2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

A Transductive Support Vector Machine with Adjustable Quasi-Linear Kernel for Semi-Supervised Data Classification

Bo ZHOU, Chenlong HU, Benhui CHEN and Jinglu HU

***Abstract*—This paper focuses on semi-supervised classification problem by using Transductive Support Vector Machine. Traditional TSVM for semi-supervised classification firstly train an SVM model with labeled data. Then use the model to predict unlabeled data and optimize unlabeled data prediction to retrain the SVM. TSVM always uses a predefined kernel and fixed parameters during the optimization procedure and they also suffers potential over-fitting problem. In this paper we introduce proposed quasi-linear kernel to the TSVM. An SVM with quasi-linear kernel realizes an approximate nonlinear separation boundary by multi-local linear boundaries with interpolation. By applying quasi-linear kernel to semi-supervised classification it can avoid potential over-fitting and provide more accurate unlabeled data prediction. After unlabeled data prediction optimization, the quasi-linear kernel can be further adjusted considering the potential boundary data distribution as prior knowledge. We also introduce a minimal set method for optimizing unlabeled data prediction. The minimal set method follows the clustering assumption of semi-supervised learning. The pairwise label switching is allowed between minimal sets. It can speed up optimization procedure and reduce influence from label constrain in TSVM. Experiment results on benchmark gene datasets show that the proposed method is effective and improves classification performances.**

I. Introduction

SEMI-SUPERVISED classification is a special form of classification task [1] [2]. Traditional classifiers use only labeled data to train. Considering that labeled data sometimes difficulty and expensive to obtain and meanwhile unlabeled data are easy to collect and there are few ways to use them. The purpose of semi-supervised classification is to use both labeled and unlabeled data to train a classifier, aim to improve the generalization ability. Under certain assumptions, unlabeled data can be used for training a semi-supervised classifier. One classical assumption is that the separation boundary should not across high density regions, many methods has been developed such like information regularization, Gaussian process, Graphic kernels and Transductive Support Vector Machine(SVM).

SVMs implement Structural Risk Minimization (SRM) principle by maximizing a margin to minimize an upper bound of the generalization error rather than minimize the training error, which result in good generalization performance, the absence of local minima and the sparse representation of solution [3]. The Transductive SVM [4], [3] can be seen as an extended solution of SVM for semi-supervised problems. The goal of Transductive SVM is to find a labeling of the unlabeled data and a separation boundary has the maximum margin on both the original labeled data and unlabeled data.

In a Transductive SVM, an SVM classifier is firstly trained from the labeled data and applied to unlabeled data. After all unlabeled instances are given a predicted label, a hinge loss function [2] is used on these unlabeled data for evaluate and optimize the SVM classifier:

$$\min \frac{1}{2}w^2 + C\sum_{i=1}^{n}\xi_i + C^*\sum_{i=n+1}^{n+m}\xi_i, \tag{1}$$

under the constraints:

$$\begin{cases} (w\cdot x_i + b) \geq 1-\xi_i & 1 \leq i \leq n \\ |w\cdot x_i + b| \geq 1-\xi_i & n+1 \leq i \leq n+m \end{cases} \tag{2}$$

where the first two phase evaluate the SVM and the third phase evaluates the prediction of unlabeled data. Many method having been introduced to optimizing TSVM. Chapelle et al. [5] [6] proposed ∇SVM using a Graphic kernel to train SVM model and a Gaussian function to approximates the hat loss function, the graphic kernel can be seen as a special RBF kernel with density similarity. Collobert et.al [7] optimize the hard TSVM directly using an concave-convex optimization procedure(CCCP). Sindhwani and Keerthi [8] proposed a fast algorithm for linear TSVM, which is suitable for large scale applications. And with the branch and bound search Chapelle et al. [9] finds a global optimal solution for small datasets.

