Neurocomputing 000 (2017) 1–13

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Scalable transfer support vector machine with group probabilities

Tongguang Ni[a], Xiaoqing Gu[a,*], Jun Wang[b,c], Yuhui Zheng[d], Hongyuan Wang[a]

[a] School of Information Science and Engineering, Changzhou University, Changzhou, Jiangsu 213164, China
[b] School of Digital Media, Jiangnan University, Wuxi, Jiangsu 214122, China
[c] Department of Radiology and BRIC, School of Medicine, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599, USA
[d] School of Computer and software, Nanjing University of Information Science & Technology, Nanjing, Jiangsu 210044,China

ARTICLE INFO

Article history:
Received 26 July 2016
Revised 22 August 2017
Accepted 29 August 2017
Available online xxx

Communicated by Dr Zhang Zhaoxiang

Keywords:
Large datasets
Classification
Support vector machine
Transfer learning
Group probability

ABSTRACT

A novel transfer support vector machine called TSVM-GP with group probabilities is proposed for the scenarios where plenty of labeled data in the source domain and the group probabilities of unlabeled data in the target domain are available. TSVM-GP integrates a transfer term and group probabilities into the support vector machine (SVM) to improve the classification accuracy. In order to reduce the high computational complexity of TSVM-GP, the scalable version of TSVM-GP called scalable transfer support vector machine with group probabilities (STSVM-GP) is further developed by selecting the representative set of the training samples as the training data in the source domain. Experimental results on synthetic datasets as well as several real-world datasets show the effectiveness of the proposed classifiers, and especially STSVM-GP is very feasible for large scale transfer datasets.

1. Introduction

Learning from group probabilities [1–6] is attractive in the scenarios where the samples are provided as groups and only the label proportion of the samples is available. One of the most natural applications comes in analyzing the outcomes of political elections, where the population of all voters in an electoral district is known, but only the total number of votes per party in each district is revealed. However, from an analysis of this data, e.g. the dependence of votes on variables such as income or household types, can show up interesting connections, and may be used to uncover election fraud when outliers from this model are uncovered. This case is quite different from the traditional supervised, unsupervised and semi-supervised learning problems. Fig. 1 shows the difference among several traditional learning algorithms, including: supervised learning, unsupervised learning, semi-supervised learning and learning from group probabilities. Learning from group probabilities can be regarded as an algorithm lying somewhere between supervised learning algorithms and semi-supervised learning algorithms.

Various algorithms have been developed by utilizing the group probabilities. For example, Quadrianto et al. [2] applied consistent estimators which could reconstruct the correct labels with high probability into a uniform convergence sense. Later, Rüping [3] proposed a parametric classifier, called IC-SVM, which integrated inverse calibration technology into support vector regression (SVR). Besides, Stolpe and Morik [4] developed a clustering based algorithm to learn from label proportions. Recently, Qi et al. [6] introduced an effective model called LLPs via nonparallel support vector machine (LLP-NPSVM). LLP-NPSVM determined the label of samples according to two nonparallel hyper-planes under the supervision of label proportion information. The current study on learning from group probabilities often assumes that the training set is large enough to train a robust classifier [3]. However, this assumption may not always hold. In practice, the data features or data distributions may be different, which leads to the lack of group probabilities since there are not enough groups in corresponding data. As a result, it may not be able to directly apply the classifiers learned on the learning tasks with group probabilities. It would be helpful if the samples of relative source domains can be transferred into the target domain. For the application of political election discussed above, the vote data in previous years will be helpful for the learning tasks of this year. In such cases, transfer learning between task domains would be desirable [7–11]. Transfer learning is motivated by the fact that people can apply previously learned knowledge to solve new problems. Transfer learning builds a model for the target domain by leveraging the label information from another related domain (source domain), such as the vote data collected in other time frames or with other data collection setups, thus it avoids the costly process of

* Corresponding author.
E-mail addresses: hbxtntg-12@163.com (T. Ni), czxqgu@163.com (X. Gu), zhengyh@vip.126.com (Y. Zheng), hywang@cczu.edu.cn (H. Wang).

http://dx.doi.org/10.1016/j.neucom.2017.08.049
0925-2312/

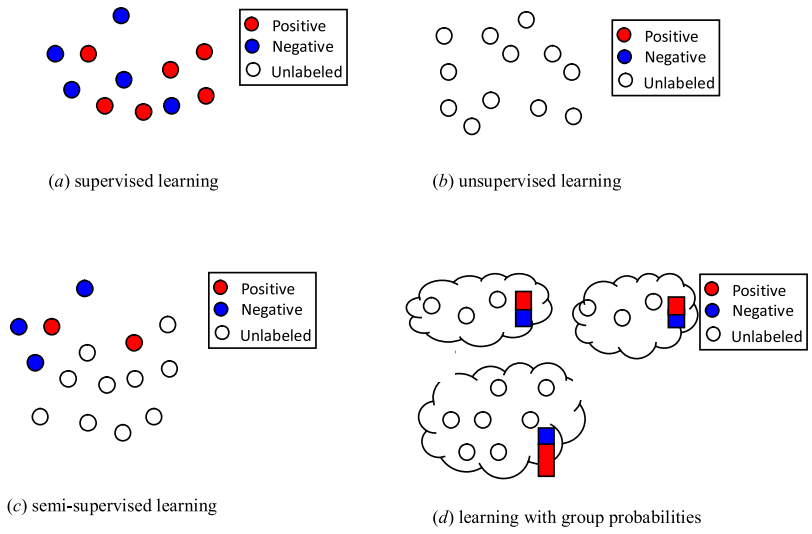


Fig. 1. Difference between (among) the proposed classifiers and the often-used learning algorithms (colors encoding class labels). (a): supervised learning: labeled data are given; (b): unsupervised learning: unlabeled data are given; (c): semi-supervised learning: labeled and unlabeled data are given; (d): learning with group probabilities: class label proportion of unlabeled data are given.

generating labels for the samples in the target domain. Different from conventional machine learning algorithms which assume that the training data should have the same distribution as that of the test data, transfer learning is able to utilize knowledge from data which follows a different distribution. Up to now, many transfer learning algorithms have been successfully applied in many areas, such as speech recognition, computer vision, information retrieval and natural language processing [12–15].

In this work, we construct a novel transfer learning framework with group probabilities using labeled samples in the source domain and the group probabilities in the target domain. Furthermore, inspired by Inverse Calibration [3], with the ε-insensitive loss function, a transfer support vector machine with group probabilities (called TSVM-GP) is proposed. TSVM-GP can be solved using the classical quadratic programming (QP) solver [16].

However, the number of samples in the source domain is usually very large and computing the corresponding kernel matrix using the naive QP solver is $O(n^3)$ (n is the number of training samples) computationally complex. All these factors heavily limit the applicability of TSVM-GP on large scale transfer datasets. Many endeavors have been made to develop various techniques for scaling up the QP solver. Typical techniques involve chunking and some other complicated decomposition methods such as sequential minimal optimization (SMO) algorithm [17], minimum enclosing ball (MEB) [18], core vector machines (CVM) [19], fastKDE [20] and so on. Recently, AESVM [21] is proposed as a fast SVM training algorithm, whose implementation is found to be much faster than many state-of-the-art kernel methods, whereas its classification accuracy is comparable to these existing algorithms. Based on the definition of the convex hull in the kernel space, AESVM effectively finds the convex hull vertices of the training data (called representative set) in the kernel space, then uses the representative set as the training data to build a SVM classifier. Obviously, the size of the training data is effectively decreased and the training time is much shorter.

In this paper, in order to scale TSVM-GP on large scale transfer datasets, we introduce the AESVM algorithm into TSVM-GP and develop a scalable transfer support vector machine with group probabilities (STSVM-GP). First, the convex hull vertices of samples (representative set) in the source domain and their corresponding weights are computed based on the idea of AESVM algorithm. Then, these selected samples and their weights as well as the group probabilities of the target domain are feed into TSVM-GP to build a classifier. Therefore, the scalable version of TSVM-GP, i.e., STSVM-GP, is proposed to implement the fast training algorithm on large scale transfer datasets.

The contribution of this work can be summarized as the following aspects.

(1) A novel transfer learning classifier TSVM-GP is proposed. TSVM-GP simultaneously utilizes both the labeled samples in the source domain and the group probabilities of the unlabeled samples in the target domain in a transfer learning framework. The training procedure of TSVM-GP can be equivalently transformed as a classical QP problem and it guarantees a global optimal solution.
(2) By using both the representative set selected from the source domain and group probabilities of the target domain as the training set, the scalable classifier STSVM-GP of TSVM-GP is developed for fast training on large scale transfer datasets. It guarantees that the selected samples can retain the greatest amount of information of data in the source domain. The performance of STSVM-GP is much closed to that of TSVM-GP but STSVM-GP consumes much less training time.
(3) Extensive experiments on synthetic and real-world datasets are conducted and the experimental results demonstrate that the proposed classifiers are at least comparable to several state-of-the-art algorithms in terms of classification accuracy; moreover, STVM-GP is very feasibility for large scale transfer datasets in terms of training time.

The rest of this paper is organized as follows. The related concepts of classic learning of group probabilities: Inverse Calibration and AESVM are reviewed in Section 2. In Section 3, the proposed classifier TSVM-GP is proposed. In Section 4, the proposed classifier STSVM-GP is proposed. The experimental results on synthetic and real-world datasets are reported in Section 5. Finally, conclusions and the potential of the proposed classifiers are given in the last section.

2. Related works

2.1. Inverse Calibration (IC)

Inverse Calibration [3] learns a classifier from group probabilities based on the idea of support vector regression (SVR) and

Please cite this article as: T. Ni et al., Scalable transfer support vector machine with group probabilities, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.08.049

classifier calibration [22]. For a binary classification task described by an unknown probability distribution $P(\mathbf{X}, \mathbf{Y})$ on an input space $\boldsymbol{X}$ and a set of labels $\mathbf{Y} = \{-1, 1\}$, let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ be drawn independently and identically distributed from P. $\boldsymbol{X}$ is fixed by several subsets. An estimate of the conditional class probability can be predicted by the Platt calibration [3]:

$$p(y = 1|\mathbf{x}) = 1/(1 + \exp(-Af(\mathbf{x}) + B)), \tag{1}$$

where the parameters A and B are fitted using maximum likelihood estimation from a fitting training set $(\boldsymbol{x}_i, y_i)$. Both A and B can be optimized using gradient descent to minimize the cross-entropy error. The scaling function in Inverse Calibration is:

$$p = \sigma(y) = \frac{1}{1 + \exp(-y)}, \tag{2}$$

which can be considered as a special case of Platt calibration with $A = 1$ and $B = 0$. Using p to imply $p(y = 1|\mathbf{x})$, the label $\tilde{y}$ can be predicted by

$$\tilde{y} = \sigma^{-1}(p) = -\log(\frac{1}{p} - 1). \tag{3}$$

In order to avoid undefined values of y, p is clipped to the interval [ε, 1-ε], where ε is a parameter defining the minimum required precision of the estimate. However, the optimal class probability estimation of the single observation varies dramatically around the average value of p. Let $f(\mathbf{x}) = \mathbf{w}^T\varphi(\mathbf{x}) + b$ be the output of a classifier in the kernel space, where $\varphi(\mathbf{x})$ is a mapping function φ: $\mathbf{R}^d \to \mathbf{R}^D (D >> d)$ that maps the original data into a high, possible dimensional feature space. Thus, f predicts $\tilde{y}$ well on average:

$$\forall i : \frac{1}{|S_i|}\sum_{j \in S_i}(\mathbf{w}^T\varphi(\mathbf{x}_j) + b) \approx \tilde{y}_i, \tag{4}$$

where S_i is the ith subset, $|S_i|$ denotes the number of samples in subset S_i.

