised Learning*. Cambridge, Massachusetts: MIT Press, 2006.

[3] P. N. M. Belkin and V. Sindhwani, “Manifold regularization: A geometric framework for learning from labeled and unlabeled examples,” in *Journal of Machine Learning Research*, 2006, pp. 2399–2434.

[4] T. Joachims, “Transductive inference for text classification using support vector machines,” in *Proceedings of the 16th International Conference on Machine Learning*, 1999, pp. 200–209.

[5] J. Weston and C. Watkins, “Multi-class support vector machines (Technical Report CSD-TR-98-04),” *Department of Computer Science, Royal Holloway, University of London*, 1998.

[6] Z. Luo and P. Tseng, “On the convergence of the coordinate descent method for convex differentiable minimization,” *Journal of Optimization Theory and Applications*, vol. 72, no. 1, pp. 7–35, 1992.

[7] A. Bordes, L. Bottou, P. Gallinari, and J. Weston, “Solving multiclass support vector machines with LaRank,” in *Proceedings of the 24th international conference on Machine learning*. ACM New York, NY, USA, 2007, pp. 89–96.

[8] C. Hsieh, K. Chang, C. Lin, 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*. ACM New York, NY, USA, 2008, pp. 408–415.

[9] S. Keerthi, S. Sundararajan, K. Chang, C. Hsieh, and C. Lin, “A sequential dual method for large scale multi-class linear SVMs,” 2008.

[10] G. Chen, J. Zhang, F. Wang, C. Zhang, and Y. Gao, “Efficient Multi-label Classification with Hypergraph Regularization,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition(CVPR)*, 2009.

[11] M. Qian, F. Nie, and C. Zhang, “Efficient Multi-class Unlabeled Constrained Semi-supervised SVM,” in *Proceedings of The 18th ACM Conference on Information and Knowledge Management*. ACM New York, NY, USA, 2009.

[12] A. Asuncion and D. Newman, “UCI machine learning repository,” 2007. [Online]. Available: http://www.ics.uci.edu/∼mlearn/MLRepository.html