The Transductive SVM methods use the powerful regularization of SVMs and give many promising benefits, it will also suffer potential problems like traditional SVM. One problem is over-fitting when using SVM model with some non-linear kernels like RBF kernel. Especially in TSVM, the kernel and its parameter are always predefined and treated like a black box tool in optimization process, the potential over-fitting can leads incorrect prediction of unlabeled data and harmful to optimizing the final SVM classifiers. Another thought is TSVM predict the unlabeled data under constrain

Bo ZHOU, Chenlong HU and Jinglu HU are with Graduate school of Information, Production and Systems, Waseda University of Hibikino 2-7, Wakamatsu-Ku, Kitakyushu-shi, Fukuoka, Japan. Email: zhoubo_bird@toki.waseda.jp; huchenlong@ruri.waseda.jp jinglu@waseda.jp. .

Benhui CHEN is with the School of Mathematics and Computer Science. Dali University. Dali, Yunnan Province, China. email: bhchen@dali.edu.cn.

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

that the label ratio should be same with labeled data. The fixed ratio constrain will misguide the unlabeled prediction because it is hard to estimate the ratio between positive and negative from unlabeled data.

In our previous research [10] [11] [12], we proposed a quasi-linear SVM to solve the potential over-fitting problem in SVM. A quasi-linear SVM implements a nonlinear separation boundary by aggregating multi-local linear separation boundaries with an interpolation function. Unlike other multi-local linear models, in quasi-linear SVM we use a composite kernel (quasi-linear kernel) to represent the local linear information and the training of quasi-linear SVM is like a standard single SVM method. According to [13], the quasi linear kernel is flexible and adjustable. In this paper, we aim to apply the quasi-linear kernel to semi-supervise classification. Consider the characteristic of quasi-linear kernel, the kernel of SVM model can also be optimized during the optimization of TSVM. Unlike traditional TSVM predefined a fixed kernel; we re-estimate the potential boundary distributed area and then using this information to build a quasi-linear kernel in each iteration of TSVM optimization. We also propose a 1-NN method to optimize unlabeled data prediction which allows the labeling constrain in TSVM can be changed under a certain level.

The rest parts of the paper are organized as follows: Section II describes the conception of quasi linear kernel and how to implement quasi linear SVM. Section III proposes how the Transductive SVM with quasi-linear kernel is optimized. Section IV presents the simulation results of proposed method and finally the conclusions are discussed in Section V.

II. Quasi-inear SVM

Local linear SVM strategy approximates a nonlinear separating boundary by estimating a series of piecewise linear boundaries. In this paper an SVM with quasi linear kernel is proposed to implement the piecewise linear approximation by using a kernel composition technique and aggregating local linear boundaries in kernel level.

A. SVM with Quasi-linear Kernel

Suppose we have the following labeled training data points of N samples $(x_1, y_1), \ldots, (x_i, y_i), \ldots, (x_k, y_k)$, $x_i \in R^d$ is the input vector corresponding to the ith sample labeled by $y_i \in \{-1, +1\}$ depending on its class. A nonlinear separating boundary $f_p(x)$ can which known as a priori knowledge be approximated by M local linear boundaries with interpolation as showed as:

$$f_p(x) = \sum_{j=1}^{M} (\Omega_j^T x + b_j) R_j(x) + b \tag{3}$$

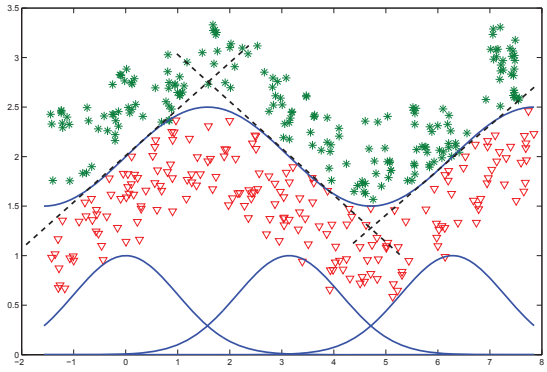


Fig. 1. Multiple local linear models with interpolation for nonlinear separation hyperplane

where $R_j(x)$'s are the radius basis function for interpolation, Ω_j's are the coordinate parameter vectors of local linear boundaries. Introducing two vectors $\Phi(x)$ and Θ defined by

$$\Phi(x) = \left[R_1(x), x^T R_1(x), \ldots, R_M(x), x^T R_M(x)\right]^T \tag{4}$$