Inverse Calibration formulates optimization problem is as following:

$$\begin{aligned}
&\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i^*),\\
&\text{s.t. } \forall_{i=1}^m : \frac{1}{|S_i|}\sum_{j \in S_i}(\mathbf{w}^T\varphi(\mathbf{x}_j) + b) \geq \tilde{y}_i - \varepsilon_i - \xi_i,\\
&\forall_{i=1}^m : \frac{1}{|S_i|}\sum_{j \in S_i}(\mathbf{w}^T\varphi(\mathbf{x}_j) + b) \leq \tilde{y}_i + \varepsilon_i + \xi_i^*,\\
&\xi_i \geq 0, \ \ \xi_i^* \geq 0.
\end{aligned} \tag{5}$$

where m is the number of groups. It is well known that the above optimization problem in Eq. (5) can be efficiently solved in its dual form, which is:

$$\begin{aligned}
&\min_{\alpha,\alpha^*} \frac{1}{2}\sum_{i,j=1}^{m}\frac{(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)}{|S_i||S_j|}\sum_{i' \in S_i, j' \in S_j} k(\mathbf{x}_{i'}, \mathbf{x}_{j'})\\
&\quad + \sum_{i=1}^{m}(\alpha_i(\varepsilon_i - \tilde{y}_i) + \alpha_i^*(\varepsilon_i + \tilde{y}_i)),\\
&\text{s.t.} \sum_{i=1}^{m}(\alpha_i - \alpha_i^*) = 0,\\
&\forall_{i=1}^m : 0 \leq \alpha_i, \ \alpha_i^* \leq C.
\end{aligned} \tag{6}$$

where the inner product $k(\mathbf{x}, \mathbf{x}') = \varphi(\mathbf{x})^T\varphi(\mathbf{x}')$ is the kernel function.

2.2. Fast SVM training using approximate extreme points (AESVM)

The basic idea of SVM is trying to offer a hyperplane that represents the largest separation (or margin) between two classes. Geometrically, the maximum-margin hyperplane is equivalent to compute the nearest samples between the convex hulls of two classes [23,24]. Based on the definition of convex hull in the kernel space, AESVM [21] selects the convex hull vertices of the positive and negative training samples (called the representative set) separately as the inputs to a SVM classifier. One of the advantages of AESVM is that its solution has been theoretically proved close to the original solution within a small approximation bound.

Given a dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \in \mathbf{R}^d$ and the corresponding class label set $\mathbf{Y} = \{y_1, y_2, \ldots, y_n\}$, where $y_i = 1$ or -1 ($i = 1, \ldots, n$). The constrained optimization problem of the AESVM model can be described as follows:

$$\min_{\mathbf{w},b} F_{AESVM}(\mathbf{w}, b) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + \frac{C}{M}\sum_{i=1}^{M}\beta_i l(\mathbf{w}, b, \varphi(\mathbf{x}_i)), \tag{7}$$

where $l(\mathbf{w}, b, \varphi(\mathbf{x}_i)) = max\{0, \ 1 - y_i(\mathbf{w}^T\varphi(\mathbf{x}_i) + b)\}$, $\mathbf{x}_i \in \mathbf{X}^*$. $\mathbf{X}^*$ is the representative set of the training set with M samples obtained by AESVM. The vector $\beta = [\beta_1, \beta_2, \ldots, \beta_M]^T$ is the weight vector of representative samples.

It is noted that the process of representative set $\mathbf{X}^*$ determination is executed on the positive and negative class separately. Firstly, the samples belonging to each class are divided into several subsets $\mathbf{X}_q$ based on the certain segregation strategy. Then, the representative set $\mathbf{X}_q^*$ and the weight vector β_q are computed in each subset. Finally, the representative set $\mathbf{X}^*$ is obtained by combining all the sets of $\mathbf{X}_q^*$ together. Here, we briefly discuss how to determine the representative set $\mathbf{X}_q^*$ with its weight vector β_q in each subset $\mathbf{X}_q$. Firstly, the initialized representative set $\mathbf{X}_q^*$ is computed by using the support vector data description (SVDD) [25] algorithm. Secondly, according to the descending order of distance between samples and the center of the SVDD in the kernel space, each sample $\mathbf{x}_i (\mathbf{x}_i \in \mathbf{X}_q$ and $\mathbf{x}_i \notin \mathbf{X}_q^*)$ is checked whether it is a representative sample by using the following form:

$$\begin{aligned}
&\max_{\mathbf{x}_i \in \mathbf{X}_q \text{ and } \mathbf{x}_i \notin \mathbf{X}_q^*} f(\varphi(\mathbf{x}_i), \mathbf{X}_q^*) \leq \tilde{\varepsilon},\\
&\text{s.t.} \quad 0 \leq \mu_{i,t} \leq 1, \sum_{t=1}^{|\mathbf{x}_q^*|}\mu_{i,t} = 1 \text{ and } \mathbf{x}_t \in \mathbf{X}_q^*,
\end{aligned} \tag{8}$$

where $f(\varphi(\mathbf{x}_i), \mathbf{X}_q^*) = \min_{\mu_{it}} \|\varphi(\mathbf{x}_i) - \sum_{t=1}^{|\mathbf{X}_q^*|}\mu_{i,t}\varphi(\mathbf{x}_t)\|^2$, and $|\mathbf{X}_q^*|$ denotes the number of samples in $\mathbf{X}_q^*$. For a given tolerance parameter $\tilde{\varepsilon}$, if the value $f(\varphi(\mathbf{x}_i), \mathbf{X}_q^*)$ is greater than $\tilde{\varepsilon}$, then the sample $\mathbf{x}_i$ can be accepted as a representative sample of the subset $\mathbf{X}_q$ in the kernel space. Then the representative set $\mathbf{X}_q^*$ can be scaled up as $\mathbf{X}_q^* = \mathbf{X}_q^* \cup \{\mathbf{x}_i\}$. Thus, any sample $\mathbf{x}_i (\mathbf{x}_i \in \mathbf{X}_q$ and $\mathbf{x}_i \notin \mathbf{X}_q^*)$ in the kernel space can be formulated as:

$$\varphi(\mathbf{x}_i) = \sum_{\mathbf{x}_t \in \mathbf{X}_q}\gamma_{i,t}\varphi(\mathbf{x}_t) + \tau_i, \tag{9}$$

where $\gamma_{i,t} = \{\mu_{i,t}$ for $\mathbf{x}_t \in \mathbf{X}_q^*$ and $\mathbf{x}_i \in \mathbf{X}_q$, and 0 otherwise $\}$ and $\|\tau_i\|^2 \leq \tilde{\varepsilon}$.

Finally, the element β_t in the weight vector $\boldsymbol{\beta}$ can be determined as follows:

$$\beta_t = \sum_{i=1}^{N}\gamma_{i,t}. \tag{10}$$

3. Transfer support vector machine with group probabilities (TSVM-GP)

In this section, we give a detailed description of the proposed transfer support vector machine with group probabilities

Please cite this article as: T. Ni et al., Scalable transfer support vector machine with group probabilities, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.08.049

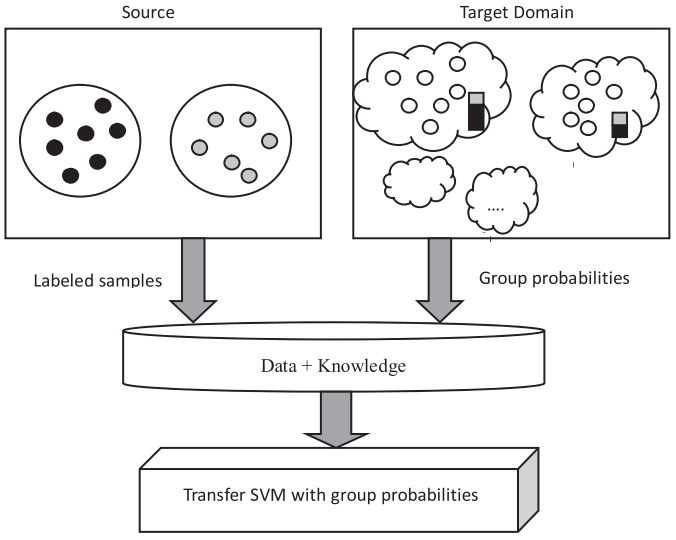


Fig. 2. Framework of TSVM-GP.

(TSVM-GP), and describe the framework of TSVM-GP in Fig. 2. As shown in Fig. 2, there are two major information sources available for the TSVM-GP learning, the labeled samples in the source domain and the unlabeled samples with group probabilities in the target domain. For simplicity, we consider binary classification problems. We denote a source domain D_s as $D_s = (\mathbf{x}_i, y_i)_{i=1,\ldots,n}$ based on the joint distribution P_s, $\mathbf{x}_i$ is the input vector and $y_i \in \{-1, 1\}$ is the corresponding label. Similarly, a target domain D_t is denoted as $D_t = (\boldsymbol{x}_i)_{i=1,\ldots,d}$ according to the joint distribution P_t. Suppose $p_k = |\{i \in S_k : y_i = 1\}|/|S_k|$ be the estimate of the conditional class probability $P(\mathbf{Y} = 1|S_k)$. Let $P_s(\boldsymbol{x}_s)$ and $P_t(\boldsymbol{x}_t)$, $P_s(y_s|\boldsymbol{x}_s)$ and $P_t(y_t|\boldsymbol{x}_t)$ be the marginal distribution and the conditional distributions of D_s and D_t, respectively. In general, $P_s \neq P_t$, $P_s(\boldsymbol{x}_s) \neq P_t(\boldsymbol{x}_t)$ and $P_s(y_s|\boldsymbol{x}_s) \neq P_t(y_t|\boldsymbol{x}_t)$. Our goal is to predict the labels of the inputs in the target domain D_t.

The framework of TSVM-GP can be formulated as the following optimization problem:

$$\min_{f_s, f_c \in \mathbf{H}_K} \frac{1}{2}\|f_s\|_K^2 + C_s \sum_{i=1}^{n} l_s(f_s, y_i) + \frac{1}{2}\|f_t\|_K^2 + C_t \sum_{i=1}^{d} l_t(f_t, y_i) + \lambda d(f_t, f_s), \tag{11}$$

where f_s and f_t are the decision functions in the source domain and target domain, respectively. $\|f_s\|_K^2$ and $\|f_t\|_K^2$ are the structure risk terms controlling the complexity of the classifier in the source domain and the target domain, respectively. Moreover $\|f\|^2$ indicates the 2-norm of function f. C_s and C_t are the regularization coefficients in the source domain and the target domain, respectively. The function $l()$ is a convex non-negative loss function. The function $d()$ is used to quantify the diversity between two domains. The λ is the trade-off parameter.

The optimization criterion in Eq. (11) contains three terms. The first term $(\frac{1}{2}\|f_s\|_K^2 + C_s \sum_{i=1}^{n} l_s(f_s, y_i))$ refers to the knowledge learning from the data in the source domain. The second term $(\frac{1}{2}\|f_t\|_K^2 + C_t \sum_{i=1}^{d} l_t(f_t, y_i))$ refers to the knowledge learning from the data in the source domain. The third term $(\lambda d(f_t, f_s))$ is the regularizer term that guarantees good generalization performance by minimizing the difference between the two domains. So TSVM-GP deals with the classification problem as accurately as possible both in the source domain and the target domain.