$$\Theta = \left[b_1, \Omega_1^T, \ldots, b_M, \Omega_M^T\right]^T \tag{5}$$

We further express (3) as

$$f_p(x) = \Theta^T \Phi(x) + b \tag{6}$$

Introducing the structural risk minimization principle into the (6), the classification problem can be described as the QP optimization problem as following:

$$\begin{aligned} &\arg\min_{\Theta, b, \xi} J_p = \frac{1}{2}\Theta^T\Theta + C\sum_{k=1}^{N}\xi_i \\ &\text{s.t.} \begin{cases} y_k[\Theta^T\Phi(x_k) + b] \geq 1 - \xi_k, \forall k \\ \xi_k \geq 0, \forall k \end{cases} \end{aligned} \tag{7}$$

The Lagrange function has been constructed, via introducing new variables (α_k, v_k) called Lagrange multipliers:

$$\begin{aligned} &L(\Theta, b.\xi; \alpha, v) = J_p(\Theta, \xi) \\ &- \sum_{k=1}^{n}(\alpha_k y_k[\Theta^T\Phi(x_k) + b] - 1 + \xi_k) - \sum_{k=1}^{N} v_k \xi_k \end{aligned} \tag{8}$$

with Lagrange multipliers $\alpha_k \geq 0, v_k \geq 0, \forall k$. The solution is given by the saddle point of the Lagrange function:

$$\max_{\alpha, v} \min_{\Theta, b, \xi} L(_\Theta, b, \xi; \alpha, v) \tag{9}$$

which leads to

$$\begin{cases} \dfrac{\partial L}{\partial \Theta} = 0 \rightarrow \Theta = \sum_{k=1}^{N} \alpha_k y_k \Theta(x_k) \\ \dfrac{\partial L}{\partial b} = 0 \rightarrow \sum_{k=1}^{N} \alpha_k y_k = 0 \\ \dfrac{\partial L}{\partial \xi} = 0 \rightarrow 0 \leq \alpha_k \leq c, \forall k \end{cases} \tag{10}$$

The dual problem becomes as follow:

$$\max_{\alpha} J_d = -\frac{1}{2} \sum_{k,l=1}^{N} \Phi^T(x_i)\Phi(x_j)\alpha_i\alpha_j + \sum_{k=1}^{N} \alpha_i$$
$$\text{s.t.} \begin{cases} \sum_{k=1}^{N} y_k \alpha_k = 0 \\ 0 \leq \alpha_k \leq C, \forall k \end{cases} \tag{11}$$

In the quadratic form, the kernel trick is applied

$$\begin{aligned} K(x_k, x_l) =& \Phi(x_k)^T \Phi(x_k) \\ =& (1 + x_k^T x_l) \sum_{j=1}^{M} R_j(x_k) R_j(x_l), \forall k \end{aligned} \tag{12}$$

Hence, the nonlinear separating boundary model $f_p(x)$ is reduced to a standard SVM based on a composite kernel (Eq.12). Finally the nonlinear SVM classifier takes the following form:

$$y = \mathbf{sign}[\sum_{k=1}^{N} \alpha_k y_k K(x, x_k) + b] \tag{13}$$

According to the description in Eq.12, a quasi-linear kernel is a flexible model and turns in a form of inner product of an explicit nonlinear mapping. The complexity of the quasi-linear kernel can fill the gap between linear and nonlinear kernel functions by adjusting the value of M. When $M = 1$, it is a linear kernel, when M becomes large, it is close to a general nonlinear kernel. Considering the adjustable characteristic of quasi-linear kernel, it can be applied in a TSVM optimized procedure.

B. Implementation of Quasi-linear SVM

According to the formulation of quasi-linear kernel above, an implementation of quasi-linear SVM follow this procedure:

1. Detecting data near to the separation boundary and constructing a border data set. The definition of whether a point belongs to border data set is: for a point x, if its k-nearest neighbour contains points having different labels it will be border data [14].