Following the common strategy in SVM, a possible classification hyperplane in TSVM-GP can be represented by $\mathbf{w}^{\mathbf{T}}\varphi(\mathbf{x}) + b = 0$. In further, a simple quadratic distance measure, i.e., $d(f_t, f_s) = \frac{\lambda}{2}\|\mathbf{w}_t - \mathbf{w}_s\|^2$, is used to quantify the diversity between domains. So that the optimization problem in Eq. (11) can be rewritten as:

$$\min_{\mathbf{w}_t, \mathbf{w}_s, b_t, b_s} \frac{1}{2}\|\mathbf{w}_t\|^2 + C_t \sum_{i=1}^{d} l_t(\mathbf{w}_t^T\varphi(\mathbf{x}) + b_t, y_i) + \frac{1}{2}\|\mathbf{w}_s\|^2 + C_s \sum_{i=1}^{n} l_s(\mathbf{w}_s^T\varphi(\mathbf{x}) + b_s, y_i) + \frac{\lambda}{2}\|\mathbf{w}_t - \mathbf{w}_s\|^2. \tag{12}$$

In order to obtain both good classification performance in the source domain and good probability estimation in the target domain, two different loss functions are chosen in two domains, i.e., hinge loss function $l(f(\mathbf{x}_i), y_i) = max\{0,\ 1 - y_i f(\mathbf{x}_i)\}$ in the source domain and ε-insensitive loss function $l(f(\mathbf{x}_i), y_i) = max\{0,\ |f(\mathbf{x}_i) - \tilde{y}_i| - \varepsilon\}$ in the target domain. Therefore, Eq. (12) can be formulated as an optimization problem:

$$\min_{\mathbf{w}_t, \mathbf{w}_s, b_t, b_s} \frac{1}{2}\|\mathbf{w}_t\|^2 + \frac{1}{2}\|\mathbf{w}_s\|^2 + C_s \sum_{i=1}^{n} \xi_i^s + C_t \sum_{i=n+1}^{n+d} (\xi_i + \xi_i^*) + \frac{\lambda}{2}\|\mathbf{w}_t - \mathbf{w}_s\|^2,$$

$$\text{s.t.} \quad y_i(\mathbf{w}_s^T\varphi(\mathbf{x}_i) + b_s) \geq 1 - \xi_i^s, i = 1, \ldots, n, \tag{13}$$

$$\forall_{i=1}^{d} : \frac{1}{|S_i|}\sum_{j \in S_i}(\mathbf{w}_t^T\varphi(\mathbf{x}_j) + b_t) \geq \tilde{y}_i - \varepsilon_i - \xi_i,$$

$$\forall_{i=1}^{d} : \frac{1}{|S_i|}\sum_{j \in S_i}(\mathbf{w}_t^T\varphi(\mathbf{x}_j) + b_t) \leq \tilde{y}_i + \varepsilon_i + \xi_i^*,$$

where ξ_i^s, ξ_i and ξ_i^* are slack variables. The goal of the first constrain is to separate these samples as accurately as possible in the source domain. The second and third constrains try to keep the class probability estimate of S_i close to p_i in the target domain. In TSVM-GP, ε_i is a parameter defining the minimum precision of the estimate of $\tilde{y}_i$, which satisfies the following scaling function:

$$p_i - \varepsilon \leq \frac{1}{1 + \exp(-\tilde{y})} \leq p_i + \varepsilon. \tag{14}$$

Following [3], ε_i is set to be $\varepsilon_i = \frac{\varepsilon'}{p_i(1-p_i)}$ where p_i is the group probability $P(Y = 1|S_k)$, and ε' is a very small positive constant.

In order to solve the optimization problem in Eq. (13), we develop Theorems 1 and 2 as follows.

Theorem 1. The dual problem of Eq. (13) is a QP problem as shown in Eq. (15).

$$\min_{\beta} \frac{1}{2}\beta^{\mathbf{T}}\tilde{\mathbf{K}}\beta + \tilde{\mathbf{e}}^{\mathbf{T}}\beta, \tag{15}$$

$$\text{s.t. } \mathbf{f}^{\mathbf{T}}\beta = 0,$$

$$\text{where } \beta = [\alpha^s, \alpha, \alpha^*]^{\mathbf{T}},\ 0 \leq \beta \leq [\underbrace{C_s, \ldots, C_s}_{n}, \underbrace{C_t, \ldots, C_t}_{d}, \underbrace{C_t, \ldots, C_t}_{d}],$$

$$\mathbf{f}^{\mathbf{T}} = [y_1, \ldots, y_n, \underbrace{1, \ldots, 1}_{d}, \underbrace{-1, \ldots, -1}_{d}],$$

$$\tilde{\mathbf{e}} = [\underbrace{0, \ldots, 0}_{n}, \varepsilon - \tilde{\mathbf{y}}, \varepsilon + \tilde{\mathbf{y}}],$$

$$\tilde{\mathbf{K}} = \begin{bmatrix} \frac{1+\lambda}{1+2\lambda}\mathbf{K}_{s,s} + \frac{1}{\lambda} & \frac{\lambda}{1+2\lambda}\mathbf{K}_{s,t} & -\frac{\lambda}{1+2\lambda}\mathbf{K}_{s,t} \\ \frac{\lambda}{1+2\lambda}\mathbf{K}_{s,t}^{\mathbf{T}} & \frac{1+\lambda}{1+2\lambda}\mathbf{K}_{t,t} & -\frac{1+\lambda}{1+2\lambda}\mathbf{K}_{t,t} \\ -\frac{\lambda}{1+2\lambda}\mathbf{K}_{s,t}^{\mathbf{T}} & -\frac{1+\lambda}{1+2\lambda}\mathbf{K}_{t,t} & \frac{1+\lambda}{1+2\lambda}\mathbf{K}_{t,t} \end{bmatrix}_{(n+2d)\times(n+2d)},$$

$$\mathbf{K}_{s,s} = (y_i y_j k(\mathbf{x}_i, \mathbf{x}_j))_{i,j=1,\cdots,n},$$

$$\mathbf{K}_{s,t} = \left(\frac{y_i}{|S_k|}\sum_{j \in S_k} k(\mathbf{x}_i, \mathbf{x}_j)\right)_{i=1,\cdots,n,k=1,\ldots,d},$$

$$\mathbf{K}_{t,t} = \left(\frac{1}{|S_i||S_j|}\sum_{i' \in S_i}\sum_{j' \in S_j} k(\mathbf{x}_{i'}, \mathbf{x}_{j'})\right)_{i,j=1,\ldots,d}.$$

The proof of Theorem 1 can be seen in Appendix 1.

Theorem 2. The quadratic form of the optimization problem of Eq. (15) is a standard convex quadratic programming problem.

The proof of Theorem 2 can be seen in Appendix 2.

It is clear from the above results that the optimization problem in Eq. (15) for TSVM-GP model training can be transformed into a convex QP problem and can be directly solved by the traditional SVM solutions.

Suppose $\tilde{\beta} = (\tilde{\alpha}^s, \tilde{\alpha}, \tilde{\alpha}^*)^{\mathbf{T}}$ solve the optimization problem. Then we can obtain the optimal value of $\mathbf{w}_t{}^*$ and $b_t{}^*$ as below:

$$\mathbf{w}_t{}^* = \frac{\lambda}{1+2\lambda}\sum_{i=1}^{n}\tilde{\alpha}_i^s y_i \varphi(\mathbf{x}_i) + \frac{1+\lambda}{1+2\lambda}\sum_{i=n+1}^{n+d}\frac{\tilde{\alpha}_i - \tilde{\alpha}_i^*}{|S_i|}\sum_{j\in S_i}\varphi(\mathbf{x}_j), \quad (16)$$

$$b_t{}^* = y_i - \frac{\lambda}{1+2\lambda}\sum_{j=1}^{n}\frac{\tilde{\alpha}_j^s y_j}{|S_i|}\sum_{k\in S_i}k(\mathbf{x}_j,\mathbf{x}_k) - \frac{1+\lambda}{1+2\lambda}\sum_{j=n+1}^{n+d}\frac{\tilde{\alpha}_j - \tilde{\alpha}_j^*}{|S_j||S_i|}\sum_{l\in S_j}\sum_{k\in S_i}k(\mathbf{x}_l,\mathbf{x}_k). \quad (17)$$

Finally, the decision of TSVM-GP can be represented as follows:

$$f(\mathbf{x}) = \mathbf{w}_t{}^{\mathbf{T}}\varphi(\mathbf{x}) + b_t \quad (18)$$

Based on the analysis above, we summarize the training process for TSVM-GP as follows.

Training algorithm for TSVM-GP

Input:	n labeled samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ in the source domain, m unlabeled samples $\{\mathbf{x}_j\}_{j=n+1}^{n+m}$ in the target domain and d group probabilities $\{(S_k, p_k)\}_{k=1}^{d}$
Output	Decision function $f(\boldsymbol{x})$
Step 1	Compute the output of Inverse Calibration $\tilde{y}_j (j = 1, \ldots, d)$ in the target domain using Eq. (3);
Step 2	Construct the matrix $\tilde{\mathbf{K}}$ using Eq. (15);
Step 3	Compute Lagrange multiplier β by solving Eq. (16) with a QP Solver;
Step 4	Compute $\boldsymbol{w}_t$ using Eq. (16);
Step 5	Compute b_t using Eq. (17);
Step 6	Output the decision function $f(\mathbf{x})$ using Eq. (18).

The computation complexity of TSVM-GP is $O(n + 2d)^3$, where n is the number of the samples in the source domain and d is the number of groups in the target domain. However, the time complexity of TSVM-GP is cubic of n, which makes TSVM-GP not efficient to large scale datasets. In the following section, TSVM-GP will be extended into its scalable version scalable transfer support vector machine with group probabilities (STSVM-GP) to address this issue.

4. Scalable transfer support vector machine with group probabilities (STSVM-GP)

In this section, we extend TSVM-GP to be scalable for large scale transfer datasets. As discussed in Section 1, it is not difficult to collect a large number of samples in the related domain (i.e., source domain) for TSVM-GP. In order to make TSVM-GP to fully utilize large scale data in the source domain, we apply AESVM algorithm to generate the representative set (i.e., convex hull vertices set) of the source domain as the new training set and then utilize this representative set for transfer learning with group probabilities in the target domain. In this way, the size of training dataset in the source domain for TSVM-GP can be greatly shortened, and the time complexity for training the classifier can also be greatly shorted. Meanwhile, the selected representative set contains the greatest amount of information of the data in the source domain. Based on this idea, TSVM-GP is extended to the scalable transfer support vector machine with group probabilities (called STSVM-GP), in which the representative set in the source domain and the group probabilities of data in the target domain are used to build a classifier. The optimization problem of the STSVM-GP classifier can be expressed as

$$\min_{\mathbf{w}_t,\mathbf{w}_s,b_t,b_s}\frac{1}{2}\|\mathbf{w}_t\|^2 + \frac{1}{2}\|\mathbf{w}_s\|^2 + \frac{C_s}{M}\sum_{i=1}^{M}\beta_i\xi_i^h + C_t\sum_{i=M+1}^{M+d}(\xi_i + \xi_i{}^*) + \frac{\lambda}{2}\|\mathbf{w}_t - \mathbf{w}_s\|^2, \quad (19)$$

$$\text{s.t.}\quad y_i(\mathbf{w}_s^T\varphi(\mathbf{x}_i) + b_s) \geq 1 - \xi_i^s, i = 1, \ldots, M,$$

$$\forall_{i=1}^{d}: \frac{1}{|S_i|}\sum_{j\in S_i}(\mathbf{w}_t^T\varphi(\mathbf{x}_j) + b_t) \geq \tilde{y}_i - \varepsilon_i - \xi_i,$$

$$\forall_{i=1}^{d}: \frac{1}{|S_i|}\sum_{j\in S_i}(\mathbf{w}_t^T\varphi(\mathbf{x}_j) + b_t) \leq \tilde{y}_i + \varepsilon_i + \xi_i^*,$$

where M is the size of the representative set, and $\beta = [\beta_1, \beta_2, \ldots, \beta_M]^T$ is the weight vector of representative samples. According to the derivation of TSVM-GP, the dual problem of STSVM-GP can be formulated as the following QP problem:

$$\min_{\mathbf{\Gamma}}\frac{1}{2}\mathbf{\Gamma}^T\tilde{\mathbf{K}}\mathbf{\Gamma} + \tilde{\mathbf{e}}^T\mathbf{\Gamma}, \quad (20)$$

$$\text{s.t. } \mathbf{f}^T\mathbf{\Gamma} = 0,$$

where

$$\mathbf{\Gamma} = [\alpha^s, \alpha, \alpha^*]^T,$$

$$0 \leq \mathbf{\Gamma} \leq [\underbrace{C_s/\beta_1 M, \ldots, C_s/\beta_M M}_{M}, \underbrace{C_t, \ldots, C_t}_{d}, \underbrace{C_t, \ldots, C_t}_{d}],$$

$$\beta_j = \sum_{i=1}^{M}\gamma_{i,j}, \quad \mathbf{f}^T = [y_1, \ldots, y_M, \underbrace{1, \ldots, 1}_{d}, \underbrace{-1, \ldots, -1}_{d}],$$

$$\tilde{\mathbf{e}} = [\underbrace{0, \ldots, 0}_{M}, \varepsilon - \tilde{\mathbf{y}}, \varepsilon + \tilde{\mathbf{y}}],$$

$$\tilde{\mathbf{K}} = \begin{bmatrix} \frac{1+\lambda}{1+2\lambda}\mathbf{K}_{s,s} + \frac{1}{\lambda} & \frac{\lambda}{1+2\lambda}\mathbf{K}_{s,t} & -\frac{\lambda}{1+2\lambda}\mathbf{K}_{s,t} \\ \frac{\lambda}{1+2\lambda}\mathbf{K}_{s,t}^T & \frac{1+\lambda}{1+2\lambda}\mathbf{K}_{t,t} & -\frac{1+\lambda}{1+2\lambda}\mathbf{K}_{t,t} \\ -\frac{\lambda}{1+2\lambda}\mathbf{K}_{s,t}^T & -\frac{1+\lambda}{1+2\lambda}\mathbf{K}_{t,t} & \frac{1+\lambda}{1+2\lambda}\mathbf{K}_{t,t} \end{bmatrix}_{(M+2d)\times(M+2d)},$$

$$\mathbf{K}_{s,s} = (y_i y_j k(\mathbf{x}_i, \mathbf{x}_j))_{i,j=1,\cdots,M},$$

$$\mathbf{K}_{s,t} = \left(\frac{\tilde{y}_i}{|S_k|}\sum_{j\in S_k}k(\mathbf{x}_i,\mathbf{x}_j)\right)_{i=1,\cdots,M,k=1,\ldots,d},$$

$$\mathbf{K}_{t,t} = \left(\frac{1}{|S_i||S_j|}\sum_{i'\in S_i}\sum_{j'\in S_j}k(\mathbf{x}_{i'},\mathbf{x}_{j'})\right)_{i,j=1,\ldots,d}.$$