2. Using clustering method to partition the border data set and decide M in quasi-linear kernel.

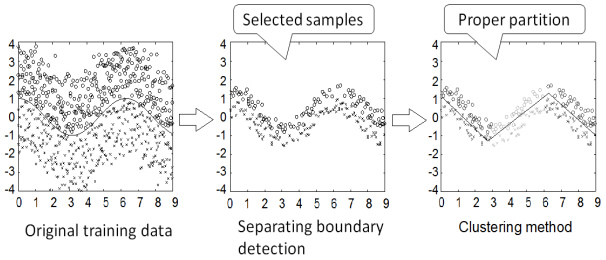


Fig. 2. Guided partition method to partition the input space along the separating boundary

3. Using center and radius of each partition to construct quasi-linear kernel.

4. Training quasi-linear SVM with the quasi-linear kernel.

Fig.2 shows how the step 2 and 3 works for constructing a quasi-linear kernel.

III. TSVM WITH QUASI-LINEAR KERNEL

In this part we introduce the quasi-linear kernel to a Transductive SVM. A TSVM with quasi-linear kernel follows under procedure:

1. Train an SVM with labeled data.
2. Predict unlabeled data with trained SVM model.
3. Evaluate unlabeled data prediction with trained SVM.
4. Minimize loss function with a pairwise label switching and updated unlabeled data prediction.
5. Re-estimate the potential border dataset and adjust quasi-linear kernel.
6. Using the labeled data and predicted label to train a new SVM.
7. Redo step 3 to step 6.

Unlike traditional TSVM predefines kernel and parameters, in our model, we firstly use a modified pairwise label switching method to optimize unlabel data prediction, then we re-estimate potential border dataset and adjust quasi-linear kernel in each iteration optimization of TSVM.

A. Optimization of Unlabeled Data Prediction

In [15], SVMlight, a famous TSVM model, is proposed by introducing a pairwise label switching between the unlabeled data prediction to optimized TSVM model. The loss function is defined by:

$$\begin{aligned} &\min_{f} \sum_{i=1}^{l} (1 - y_i f(x_i))_+ + \lambda_1 ||w||^2 \\ &+\lambda_2 \sum_{i=l+1}^{n} (1 - |f(x_i)|)_+, \\ &\text{s.t.} \quad \tfrac{1}{n-l} \sum_{i=l+1}^{n} f(x_i) = \tfrac{1}{l} \sum_{i=1}^{l} y_i \end{aligned} \tag{14}$$

And a pairwise label switching is happened if:

$$\begin{aligned} &loss(y_i = 1, f(x_i)) + loss(y_j = -1, f(x_j)), \\ &> loss(y_i = -1, f(x_i)) + loss(y_j = 1, f(x_j)) \end{aligned} \tag{15}$$

The label constrain is defined that the unlabeled data prediction should follow ratio between two classes distributions. It can avoid all unlabeled data prediction falls to one class. In SVMlight the pairwise label switching can guarantee the label constrain while in our thoughts a fixed label ratio sometimes can give a misguidance to unlabeled data prediction. Fig.3(a) shows a margin obtained under a label constarin.In this dataset the given positive and negtive label ratio is 1:1, while in the unlabeled data more points are surround the positive labels. So using a hard label constrain, there is a bias on the margin and the margin is straitness. Thus,we propose a minimal set for pairwise label switching.

In this method the training set is firstly been divided to many small subsets, we use the conception of minimal spanning tree and each data is connected to its nearest neighbor (like 1-NN clustering). Secondly, for some isolated point, their connection to its nearest neighbor is cut. The judgment of whether a point is isolated or not, is based on whether its distance to nearest neighbor lager than 1.5 times of average distance over the remaining points in that cluster. Then the minimal set is obtained and we defines that all unlabeled points in same minimal set shares the same label prediction. And the pairwise label switching is applied between the minimal set. Fig.3(b) shows the minimal set division of the given unlabeled dataset,and Fig.3(c) shows the margin obtained under the label constrain on minimal set. The ratio of the unlabeled predition on set is 2:3 while the real prediction ratio on data is 3:7. The obtained margin is more wide and has a better generallization abillity.