Suppose $\tilde{\mathbf{\Gamma}} = (\tilde{\alpha}^s, \tilde{\alpha}, \tilde{\alpha}^*)^T$ solve the optimization problem. The optimal $\mathbf{w}_t{}^*$ and $b_t{}^*$ can be written as

$$\mathbf{w}_t{}^* = \frac{\lambda}{1+2\lambda}\sum_{i=1}^{M}\tilde{\alpha}_i^s y_i\varphi(\mathbf{x}_i) + \frac{1+\lambda}{1+2\lambda}\sum_{i=n+1}^{M+d}\frac{\tilde{\alpha}_i - \tilde{\alpha}_i^*}{|S_i|}\sum_{j\in S_i}\varphi(\mathbf{x}_j), \quad (21)$$

$$b_t{}^* = y_i - \frac{\lambda}{1+2\lambda}\sum_{j=1}^{M}\frac{\tilde{\alpha}_j^s y_j}{|S_i|}\sum_{k\in S_i}k(\mathbf{x}_j,\mathbf{x}_k) - \frac{1+\lambda}{1+2\lambda}\sum_{j=n+1}^{M+d}\frac{\tilde{\alpha}_j - \tilde{\alpha}_j^*}{|S_j||S_i|}\sum_{l\in S_j}\sum_{k\in S_i}k(\mathbf{x}_l,\mathbf{x}_k). \quad (22)$$

Let us recall two important properties from AESVM in [21]. These two properties reveal that the optimal solution of AESVM

is very close to that of SVM and the upper error bound of AESVM does not depend on the size of training set. Let $F_S(\mathbf{w}, b) = \frac{1}{2}\|\mathbf{w}_s\|^2 + \frac{C_s}{M}\sum_{i=1}^{M}\beta_i\xi_i^h$ represent knowledge learning from the data in the source domain in STSVM-GP. Since the term $F_S(\mathbf{w}, b)$ has the same formulation with $F_{AESVM}(\mathbf{w}, b)$, the properties concluded on the AESVM algorithm will hold true for STSVM-GP. Thus, we can state that STSVM-GP exhibits almost the same classification performance or at least comparable performance with TSVM-GP.

Property 1 [21]: $F_{SVM}(\mathbf{w}_1^*, b_1^*) - F_{AESVM}(\mathbf{w}_2^*, b_2^*) \le C\sqrt{C\tilde{\varepsilon}}$.
Property 2 [21]: $F_{SVM}(\mathbf{w}_2^*, b_2^*) - F_{AESVM}(\mathbf{w}_1^*, b_1^*) \le 2C\sqrt{C\tilde{\varepsilon}}$.

where $F_{SVM}(\mathbf{w}, b)$ and $F_{AESVM}(\mathbf{w}, b)$ are the optimal problem for SVM and AESVM, respectively. $(\mathbf{w}_1^*, b_1^*)$ and $(\mathbf{w}_2^*, b_2^*)$ are the optimal solutions of SVM and AESVM respectively. C is the regularization coefficient, and $\tilde{\varepsilon}$ is the tolerance parameter.

Based on the above, the learning algorithm of the proposed STSVM-GP is developed and stated as follows:

Learning algorithm for STSVM-GP	
Input:	n labeled examples $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ in the source domain, m unlabeled examples $\{\mathbf{x}_j\}_{j=n+1}^{n+m}$ in the target domain and d group probabilities $\{(S_k, p_k)\}_{k=1}^{d}$
Output	Decision function $f(\boldsymbol{x})$
Step 1	Compute the output of Inverse Calibration $\tilde{y}_j (j = 1, \ldots, d)$ in the target domain using Eq. (3);
Step 2	Compute the representative set of the data of the source domain [21];
Step 3	Compute Lagrange multiplier $\boldsymbol{\Gamma}$ by solving Eq. (20) with a QP Solver;
Step 4	Compute $\boldsymbol{w}_t$ using Eq. (21);
Step 5	Compute b_t using Eq. (22);
Step 6	Output the decision function $f(\mathbf{x})$ using Eq. (18).

The overall complexity of STSVM-GP includes two stages: representative set selection stage by AESVM and classifier building stage. Given an original training dataset with n samples in the source domain, the time complexity in representative set selection is $O(n(\log_2\frac{P}{V} + \frac{P}{V} + \log_2 V) + V\sum_{q=1}^{Q}\sum_{r=1}^{R}A_{qr}^2)$, where P, V, Q and R are four parameters involved in the segregation strategy in AESVM, and A_{qr} is the latest size of representative set of each segregation during the iterations. In classifier building stage, with M representative samples in the source domain and d samples in the target domain, the corresponding time complexity is $O(M + 2d)^3$. Since the time complexity in representative set selection is much less than that in the latter stage, the overall time complexity of STSVM-GP will scale at $O(M + 2d)^3$. In contrast to the TSVM-GP, the time cost of STSVM-GP is smaller.

5. Experimental results

In this section, the proposed TSVM-GP and STVM-GP are evaluated by both toy and real-world datasets and its performance is benchmarked with four representative classifiers, including: SVM [26], IC-SVM [3], locally-weighted ensemble (LWE) classifier [27], large margin kernel projected TSVM (LMPROJ) [28]. The experiments are organized as follows. The experimental settings are firstly described in Section 5.1. Then, Sections 5.2 and 5.3 report the experimental results in synthetic and real-world datasets respectively. Finally, parameter sensitivity analysis of the proposed classifiers is reported in Section 5.4. The TSVM-GP, STVM-GP, IC-SVM, LWE, and LMPROJ are implemented with MATLAB. In particular, SVM is implemented using LIBSVM [29]. All experiments are carried on a 2.6 GHz machine with 3 G RAM.

5.1. Experimental setting

In order to make the experimental results fair, we repeat the 5-fold cross validation strategy three times by randomly generating five folds on the available data in which each fold keeps the same proportion of positive and negative samples as in the available data. For four transfer learning classifiers (LWE, LMPROJ, TSVM-GP and STSVM-GP), all labeled source data and the 5% randomly selected unlabeled target data are used as training set; while only the labeled source data is used to train SVM and 5% selected unlabeled target data is used to train IC-SVM. The remaining unlabeled target data is used for testing.

In our experiments, the average classification accuracy with standard deviation and training time (in seconds) on each classification task for 15 runs are recorded. The classification accuracy which is widely used in literature [30] is represented as,

$$\text{Accuracy} = \frac{|\mathbf{x} : \mathbf{x} \in D_t \wedge \tilde{y}(\mathbf{x}) = y(\mathbf{x})|}{|\mathbf{x} : \mathbf{x} \in D_t|} \times 100\%,$$

where D_t is the set of test data, $y(\mathbf{x})$ is the truth label of $\boldsymbol{x}$, $\tilde{y}(\mathbf{x})$ is the label predicted by the classifiers.

For each classifier, we select a Gaussian kernel function in the form $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|/2\sigma^2)$, the width parameter $2\sigma^2$ is searched according to the following grids $\{s/64, s/32, s/16, s/8, s/4, s/2, s, 2s, 4s, 8s, 16s, 32s, 64s\}$, where s is the mean squared norm of the training data. The parameters C_1, C_2 and λ of the proposed classifiers are determined by searching the grid $\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}, 10^{4}\}$. Following [3], the group size k is set as $k = 16$, i.e., the target datasets are randomly partition into sets of size 16 for IC-SVM, TSVM-GP and STSVM-GP. For LWE, LMPROJ, and AESVM, the default parameter settings in their literatures are adopted in our experiments.

5.2. Experiments on synthetic datasets

The original source data in three two-moon datasets [31] (called two-moon1, two-moon2 and two-moon3) contains 200, 800 and 5000 samples respectively, with the ratio of positive samples and negative samples 1:1. The target data is generated by rotating counter-clockwise the source data by 15°, 30°, 45°. Thus, there are three datasets with different rotating degrees involved in two-moon1, two-moon2 and two-moon3, respectively. Due to the limited size of the paper, we only present the experimental results for those data of two-moon1, two-moon2, and two-moon3 datasets where target data is generated by rotating counter-clockwise the original source data by 15° as shown in Figs. 3–5. As in Fig. 3 for example, the dataset is composed of 100 positive class samples (i.e., SD+) in the source domain, denoted by red ".", 100 negative class samples (i.e., SD-) in the source domain, denoted by blue ".", and 200 unlabeled samples (i.e., TD) in the target domain, denoted by black ".". The two-moon1 dataset in Fig. 4 and the two-moon3 dataset in Fig. 5 are composed of 1600 and 10,000 samples, respectively, generated by rotating counter-clockwise the original source data by 15°. Thus, the samples in the source and target domains belonging to these three two-moon datasets exhibit different distributions due to rotation. Particularly, the greater is the rotation angle, the more differences between the source domain and the target domain. Figs. 3–5 also show the best decision surfaces of different classifiers on three datasets generated by rotating counter-clockwise the original source data by 15°. Table 1 reports the average running time of different classifiers on three two-moon datasets shown in Figs. 3–5. Moreover, Table 2 reports the average classification accuracy of different classifiers on the two-moon datasets with different sizes and different rotating degrees. In order to save space, here we did not report the running time for two-moon2 and two-moon3 by rotating 30° and 45°.

From Figs. 3–5, Tables 1 and 2, we can observe the following results:

(1) SVM can only separate the binary class samples in the source domain correctly by considering the margin maxi-

Please cite this article as: T. Ni et al., Scalable transfer support vector machine with group probabilities, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.08.049

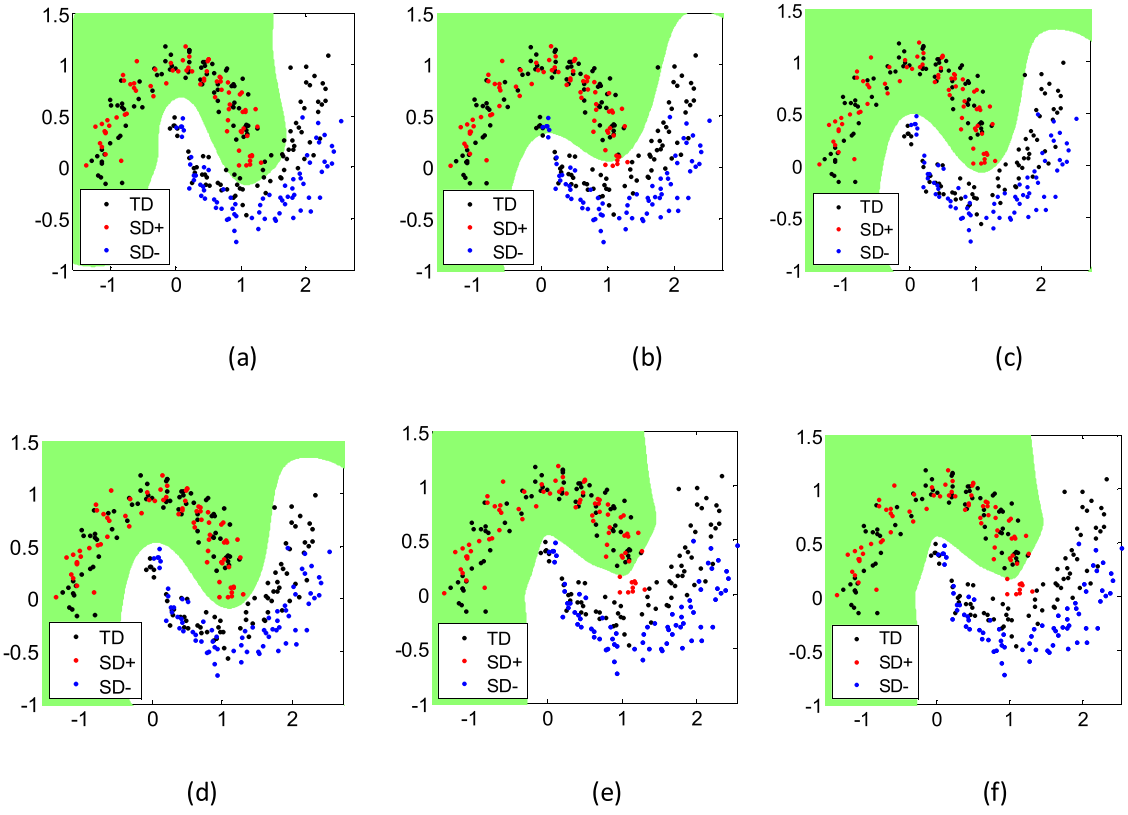


Fig. 3. Best decision surfaces obtained by using different algorithms on the two-moon1 generated by rotating counter-clockwise the original source data 15°. (a) SVM, (b) IC-SVM, (c) LWE, (d) LMPROJ, (e) TSVM-GP, (f) STSVM-GP.

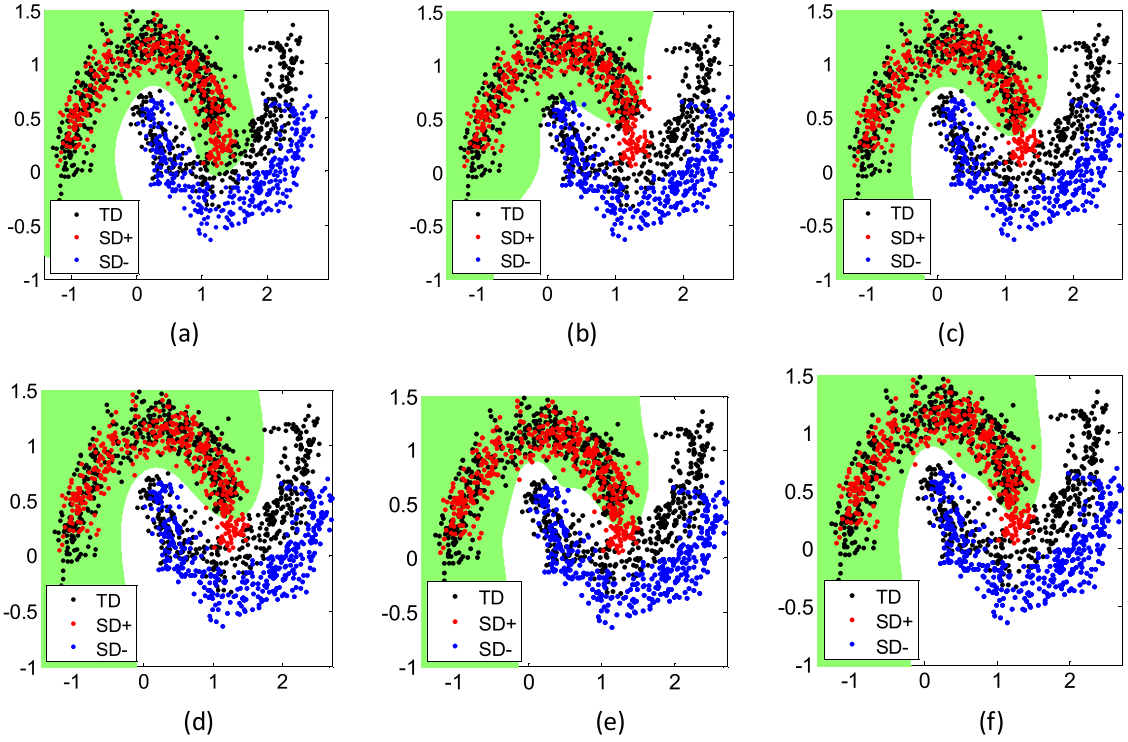


Fig. 4. Best decision surfaces obtained by using different algorithms on the two-moon1 generated by rotating counter-clockwise the original source data 15°. (a) SVM, (b) IC-SVM, (c) LWE, (d) LMPROJ, (e) TSVM-GP, (f) STSVM-GP.

Please cite this article as: T. Ni et al., Scalable transfer support vector machine with group probabilities, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.08.049

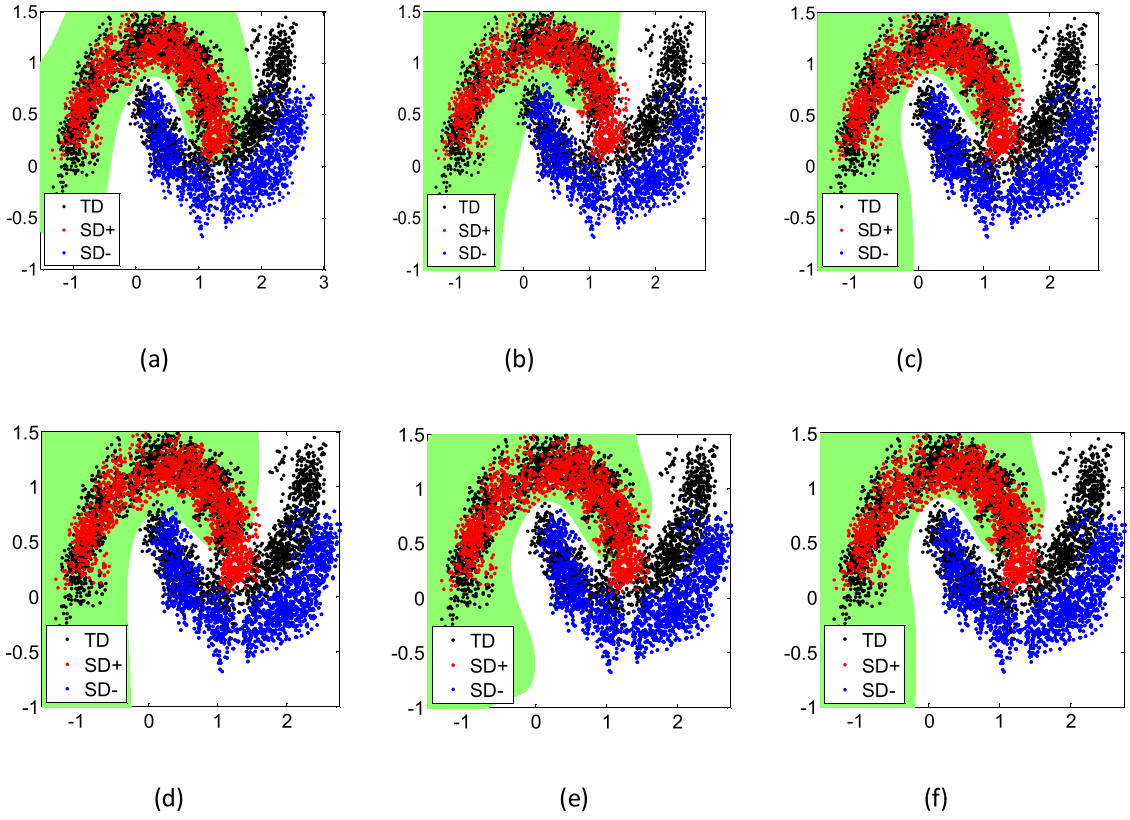


Fig. 5. Best decision surfaces obtained by using different algorithms on the two-moon3 generated by rotating counter-clockwise the original source data 15°. (a) SVM, (b) IC-SVM, (c) LWE, (d) LMPROJ, (e) TSVM-GP, (f) STSVM-GP.

Table 1
Comparison of the average running time with standard deviation on the two-moon datasets with different sizes generated by rotating counter-clockwise the original source data 15°.

Classifiers	Training time (s)		
	two-moon1	two-moon2	two-moon3
SVM	$6.30(\pm0.35)\times10^{-4}$	$1.47(\pm0.12)\times10^{-3}$	$2.65(\pm0.3)\times10^{-2}$
IC-SVM	$2.21(\pm0.21)\times10^{-4}$	$2.51(\pm0.31)\times10^{-4}$	$8.18(\pm0.02)\times10^{-3}$
LWE	$7.49\ (\pm0.39)\times10^{-2}$	$3.93\ (\pm0.21)\times10^{-1}$	$3.12(\pm0.11)$
LMPROJ	$5.64\ (\pm0.34)\times10^{-2}$	$1.88\ (\pm0.13)\times10^{-1}$	$1.33(\pm0.05)$
TSVM-GP	$1.09\ (\pm0.44)\times10^{-2}$	$3.96(\pm0.12)\times10^{-2}$	$1.43\ (\pm0.04)\times10^{-1}$
STSVM-GP	$1.89\ (\pm0.31)\times10^{-3}$	$6.53\ (\pm0.13)\times10^{-3}$	$4.22(\pm0.02)\times10^{-2}$

Table 2
Comparison of the average classification accuracy with standard deviation on the two-moon datasets with different sizes and different rotating degrees.

Datasets	Classifiers	Rotating degrees		
		15°	30°	45°
two-moon1	SVM	78.00 (±0.05)	80.00 (±0.00)	78.00 (±0.05)
	IC-SVM	78.00 (±0.05)	80.00 (±0.00)	78.00 (±0.05)
	LWE	100 (±0.00)	100 (±0.00)	100 (±0.00)
	LMPROJ	100 (±0.00)	100 (±0.00)	100 (±0.00)
	TSVM-GP	100 (±0.00)	100 (±0.00)	100 (±0.00)
	STSVM-GP	100 (±0.00)	100 (±0.00)	100 (±0.00)
two-moon2	SVM	93.85(±0.05)	92.00 (±0.05)	92.12(±0.06)
	IC-SVM	92.80 (±0.08)	92.00 (±0.05)	92.45(±0.08)
	LWE	100 (±0.00)	100 (±0.00)	100 (±0.00)
	LMPROJ	100 (±0.00)	100 (±0.00)	100 (±0.00)
	TSVM-GP	100 (±0.00)	100 (±0.00)	100 (±0.00)
	STSVM-GP	100 (±0.00)	100 (±0.00)	100 (±0.00)
two-moon3	SVM	87.33(±0.05)	85.00 (±0.00)	87.12(±0.06)
	IC-SVM	87.79(±0.10)	85.00 (±0.00)	87.25(±0.06)
	LWE	92.50 (±0.19)	92.00 (±0.12)	92.35(±0.18)
	LMPROJ	92.50 (±0.11)	92.40 (±0.09)	92.30 (±0.12)
	TSVM-GP	92.30 (±0.09)	91.70 (±0.05)	91.20 (±0.07)
	STSVM-GP	91.90 (±0.08)	90.90 (±0.05)	90.70 (±0.05)

mization between two classes in the source domain; while for the target domain, which has a different distribution from the source domain, SVM cannot obtain satisfactory performance. IC-SVM also performs poorly for the target domain because the limited groups cannot reveal the enough structure information of the data in the target domain. TSVM-GP and STSVM-GP take full use of both the labeled samples in the source domain and the group probabilities in the target domain, so these two classifiers achieve more precise classification hyperplanes than SVM and IC-SVM. TSVM-GP and STSVM-GP obtain the comparable performance with two transfer learning classifiers, i.e., LWE and LMPROJ. It is worth to mention that TSVM-GP and STSVM-GP only use the group probabilities in the target domain. So they are appropriate to be adopted for certain classification problems, such as demanding for privacy protection.