Here is some comment about introducing minimal set:

1. It follows the cluster assumption about semi-supervised classification: a separation boundary should not through the high density area. In high density region each point will have same label with its nearest neighbor and in low density area, point will likely be isolated.

2. The number of individuals in each minimal set differs. The label constrain now works on minimal sets instead individuals. So label switching between sets can make the label constrain more flexible.

3. The minimal set can be seen as one individual point in the prediction optimization. It can speed up the pairwise label switching because the number of individuals is reduced.

B. Quasi-linear Kernel Composition in TSVM

In this part the TSVM with quasi-linear kernel is introduced. As mentioned before, two steps of constructing a quasi-linear kernel are border dataset detection and partition the border dataset. These two steps in TSVM are little different than SVM.

1) Border dataset detection: Considering the i-th iteration of TSVM, all minimal sets contains support vectosr (points located in and on the margin) of i-1-th iteration will belongs to border dataset automatically. Then during the optimization procedure, if the label of a minimal set has been switched, we detected its k-nearest neighbours minimal set, the neighbourhood of minimal set is detected by single linkage method [16]. If one of its neighbour set located outside the margin and has an opposite label (after changed), it indicates that the neighbour set will likely contains support vectors of i-th iteration SVM, the neighbour set will be add to border dataset.

2) Border dataset partition: Since we introduced minimal set based on minimal spanning tree and 1-NN before, a minimal spanning tree(MST clustering) method is proposed to partition the border dataset. In MST, a data space with N samples can be considered as a connected, undirected, weighted graph. The MST connects all these samples with N -1 edges with the less weighted. Conventional MST clustering method use top-down strategy by by first constructing the MST structure of data space and then disconnecting the most weighted MST edges to create subspaces. Each subspace will represent a cluster [17], [18]. Conventional top-down MST clustering method always produced small partition while in quasi-linear kernel, partitions with a mount of population is preferred. The bottom-up MST merging is introduced based on Kruskals algorithm [19].

In bottom-up MST merging us firstly considers population of each partition should maintain a certain level. Then when merge two partitions both satisfied basic population need, a merge criterion based on the ratio of geographical distance and real distance is applied:

According to MST algorithm we can define nearest link point P_{near} and farthest link point P_{far} between two partitions $P1$ and $P2$. Thus 2 distances are calculated

$$\begin{aligned} D_{geo} &= P1_{near}P1_{far} + P2_{near}P2_{far} + P1_{near}P2_{near} \\ D_{real} &= P1_{far}P2_{far} \end{aligned} \tag{16}$$

by a predifined threshold θ, if

$$D_{geo}/D_{real} \leq \theta \tag{17}$$

P1 and P2 can merge to a bigger partition.

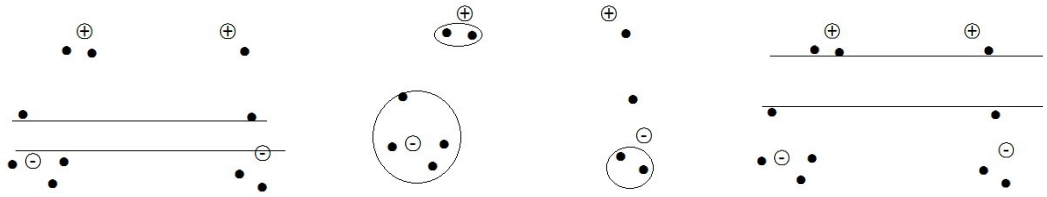
With the merge criterion we can automatically detect local partition numbers. In our experiments we detect 3 nearest neighbor of a partition and merge it to its neighbor with the less available values, the available threshold is set to 2.

An algorithm of proposed TSVM with quasi-linear kernel is given by Algorithm.1.