(2) In terms of Tables 1 and 2, although the classification performance of SVM is not prominent, its training time is efficient on all generated two-moon datasets. Since SVM is implemented by LIBSVM in C++code, it runs much faster than Matlab code under the same environment. IC-SVM occupies less training time than the other four transfer learning algorithms because its training dataset only contains the group probabilities constructed from 5% randomly selected unlabeled target data which is far less than the size of training data for the other classifiers. Both TSVM-GP and STSVM-GP obtain satisfactory classification performance on all generated two-moon datasets, while they have evident advantage on training time compared to LWE and LMPROJ.

Please cite this article as: T. Ni et al., Scalable transfer support vector machine with group probabilities, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.08.049

Table 3
The detailed information of generated transfer datasets on 20newsgroups and email spam datasets.

Datasets	Sizes	Dimensions	Tasks	Source domain		Target domain	
				Positive	Negative	Positive	Negative
20newsgroups	6000	20	comp vs. rec	1500	1500	1500	1500
	6000	20	comp vs. sci	1500	1500	1500	1500
	6000	20	com vs. talk	1500	1500	1500	1500
Email spam data set	5000	20	User1 vs. User2	1250	1250	1250	1250
	5000	20	User2 vs. User3	1250	1250	1250	1250
	5000	20	User3 vs. User1	1250	1250	1250	1250

Table 4
Detailed information of four large scale transfer datasets.

Datasets	Sizes	Dimensions	Tasks	Classes	Source domain	Target domain
USPS	9298	20	USPS vs. MNIST	10	9298	70,000
MNIST	70,000	20	MNIST vs. USPS	2	70,000	9298
5SrRNA	95,172	8	5SrRNA vs. tRNA	2	90,000	90,000
tRNA	228,962	8	tRNA vs. 5SrRNA	2	90,000	90,000

With the increment of training size, STSVM-GP degrades a little in classification performance but consumes much less training time than TSVM-GP. It is noted that we use the quadprog function in Matlab to solve the QP problem for both TSVM-GP and STSVM-GP, and their computation complexity is cubic to the training set size. If a more efficient QP solver is used, TSVM-GP and STSVM-GP will run much faster.

5.3. Experiments on real-world datasets

5.3.1. 20newsgroups and email spam transfer datasets

20newsgroups [32] is a collection of 18,774 news documents and organized in a hierarchical structure, which consists of six main categories and 20 subcategories. In our experiments, in order to generate transfer 20newsgroups dataset, two top categories are chosen, one as positive class and the other as negative class. Then the samples are split based on subcategories, i.e., different subcategories are considered as different domains, and the binary classification task is defined as a top category classification. This splitting strategy ensures that the domains of labeled and unlabeled data are related, since they are under the same top categories. Besides, these domains are also ensured to be different, since they are drawn from different subcategories. The generated three transfer datasets are named as “comp vs. rec”, “comp vs. sci” and “comp vs. talk” with 6000 samples respectively.

Email spam dataset [33] has three email subsets (denoted by User1, User2, and User3, respectively) annotated by three different users. The task is to classify spam and non-spam emails. Since the spam and non-spam emails in the subsets have been differentiated by different users, the data distributions of these three subsets are related but different. Each subset has 2500 emails, in which one half of the emails are non-spam and the other half are spam.

Table 3 lists the detailed information of six transfer datasets on 20newsgroups and email spam datasets. In our experiments, the original high-dimensional data is preprocessed into the final dataset containing 20 effective features using the same feature extraction strategy as that in [34,35] with principal component analysis (PCA).

5.3.2. Large scale transfer dataset

USPS dataset [36] consists of 9298 16×16 image datasets with 256 dimensions. MNIST dataset [36] consists of 70,000 28×28 image datasets with 256 dimensions. By using the feature extraction strategy PCA, we compress these image features into 20 effective features in our experiment. Here two large scale transfer datasets are constructed, i.e., USPS vs. MNIST and MNIST vs. USPS. Both USPS and MNIST have ten classes. By using the one-against-all (OAA) way [37], a multi-class classification task is decomposed into the corresponding binary classification tasks. 5SrRNA and tRNA are two bio-medical datasets composed of two ncRNAs of cod fish (see [38]), which are consist of 95,172 and 228,962 samples, respectively, with eight attributes. The ratio of positive samples and negative samples in both dataset is 1:1. In our experiments, 90,000 samples in 5SrRNA and 90,000 samples in tRNA are randomly selected to construct two large scale transfer datasets: 5SrRNA vs. tRNA and tRNA vs. 5SrRNA.

5.3.3. Experiment results and analysis

Tables 5 and 6 show the average classification accuracies with their standard deviations and the average running time of all the benchmarking classifiers on different transfer learning tasks. From these results, we can have the following conclusions:

(1) In terms of classification accuracy, it can be seen from Table 5 that both TSVM-GP and STSVM-GP obtain improved or at least comparable performance on all transfer learning tasks. Particularly, STSVM-GP is comparable to TSVM-GP in classification performance on all generated transfer datasets as it works on representative set of the source domain and group probabilities of the target domain which indeed reveals the structure of the datasets and exhibits competitive performance to TSVM-GP. Similar to the observations presented in previous subsections, two traditional classifiers (SVM and IC-SVM) have the lower classification accuracy, compared with four transfer learning classifiers.

(2) In terms of training time shown in Table 6, it can also be seen that STSVM-GP has obvious advantage over LWE, LMPROJ and TSVM-GP in training time, especially for large datasets, due to its strong scalability. IC-SVM requires less training time than other five classifiers in our experiments. This is because its training data only contains the group probabilities constructed from the 5% randomly selected unlabeled target data which is much less than the size of the training data for the other classifiers. Although SVM is effective in training time on some small scale datasets, its classification accuracy is not prominent on transfer learning problems.

5.4. Parameter sensitivity analysis

In this subsection, a sensitivity analysis of three parameters, i.e., C_s, C_t and λ in our proposed classifiers is presented to illustrate their influence on the classification performance, while the

Please cite this article as: T. Ni et al., Scalable transfer support vector machine with group probabilities, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.08.049

Table 5
Comparison of average classification accuracy with standard deviation on real-world transfer datasets.

Datasets	SVM	IC-SVM	LWE	LMPROJ	TSVM-GP	STSVM-GP
comp vs. rec	79.65 (±2.11)	82.31 (±2.11)	86.98 (±2.00)	85.09 (±1.77)	85.38 (±1.11)	85.33 (±1.13)
comp vs. sci	72.09 (±1.98)	71.99 (±1.98)	83.19 (±1.12)	82.98 (±1.98)	83.20 (±1.18)	83.07 (±1.01)
com vs. talk	82.29 (±2.85)	84.99 (±2.85)	93.81 (±1.20)	92.92 (±0.90)	92.91 (±0.85)	92.78 (±1.25)
User1 vs. User2	85.65 (±1.51)	90.65 (±1.51)	96.10 (±1.60)	96.30 (±0.22)	96.65 (±1.51)	95.80 (±1.10)
User1vs. User3	85.78 (±2.02)	81.78 (±2.02)	95.50 (±2.05)	95.15 (±1.19)	95.78 (±1.02)	95.38 (±1.20)
User2 vs. User3	81.01 (±1.24)	82.01 (±1.90)	91.87 (±1.01)	92.01 (±0.95)	91.01 (±0.90)	90.85 (±0.99)
USPS vs. MNIST	37.87 (±1.00)	35.00 (±1.97)	54.41 (±1.22)	54.36 (±1.44)	54.07 (±1.01)	54.01 (±1.31)
MNIST vs. USPS	58.05 (±2.00)	57.23 (±1.56)	61.52 (±1.72)	61.72 (±1.37)	61.32 (±1.75)	61.18 (±1.28)
5SrRNA vs. tRNA	83.31 (±1.57)	84.11 (±1.90)	92.29 (±1.29)	92.43 (±1.21)	92.29 (±1.33)	92.08 (±1.72)
tRNA vs. 5SrRNA	85.19 (±1.10)	85.88 (±2.21)	93.56 (±1.30)	93.11 (±1.54)	93.97 (±1.02)	93.41 (±1.12)

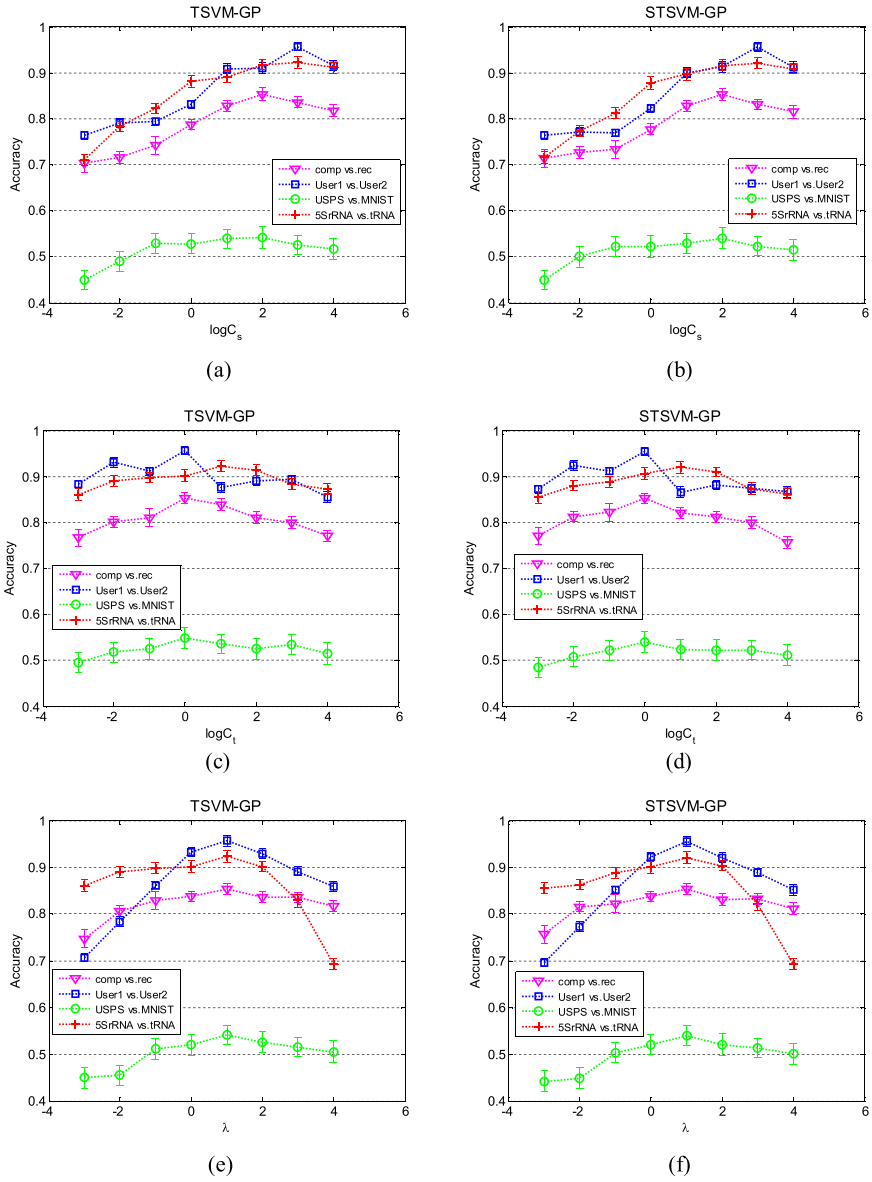


Fig. 6. Parameter sensitivity analysis. (a) Sensitivity of parameter C_s for TSVM-GP, (a) sensitivity of parameter C_s for STSVM-GP, (c) sensitivity of parameter C_t for TSVM-GP, (d) sensitivity of parameter C_t for STSVM-GP, (e) sensitivity of parameter λ for TSVM-GP, (f) sensitivity of parameter λ for STSVM-GP.

parameters for representative set selection in STSVM-GP are set to be the default values in [21]. For each parameter, we fix the other two parameters at their best values determined by cross-validation and then observe the influence of this parameter on classification with different values. The experimental results in the comp vs. rec, User1 vs. User2, USPS vs. MNIST and 5SrRNA vs. tRNA datasets are reported in Fig. 6. From the results, several insights can be obtained as follows:

(1) From Fig. 6(a)–(d), we can see that TSVM-GP is considerably sensitive to regularization parameter C_s and C_t for a wide range of values, as well as STSVM-GP. This implies that it is

Please cite this article as: T. Ni et al., Scalable transfer support vector machine with group probabilities, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.08.049

Table 6
Comparison of average running time with standard deviation on real-world transfer datasets.

Datasets	SVM	IC-SVM	LWE	LMPROJ	TSVM-GP	STSVM-GP
comp vs. rec	1.21 (±0.11)	0.02 (±0.00)	7.68 (±0.32)	8.25 (±0.88)	2.43 (±0.83)	0.58 (±0.32)
comp vs. sci	1.17 (±0.12)	0.02 (±0.00)	7.65 (±0.33)	8.20 (±0.83)	2.27 (±0.72)	0.55 (±0.32)
com vs. talk	1.43 (±0.11)	0.04 (±0.01)	8.12 (±0.21)	8.49 (±0.82)	2.93 (±0.91)	0.62 (±0.31)
User1 vs. User2	1.76 (±0.11)	0.05 (±0.01)	11.22 (±0.85)	12.34 (±0.94)	3.88 (±0.93)	0.99 (±0.35)
User1vs.User3	1.76 (±0.12)	0.05 (±0.01)	11.13 (±0.84)	12.40 (±0.94)	3.86 (±1.12)	0.97 (±0.34)
User2vs.User3	1.73 (±0.11)	0.05 (±0.01)	11.12 (±0.82)	11.39 (±0.91)	3.57 (±1.03)	0.92 (±0.22)
USPS vs. MNIST	71.21 (±0.21)	8.51 (±0.31)	398.18 (±0.62)	402.21 (±8.21)	112.51 (±10.11)	43.18 (±2.02)
MNIST vs. USPS	401.30 (±0.15)	3.47 (±1.12)	1389.65 (±11.30)	1401.30 (±7.35)	1111.47 (±12.12)	142.65 (±3.30)
5SrRNA vs. tRNA	381.49 (±0.79)	13.93 (±1.21)	1253.12 (±9.11)	1231.49 (±10.39)	1023.93 (±10.21)	153.12 (±11.11)
tRNA vs. 5SrRNA	381.64 (±0.64)	15.88 (±1.13)	1259.33 (±11.00)	1232.64 (±7.34)	1021.88 (±11.73)	151.33 (±9.95)

critical to determine this parameter value by some effective strategies, such as cross-validation strategy.

(2) In Fig. 6(e)–(f), we can see that both TSVM-GP and STSVM-GP are sensitive to kernel width parameter λ. It is shown that when λ approaches to 1, the proposed classifiers achieve their best performance. When λ is too small, they cannot achieve the satisfactory performance since the difference between source and target domains might be ignored; in contrast, when λ is too large, although the larger values of λ can make the distribution discrepancy between source domain and target domain more larger in the kernel space in theory, the term $\|\mathbf{w}_t\|^2 + \|\mathbf{w}_s\|^2$ will be no longer useful for building TSVM-GP and STSVM-GP.

6. Conclusions

Designing the classifiers from datasets with group probabilities is an important learning task for practical applications, as well as for scalable datasets. In view of this, we propose a transfer support vector machine with group probabilities (TSVM-GP) by incorporating additional group probabilities into the transfer learning framework. Furthermore, in order to make TSVM-GP scalable to large scale transfer datasets, a scalable transfer support vector machine with group probabilities (STSVM-GP) is proposed by using the representative set of the source domain as the new training set and then utilize them for transfer learning with group probability. The effectiveness of the classifiers is demonstrated using several datasets from the real-world classification datasets as well as using the synthetic datasets.

Although the proposed TSVM-GP and STSVM-GP have shown promising performance, there are still many aspects that deserve further investigation. For example, how to further reduce the computation in proposed classifiers by using more efficient QP solver is a research topic worth to be studied. Furthermore, how to develop a robust classifier for noisy data with group probabilities is also worth to be studied.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grants 61502058 and 61572085.

Appendix 1

Proof of Theorem 1

By using the Lagrangian optimization theorem, we can obtain the following Lagrangian function for Eq. (15)

$$
\begin{aligned}
&L(\mathbf{w}_t, \mathbf{w}_s, b_t, b_s, \xi, \xi^*, \xi^s, \alpha, \alpha^*, \alpha^s, r^s, r, r^*) \\
&= \frac{1}{2}\|\mathbf{w}_t\|^2 + \frac{1}{2}\|\mathbf{w}_s\|^2 + C_s \sum_{i=1}^{n} \gamma_i {\xi_i}^h + C_t \sum_{i=n+1}^{n+d} (\xi_i + {\xi_i}^*) \\
&\quad + \frac{\lambda}{2}\|\mathbf{w}_t - \mathbf{w}_s\|^2 - \sum_{i=1}^{n} r_i^s {\xi_i}^s - \sum_{i=n+1}^{n+d} r_i \xi_i - \sum_{i=n+1}^{n+d} r_i^* {\xi_i}^* \\
&\quad - \sum_{i=1}^{n} \alpha_i^s (y_i(\mathbf{w}_s^{\mathrm{T}} \varphi(\mathbf{x}_i) + b_s) - 1 + \xi_i^s) \\
&\quad - \sum_{i=n+1}^{n+d} \alpha_i \left(\frac{1}{|S_i|} \sum_{j \in S_i} (\mathbf{w}_t^{\mathrm{T}} \varphi(\mathbf{x}_j) + b_t) - \tilde{y}_i + \varepsilon_i + \xi_i \right) \\
&\quad - \sum_{i=n+1}^{n+d} \alpha_i^* (\tilde{y}_i + \varepsilon_i + \xi_i^* - \frac{1}{|S_i|} \sum_{j \in S_i} (\mathbf{w}_t^{\mathrm{T}} \varphi(\mathbf{x}_j) + b_t)),
\end{aligned} \tag{1.1}
$$

where $\alpha^s = (\alpha_1^s, \ldots, \alpha_n^s), \alpha = (\alpha_1, \ldots, \alpha_d), \alpha^* = (\alpha_1^*, \ldots, \alpha_d^*), \mathbf{r}^s = (r_1, \ldots, r_n), \mathbf{r} = (r_1, \ldots, r_d), \mathbf{r}^* = (r_1^*, \ldots, r_d^*)$ are Lagrange multipliers.

According to the dual theorem, the minimum it with respect to $\mathbf{w}_t$, $\mathbf{w}_s$, b_t, b_s, ξ, ξ^*, ξ^s and maximize with respect to Lagrange multipliers $\alpha_i^s \geq 0, \alpha_i \geq 0, \alpha_i^* \geq 0, r_i^s \geq 0, r_i \geq 0, r_i^* \geq 0$. Then the following equations can be considered as the necessary conditions of the optimal solution:

$$\frac{\partial L}{\partial {\xi_i}^s} = 0 \Rightarrow C_s = {\alpha_i}^s + r_i^s, \tag{1.2}$$

$$\frac{\partial L}{\partial {\xi_i}^{(*)}} = 0 \Rightarrow C_t = {\alpha_i}^{(*)} + r_i^{(*)}, \tag{1.3}$$

$$
\begin{aligned}
\frac{\partial L}{\partial \mathbf{w}_t} = 0 \Rightarrow \mathbf{w}_t + \lambda(\mathbf{w}_t - \mathbf{w}_s) - \sum_{i=n+1}^{n+d} \frac{\alpha_i}{|S_i|} \sum_{j \in S_i} \varphi(\mathbf{x}_j) \\
+ \sum_{i=n+1}^{n+d} \frac{{\alpha_i}^*}{|S_i|} \sum_{j \in S_i} \varphi(\mathbf{x}_j) = 0,
\end{aligned} \tag{1.4}
$$

$$\frac{\partial L}{\partial \mathbf{w}_s} = 0 \Rightarrow \mathbf{w}_s - \lambda(\mathbf{w}_t - \mathbf{w}_s) - \sum_{i=1}^{n} {\alpha_i}^s y_i \varphi(\mathbf{x}_i) = 0, \tag{1.5}$$

$$\frac{\partial L}{\partial b_t} = 0 \Rightarrow \sum_{i=n+1}^{n+d} (\alpha_i - {\alpha_i}^*) = 0, \tag{1.6}$$

$$\frac{\partial L}{\partial b_s} = 0 \Rightarrow \sum_{i=1}^{n} {\alpha_i}^s y_i = 0. \tag{1.7}$$

Substituting Eqs. (1.2)–(1.7) into Eq. (1.1), we have the dual of Eq. (1.8),

$$
\begin{aligned}
&\min_{\alpha^s, \alpha, \alpha^*} \frac{1+\lambda}{2(1+2\lambda)} \left(\sum_{i=n+1}^{n+d} \sum_{j=n+1}^{n+d} \frac{(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)}{|S_i||S_j|} \sum_{i' \in S_i} \sum_{j' = S_j} k(\mathbf{x}_{i'}, \mathbf{x}_{j'}) \right. \\
&\quad \left. + \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i^s \alpha_j^s y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \right) \\
&\quad + \frac{\lambda}{2(1+2\lambda)} \left(\sum_{i=1}^{n} \sum_{j=n+1}^{n+d} \alpha_i^s y_i \frac{(\alpha_j - \alpha_j^*)}{|S_j|} \sum_{k \in S_j} k(\mathbf{x}_i, \mathbf{x}_k) \right.
\end{aligned}
$$

$$+\sum_{i=1}^{n}\sum_{j=n+1}^{n+d}\alpha_i^s y_i\frac{(\alpha_j-\alpha_j^*)}{|S_j|}\sum_{k\in S_j}k(\mathbf{x}_i,\mathbf{x}_k)\Big)$$
$$+\sum_{i=n+1}^{n+d}\alpha_i(\varepsilon_i-\tilde{y}_i)+\sum_{i=n+1}^{n+d}\alpha_i^*(\varepsilon_i+\tilde{y}_i), \tag{1.8}$$

s.t. $\alpha_i^s\in[0,C_s]$, $\alpha_i,\alpha_i^*\in[0,C_t]$, $\sum_{i=1}^{n}\alpha_i^s y_i+\sum_{i=n+1}^{n+d}(\alpha_i-\alpha_i^*)=0$.

To obtain a quadratic form of the optimization problem, we denote

$$\beta=[\alpha^s,\alpha,\alpha^*]^{\mathbf{T}},\quad 0\le\beta\le[\underbrace{C_s,\ldots,C_s}_{n}\underbrace{C_t,\ldots,C_t}_{d}\underbrace{C_t,\ldots,C_t}_{d}],$$

$$\mathbf{f}^{\mathbf{T}}=[y_1,\ldots,y_n,\underbrace{1,\ldots,1}_{d},\underbrace{-1,\ldots,-1}_{d}],\ \tilde{\mathbf{e}}=[\underbrace{0,\ldots,0}_{n},\varepsilon-\mathbf{y},\varepsilon+\mathbf{y}],$$

$$\tilde{\mathbf{K}}=\begin{bmatrix}\frac{1+\lambda}{1+2\lambda}\mathbf{K}_{s,s}+\frac{1}{\lambda} & \frac{\lambda}{1+2\lambda}\mathbf{K}_{s,t} & -\frac{\lambda}{1+2\lambda}\mathbf{K}_{s,t}\\ \frac{\lambda}{1+2\lambda}\mathbf{K}_{s,t}^{\mathbf{T}} & \frac{1+\lambda}{1+2\lambda}\mathbf{K}_{t,t} & -\frac{1+\lambda}{1+2\lambda}\mathbf{K}_{t,t}\\ -\frac{\lambda}{1+2\lambda}\mathbf{K}_{s,t}^{\mathbf{T}} & -\frac{1+\lambda}{1+2\lambda}\mathbf{K}_{t,t} & \frac{1+\lambda}{1+2\lambda}\mathbf{K}_{t,t}\end{bmatrix}_{(n+2d)\times(n+2d)},$$

$$\mathbf{K}_{s,s}=(y_iy_jk(\mathbf{x}_i,\mathbf{x}_j))_{i,j=1,\cdots,n},\ \mathbf{K}_{s,t}=\left(y_i\frac{1}{|S_k|}\sum_{j\in S_k}k(\mathbf{x}_i,\mathbf{x}_j)\right)_{i=1,\cdots,n,k=1,\ldots,d},$$

$$\mathbf{K}_{t,t}=\left(\frac{1}{|S_i||S_j|}\sum_{i'\in S_i}\sum_{j'\in S_j}k(\mathbf{x}_{i'},\mathbf{x}_{j'})\right)_{i,j=1,\ldots,d}$$

and $\{y_i\}_{i=1}^{n}$ are class labels of training samples in the source domain. Eq. (1.8) is equivalent to the following equation

$$\min_{\beta}\frac{1}{2}\beta^{\mathbf{T}}\tilde{\mathbf{K}}\beta+\tilde{\mathbf{e}}^{\mathbf{T}}\beta, \tag{1.9}$$

s.t. $\mathbf{f}^{\mathbf{T}}\beta=0$.