IV. EXPERIMENTS AND RESULTS

A. Evaluation Metrics

Three classical evaluation metrics of Precision, Recall and F-score are used to evaluate the efficiency of the proposed



(a) Margin with a label constrain on instances

(b) Minimal set detection

(c) Margin with label constrain on sets

Fig. 3. The margin with label constrain comparison on instance and minimal set

Algorithm 1 TSVM with quasi-linear kernel

Input: labeled data X_L;unlabeled data X_U
Output: Transductive SVM classifier TS
Train a quasi-linear SVM L with X_L;
Divide minimal set $A = A_1, A_2, \ldots, A_N$;
let C^* = $2^{-10}C$
repeat
Predict label of X_U with L under label constrain $\frac{1}{u}\sum_{i=1}^{U} f(x_{ui}) = \frac{1}{l}\sum_{i=1}^{L} y_{li}$;
Label switching within minimal set and let all data point shares same label prediction;
Detected border dataset $S = S_1, S_2, \ldots, S_M$, $S \subset A$;
for $i = 1$; $i \leq M$; $i++$ **do**
if $loss(Y_i = 1, f(S_i)) + loss(Y_j = -1, f(S_j)) > loss(Y_i = -1, f(S_i)) + loss(Y_j = 1, f(S_j))$ **then**
Switch label of S_i and S_j and Detect 3-nearest neighbor $A_{neighbor}$ of S_i;
if $Y_{neighbor} \neq Y_i$ and $A_{neighbor} \notin S$ **then**
Add $A_{neighbor}$ to border dataset S.
end if
end if
end for
Bottom up merge partitions with MST method.
Use the partition information to construct a new quasi-linear kernel
Use the new kernel to train a new SVM and increasing C^*
until C^*=C
return TS;

method. Precision, Recall and F-score are defined for an imbalanced binary classification task with positive and negative classes. Precision is the proportion of positive predictions that are correct, and recall is the proportion of positive samples that are correctly predicted positive. That is:

$$\begin{aligned} precision &= \frac{TP}{TP+FP} \\ recall &= \frac{TP}{TP+FN} \\ F-score &= \frac{2 \star precision \star recall}{precision + recall} \end{aligned} \tag{18}$$

with TP the number of true positives (correctly predicted positive samples), FP the number of false positives (positive predictions that are incorrect), and FN the number of false negatives (positive samples that are incorrectly predicted negative).

B. Dataset Description

In this papera yeast gene data is used to evaluate proposed TSVM with quasi-linear kernel in our experiments. The yeast gene data classification are nonlinear problems with characteristics of high noise and large number of input features compared with the relatively small number of training examples, the conventional nonlinear kernel SVM models are severely over-fitting at times.

The yeast gene data is always a multi-label task [20], [21]. Since we want to improve the basis classifier performance for each label (class), so we only implement per-label (binary classification) experiments for evaluating, several label has been selected for our experiment.

TABLE II
COMPARISON RESULTS ON LABELED DATA

dataID	SVM with density RBF kernel			SVM with linear kernel			the quasi-linear SVM		
	precision	recall	F-score	precision	recall	F-score	precision	recall	F-score
14.01	0.3592	0.0175	0.0335	0.3378	0.3229	0.3301	0.2375	0.4603	0.3133
14.04	0.6813	0.0946	0.1661	0.3781	0.3382	0.3570	0.4424	0.5226	0.4792
14.07	0.8189	0.0784	0.1431	0.6043	0.5526	0.5773	0.6317	0.7805	0.6982
14.10	0.5903	0.0872	0.1519	0.3637	0.4222	0.3907	0.3921	0.4637	0.4249
14.13	0.5048	0.0115	0.0225	0.3077	0.4122	0.3523	0.3637	0.5474	0.4370

TABLE III
COMPARISON RESULTS ON LABELED AND UNLABELED DATA 1:1

dataID	SVM with density RBF kernel			SVM with linear kernel			the quasi-linear SVM		
	precision	recall	F-score	precision	recall	F-score	precision	recall	F-score
14.01	0.3193	0.0344	0.0621	0.2037	0.3426	0.2757	0.2375	0.4603	0.3133
14.04	0.5742	0.1176	0.1952	0.2974	0.3517	0.3223	0.4089	0.5317	0.4622
14.07	0.5585	0.0842	0.1463	0.5515	0.5138	0.5319	0.7019	0.7313	0.7162
14.10	0.3193	0.1321	0.1869	0.2313	0.4505	0.3057	0.4017	0.4463	0.4228
14.13	0.5215	0.0326	0.0614	0.2275	0.5474	0.3204	0.3492	0.5186	0.4173