Thus, Theorem 1 is hold.

Appendix 2

Proof of Theorem 2

We denote the matrix $\tilde{\mathbf{K}}$ as $\tilde{\mathbf{K}}=\tilde{\mathbf{K}}_1+\tilde{\mathbf{K}}_2+\tilde{\mathbf{K}}_3$ where

$$\tilde{\mathbf{K}}_1=\frac{\lambda}{1+2\lambda}\begin{bmatrix}\mathbf{K}_{s,s} & \mathbf{K}_{s,t} & -\mathbf{K}_{s,t}\\ \mathbf{K}_{s,t}^{\mathbf{T}} & \mathbf{K}_{t,t} & -\mathbf{K}_{t,t}\\ -\mathbf{K}_{s,t}^{\mathbf{T}} & -\mathbf{K}_{t,t} & \mathbf{K}_{t,t}\end{bmatrix}_{(n+2d)\times(n+2d)},$$

$$\tilde{\mathbf{K}}_2=\frac{1}{1+2\lambda}\begin{bmatrix}\mathbf{K}_{s,s} & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}_{(n+2d)\times(n+2d)},$$

$$\tilde{\mathbf{K}}_3=\frac{1}{1+2\lambda}\begin{bmatrix}0 & 0 & 0\\ 0 & \mathbf{K}_{t,t} & -\mathbf{K}_{t,t}\\ 0 & -\mathbf{K}_{t,t} & \mathbf{K}_{t,t}\end{bmatrix}_{(n+2d)\times(n+2d)}.$$

Set $\mathbf{Q}_1=\sqrt{\frac{\lambda}{1+2\lambda}}(y_1\varphi(\mathbf{x}_1),\ldots,y_n\varphi(\mathbf{x}_n),\frac{1}{|S_1|}\sum_{i\in S_1}\varphi(\mathbf{x}_i),\ldots,\frac{1}{|S_d|}\sum_{i\in S_d}\varphi(\mathbf{x}_i),-\frac{1}{|S_1|}\sum_{i\in S_1}\varphi(\mathbf{x}_i),\ldots,-\frac{1}{|S_d|}\sum_{i\in S_d}\varphi(\mathbf{x}_i))$, then $\tilde{\mathbf{K}}_1=\mathbf{Q}_1{}^{\mathbf{T}}\mathbf{Q}_1$. So the matrix $\tilde{\mathbf{K}}_1$ is positive semi-definite. Similarly, the matrixes $\tilde{\mathbf{K}}_2$ and $\tilde{\mathbf{K}}_3$ can be proved to be positive semi-definite. That is, the matrix $\tilde{\mathbf{K}}$ is symmetric and positive semi-definite. Thus, Theorem 2 is hold.

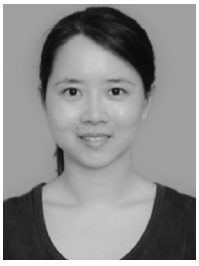
References

[1] N. Quadrianto, A.J. Smola, T.S Caetano, Estimating labels from label proportions, in: Proceedings of Twenty-fifth International Conference on Machine Learning, ICML'08, Omnipress, 2008, pp. 776–783.
[2] N. Quadrianto, A.J. Smola, T.S. Caetano, Estimating labels from label proportions, J. Mach. Learn. Res. 10 (2009) 2349–2374.
[3] S. Rüping, SVM classifier estimation from group probabilities, in: Proceedings of Twenty-seventh International Conference on Machine Learning, ICML'10, Haifa, 2010, pp. 911–918.
[4] M. Stolpe, K. Morik, Learning from label proportions by optimizing cluster model selection, in: Proceedings of the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases, ECML PKDD'2011, Berlin, Heidelberg, 2011, pp. 349–364. Part III.
[5] E.M. Ardehaly, A. Culotta, Domain adaptation for learning from label proportions using self-training, in: Proceedings of Twenty-fifth International Joint Conference on Artificial Intelligence, New York, USA, 2016, pp. 3670–3676.
[6] Z.Q. Qi, B. Wang, F. Meng, L.F. Niu, Learning with label proportions via NPSVM, IEEE Trans. Cybern. 99 (2016) 1–13.
[7] S.J. Pan, Q. Yang, A survey on transfer learning, Trans. Knowl. Data Eng. 22 (10) (2010) 1345–1359.
[8] J. Davis, P. Domingos, Deep transfer via second-order Markov logic, in: Proceedings of the Twenty-sixth Annual International Conference on Machine Learning (ICML'09), ACM, 2009, pp. 217–224.
[9] S.J. Pan, I.W. Tsang, J.T. Kwok, Q. Yang, Domain adaptation via transfer component analysis, IEEE Trans. Neural Netw. 22 (2) (2011) 199–210.
[10] L.X. Duan, D. Xu, I.W. Tsang, Domain adaptation from multiple sources: a domain-dependent regularization approach, IEEE Trans. Neural Netw. Learn. Syst. 23 (3) (2012) 504–518.
[11] L.X. Duan, I.W. Tsang, D. Xu, Domain transfer multiple kernel learning, IEEE Trans. Pattern Anal. Mach. Intell. 34 (3) (2012) 465–479.
[12] Z. Huang, S.M. Siniscalchi, C.H. Lee, A unified approach to transfer learning of deep neural networks with applications to speaker adaptation in automatic speech recognition, Neurocomputing 218 (2016) 448–459.
[13] Y.X. Ma, J.Y. Xu, X.Y. Wu, F. Wang, W. Chen, A visual analytical approach for transfer learning in classification, Inf. Sci. 390 (2017) 54–69.
[14] Q.S. Lian, B.S. Shi, S.Z. Chen, Transfer orthogonal sparsifying transform learning for phase retrieval, Digital Signal Process. 62 (2017) 11–25.
[15] H.B. Yan, Transfer subspace learning for cross-dataset facial expression recognition, Neurocomputing 208 (2016) 165–17.
[16] B. Scholkopf, R. Herbrich, A. Smola, A generalized representer theorem, Lect. Notes Comput. Sci. 2111 (2001) 416–426.
[17] J. Platt, Fast training of support vector machines using sequential minimal optimization, Advances in Kernel Methods-Support Vector Learning, MIT Press, Cambridge, MA, 2000, pp. 185–208.
[18] M. Badoiu, K.L. Clarkson, Optimal core sets for balls, Comput. Geomet. 40 (1) (2002) 14–22.
[19] I. Tsang, J. Kwok, P. Cheung, Core vector machines: Fast SVM training on very large datasets, J. Mach. Learn. Res. 6 (2005) 363–392.
[20] S Wang, J. Wang, F. Chung, Kernel density estimation, kernel methods, and fast learning in large datasets, IEEE Trans. Cybern. 44 (1) (2014) 1–20.
[21] M. Nandan, P.P. Khargonekar, S.S. Talathi, Fast SVM training using approximate extreme points, J. Mach. Learn. Res. 15 (2014) 59–98.
[22] J.C. Platt, Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, Advances in Large Margin Classifiers, MIT Press, Cambridge, 1999, pp. 61–74.
[23] D. Wang, H. Qiao, B. Zhang, M. Wang, Online support vector machine based on convex hull vertices selection, IEEE Trans. Neural Netw. Learn. Syst. 24 (4) (2013) 593–609.
[24] A.L. Chau, X.O. Li, W. Yu, Convex and concave hulls for classification with support vector machine, Neurocomputing 122 (2013) 198–209.
[25] D. Tax, R. Duin, Support vector domain description, Pattern Recognit. Lett. 20 (11) (1999) 1191–1199.
[26] V. Vapnik, Statistical Learning Theory, John Wiley and Sons, 1998.
[27] J. Gao, W. Fan, J. Jiang, J. Han, Knowledge transfer via multiple model local structure mapping, in: Proceedings of the Fourteenth 44 ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2008.
[28] B. Quanz, J. Huan, Large margin transductive transfer learning, in: Proceeding of the Eighteenth ACM Conference on Information and Knowledge Management (CIKM), New York, 2009, pp. 1327–1336.
[29] C.C. Chang, C.J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (3) (2011) 1–27.
[30] Z.H. Deng, K.S. Choi, F.L Chung, S.T. Wang, Scalable TSK fuzzy modeling for very large datasets using minimal enclosing ball approximation, IEEE Trans. Fuzzy Syst. 19 (2) (2011) 210–226.
[31] W.J. Chen, Y.H. Shao, N. Hong, Laplacian smooth twin support vector machine for semi-supervised classification, Int. J. Mach. Learn. Cybern. (2013) 1–10.
[32] C.C. Chang, C.J. Lin, Training ν-support vector classifiers: theory and algorithms, Neural Comput. 13 (9) (2001) 2119–2147.
[33] M.S. Long, J.M. Wang, G.G Ding, J.G. Sun, P.S. Yu, Transfer feature learning with joint distribution adaptation, in: Proceeding of IEEE International Conference on Computer Vision (ICCV), Sydney, 2013, pp. 2200–2207.
[34] X. Gu, F.L. Chung, H. Ishibuchi, S.T. Wang, Multitask coupled logistic regression and its fast implementation for large multitask datasets, IEEE Trans. Cybern. 45 (9) (2015) 1953–1966.
[35] H. Abdi, L.J. Williams, Principal component analysi, Wiley Interdiscip. Rev. Comput. Stat. 2 (4) (2010) 433–459.
[36] M.S. Long, J.M. Wang, G.G Ding, S.J.L. Pan, P.S. Yu, Adaptation regularization: a general framework for transfer learning, IEEE Trans. Know. Data Eng. 26 (5) (2014) 1076–1089.
[37] Z.H. Deng, K.S. Choi, F.L Chung, S.T. Wang, Scalable TSK fuzzy modeling for very large datasets using minimal enclosing ball approximation, IEEE Trans. Fuzzy Syst. 19 (2) (2011) 210–226.

Please cite this article as: T. Ni et al., Scalable transfer support vector machine with group probabilities, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.08.049

[38] A.V. Uzilov, J.M. Keegan, D.H Mathews, Detection of non-coding RNAs on the basis of predicted secondary structure formation free energy change, BMC Bioinf. 173 (7) (2006) 1–30.

Tongguang Ni received his Ph.D. from Jiangnan University in May 2015. He is currently a Lecturer in the School of Information Science and Engineering, Changzhou University, Changzhou, China. His current research interests include pattern recognition, intelligent computation and their application.



Xiaoqing Gu received the M.S. degree in computer science from Jiangnan University, China, in 2006. She is currently pursuing a Ph.D. degree in the School of Digital Media, Jiangnan University, Wuxi, China.

She is currently a Lecturer in the School of Information Science and Engineering, Changzhou University, China. She has published over ten papers in international/national authoritative journals. Her current research interests include pattern recognition and machine learning

Jun Wang received his Ph.D. degree in pattern recognition and intelligence systems from the School of Computer Science and Technology in Nanjing University of Science and Technology, Nanjing (NUST), China, in 2011. He is currently an Associate Professor with the School of Digital Media, Jiangnan University, China, and a research associate in the Department of Radiology and BRIC, School of Medicine, University of North Carolina at Chapel Hill, USA. His research interests include fuzzy clustering, machine learning in Neuroimaging and fMRI analysis.

Yuhui Zheng received the Ph.D. degree in the Nanjing University of Science and Technology, China in 2009. Now, he is an associate professor in the School of Computer and Software, Nanjing University of Information Science and technology. His research interests cover image processing, pattern recognition, and remote sensing information system.

Hongyuan Wang received the master's degree from Zhejiang University, Hangzhou, China, in 2010, and the Ph.D. degree from Nanjing University of Science and Technology, Nanjing, china, in 2014. He is currently a professor in the School of Information Science and Engineering, Changzhou University, Changzhou, China, His current research interests include computer vision, machine learning, and other data science related areas.

Please cite this article as: T. Ni et al., Scalable transfer support vector machine with group probabilities, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.08.049