TABLE IV
COMPARISON RESULTS ON LABELED AND UNLABELED DATA 2:5

dataID	SVM with density RBF kernel			SVM with linear kernel			the quasi-linear SVM		
	precision	recall	F-score	precision	recall	F-score	precision	recall	F-score
14.01	Nan	Nan	Nan	0.3103	0.2594	0.2825	0.2218	0.4417	0.2951
14.04	0.5413	0.1869	0.2777	0.3177	0.3456	0.3310	0.4011	0.4893	0.4408
14.07	0.6714	0.0613	0.1123	0.5816	0.5914	0.5864	0.6835	0.7209	0.7017
14.10	Nan	Nan	Nan	0.2870	0.4121	0.3381	0.3719	0.4811	0.4195
14.13	0.5338	0.0257	0.0494	0.2436	0.4545	0.3171	0.3318	0.4862	0.3938

TABLE I
YEAST DATASET DISCRIPTION

dataID	Data discription
14.01	Protein folding and stabilization
14.04	Protein targeting, sorting and translocation
14.07	Protein modification
14.10	Assembly for protein complexes
14.13	Protein/peptide degradation

C. Simulation Results

In our experiment, we use a libSVM toolbox for training our SVM model. The data is split to 625 training samples and 328 testing samples. Three experiments are made to compare the proposed TSVM model. In first experiment we only consider labeled data just like a conventional SVM. In the next two experiments, both labeled and unlabeled data are used to evaluate the performance of TSVM with different kernels,a RBF kernel with density measures [5], linear kernel and quaisi-linear kernel. In experiment II ratio of labeled and unlabeled data is about 1:1 while in experiment III the ratio is 2:5.

Table II shows classification result using SVM with 3 different kernels on all labeled dataset. It shows that the quasi-linear kernel performs better than other two kernels on the yeast dataset. We use this result as basis result and compare it to TSVM classification.

In Table III the labeled and unlabeled data ratio is 1:1, from the table we can see that the TSVM with denstity RBF performs better than SVM training with 625 labeled data, but it still suffers over-fitting. And the performance of TSVM with linear kernel is reduced compared to SVM with linear kernel. The proposed TSVM with quasi-linear kernel gives an equivalent performance like SVM, and improves classification results on 14.07 dataset, even with less training samples.

When the labeled and unlabeled data ratio deduced to 2:5, Table IVthe TSVM with density RBF cannot give a meaningful value on 14.01 and 14.10 dataset. The TSVM with linear kernel continuously reduces while quasi-linear kernels are still stable. TSVM with quasi-linear kernel still has advantage than other two kernels.

These expertiments demonstrate the TSVM with quasi-linear kernel has advantage on classification of both supervised and semi-supervised cases. For the classification of dataset with characteristics of high noise and large number of input features compared with the relatively small number of training examples, using the quasi-linear kernel can provide more accurate unlabeled data prediction compared to a RBF based kernel. The adjusable characteristic of TSVM with quasi-linear kernel can make a kernel more suitable to a

labeled and unlabeled cases.

V. CONCLUSION

In this paper we introduced a quasi-linear SVM to a semi-supervised classification problem. Unlike traditional Transductive SVM using fixed kernel and parameters, we proposed an adjustable quasi-linear kernel and optimize it with the unlabeled data prediction information during the TSVM optimization procedure. The adjustable quasi linear kernel is a nonlinear kernel which realizes multi local linear separations. It uses data distribution information as prior knowledge and has better generalization ability than traditional RBF kernels. In semi-supervised problem, the kernel can be adjusted in optimization procedure and can improve classification accuracy of TSVM. The simulation results demonstrate that a quasi-linear kernel can outperform than other nonlinear kernel in the semi-supervised task on yeast gene classification.

We also introduced a minimal set to the TSVM optimization procedure, the idea is based on minimal spanning tree and 1-nn method. Introducing minimal set to a TSVM optimization (SVMlight optimization) can speed up the optimization speed and adjust the label constrain of TSVM under a certain level, which can further improves generalization ability of TSVM. In the future we want to further develop our work from two aspects: enlarge the minimal set in some high region area to improve the efficiency of our method; considering an incremental case where the unlabeled data adds to the training set incrementally, develop a more efficient method than the minimal spanning tree based on structure.

REFERENCES

[1] O. Chapelle, B. Schölkopf, A. Zien *et al.*, *Semi-supervised learning*. MIT press Cambridge, 2006, vol. 2.

[2] X. Zhu, "Semi-supervised learning literature survey," *Computer Science, University of Wisconsin-Madison*, vol. 2, p. 3, 2006.

[3] V. Vapnik, *The Nature of Statistical Learning Theory, 2nd Ed.* Springer-Verlag, 1999.

[4] K. Bennett, A. Demiriz *et al.*, "Semi-supervised support vector machines," *Advances in Neural Information processing systems*, pp. 368–374, 1999.

[5] O. Chapelle and A. Zien, "Semi-supervised classification by low density separation," in *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (AISTAT).*, 2005.

[6] C. M. Chapelle, O. and A. Zien, "A continuation method for semisupervised svms," in *23rd International Conference on Machine Learning(ICML06).Pittsburgh, USA.*, 2006.

[7] W. J. . B. L. Collobert, R., "Trading convexity for scalability."

[8] . K. S. S. Sindhwani, V., "Large scale semisupervised linear svms." in *SIGIR 2006.*

[9] S. V. . K. S. S. Chapelle, O., "Branch and bound for semisupervised support vector machines," in *Advances in Neural Information Processing Systems (NIPS)..*, 2006.

[10] B. Chen, F. Sun, and J. Hu, "Local linear multi-SVM method for gene function classification," in *Proc of the World Congress on Nature and Biologically Inspired Computing (NaBIC'10) (Kitakyushu)*, Dec. 2010, pp. 183–188.

[11] B. Zhou, C. Yang, H. Guo, and J. Hu, "A quasi-linear SVM combined with assembled SMOTE for imbalanced data classification," in *Proc. of 2013 IEEE International Joint Conference on Neural Networks (IJCNN'2013) (Dallas)*, August 2013, pp. 2351–2357.

[12] Y. Cheng, L. Wang, and J. Hu, "Identification of quasi-ARX neuro-fuzzy model with an SVR and GA approach," *IEICE Trans on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E95-A, no. 5, pp. 876–883, May 2012.

[13] J. Hu, K. Kumamaru, and K. Hirasawa, "A quasi-ARMAX approach to modeling of nonlinear systems," *International Journal of Control*, vol. 74, no. 18, pp. 1754–1766, 12 2001.

[14] B. Li, Q. Wang, and J. Hu, "Fast SVM training using edge detection on very large datasets," *IEEJ Trans. on Electrical and Electronic Engineering*, vol. 8, no. 3, May 2013.

[15] T. Joachims, "Transductive inference for text classification using support vector machines," in *Proc. 16th International Conf. on Machine Learning*, 1999, pp. 200–209.

[16] J. C. Gower and G. Ross, "Minimum spanning trees and single linkage cluster analysis," *Applied statistics*, pp. 54–64, 1969.

[17] H. Zhao, Z. Xu, S. Liu, and Z. Wang, "Intuitionistic fuzzy MST clustering algorithms," *Computers & Industrial Engineering*, vol. 62, no. 4, pp. 1130–1140, May 2012.

[18] O. Grygorash, Y. Zhou, and Z. Jorgensen, "Minimum spanning tree based clustering algorithms," in *Proc. of 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06)*, 2006, pp. 73–81.

[19] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proceedings of the American Mathematical Society*, vol. 7, no. 1, pp. 48–50, 1956.

[20] Z. Barutcuoglu, R. Schapire, and O. Troyanskaya, "Hierarchical multi-label prediction of gene function," *Bioinformatics*, vol. 22, no. 7, pp. 830–836, 2006.

[21] A. Elisseeff and J. Weston, "A kernel method for multilabelled classification,," 2001, pp. 681–687.