Neural Processing Letters (2022) 54:4921–4950
https://doi.org/10.1007/s11063-022-10841-6

Multi-Source Selection Transfer Learning with Privacy-Preserving

Weifei Wu[1]

Accepted: 9 April 2022 / Published online: 7 May 2022

Abstract
Transfer learning has ability to create learning task of weakly labeled or unlabeled target domain by using knowledge of source domain to help, which can effectively improve the performance of target learning task. At present, the increased awareness of privacy protection restricts access to data sources and poses new challenges to the development of transfer learning. However, the research on privacy protection in transfer learning is very rare. The existing work mainly uses differential privacy technology and does not consider the distribution difference between data sources, or does not consider the conditional probability distribution of data, which causes negative transfer to harm the effect of algorithm. Therefore, this paper proposes multi-source selection transfer learning algorithm with privacy-preserving MultiSTLP, which is used in scenarios where target domain contains unlabeled data sets with only a small amount of group probability information and multiple source domains with a large number of labeled data sets. Group probability means that the class label of each sample in target data set is unknown, but the probability of each class in a given data group is available, and multiple source domains indicate that there are more than two source domains. The number of data set contains more than two data sets of source domain and one data set of target domain. The algorithm adapts to the marginal probability distribution and conditional probability distribution differences between domains, and can protect the privacy of target data and improve classification accuracy by fusing the idea of multi-source transfer learning and group probability into support vector machine. At the same time, it can select the representative dataset in source domains to improve efficiency relied on speeding up the training process of algorithm. Experimental results on several real datasets show the effectiveness of MultiSTLP, and it also has some advantages compared with the state-of-the-art transfer learning algorithm.

Keywords Multi-source transfer learning · Group probabilities · Privacy-preserving

✉ Weifei Wu
wuweifei@hrbeu.edu.cn

[1] Beijing Institute of Remote Sensing Equipment, Beijing 100084, People's Republic of China

1 Introduction

Machine learning has progressed dramatically over the past two decades, from laboratory curiosity to a practical technology in widespread commercial use [1]. A prominent aspect of machine learning is the ability to deal with large amounts of unorganized information problems by learning models from labeled data in domain, so sufficiently available labeled data is the basis for reliable results from machine learning models. Currently, machine learning has been widely used in computer vision [2], intrusion detection [3], speech emotion recognition [4, 5], natural language processing [6] and text classification [7].

At present, in order to obtain better accuracy and reliability, the existing traditional machine learning models usually need to meet two basic assumptions: there are enough available data samples in training data set; the training and test data come from the same feature space and distribution [8]. However, in practical applications, training and test data usually come from different fields, and it is difficult to ensure that the data distribution is consistent. In addition, labeled data is scarce in some areas. When the distribution changes, the machine learning algorithm needs to re-collect and re-label training data. In many real-world applications, the cost of re-collecting training data and reconstructing model is very expensive, or even impossible [9]. In this case, transfer learning between learning task domains is desirable, the motivation is that people can use the previously learned knowledge to better solve new problems, and the purpose is to use the label information of another related domain (source domain) for building the model of target domain [10–14]. Unlike traditional machine learning algorithms, them assume that training and test data have the same distribution, transfer learning can use knowledge from different distributions of data. In view of the advantages of transfer learning, a lot of research on it has been launched [15–18].

On the other hand, the smooth implementation of transfer learning usually requires source and target domains to directly share the original data, which cannot be met in some cases, especially when it involves some confidential or sensitive data. So, the protection of data privacy in transfer learning is becoming an important issue that people pay attention to. The research on privacy protection in transfer learning is very rare. The most recent related work is the differential privacy hypothesis transfer learning method for logistic regression proposed by Wang et al. [19], which uses the public unlabeled source data set to measure the relationship between source and target domain with the hypothesis trained on source domain to improve the learning of target hypothesis. Other related researches focus on variants of transfer learning, such as iterative differential privacy multi-task learning [20], distributed training data aggregation that considers covariate shift (covariate shift) [21], these works either did not consider the distribution differences between data sources, or did not consider the conditional distribution of data.

Recently, a class of machine learning methods that use information in group probabilities to train classifier provides an effective way to protect data privacy, this is a type of semi-supervised learning method between supervised learning and unsupervised learning [21–24]. As shown in Fig. 1, for a set of training data without class labels, if you only know the probability of belonging to a certain class label in each data group, that is, the group probability, then use the group probability information to obtain a classifier that can effectively label the data. A typical application related to group probability is in political election events, where the number of voters in a constituency is known. In order to protect the privacy of each voter, only the number of candidates votes will be provided, and the specific ballot information for each voter is unknown. It can be seen that the group probability information provides an effective means for protecting the privacy of data.

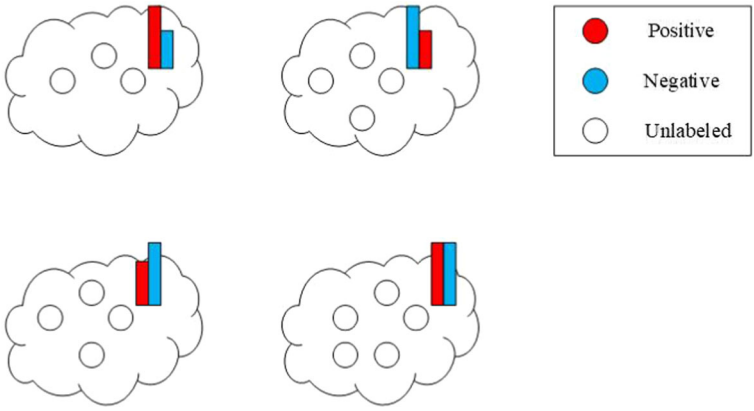

Fig. 1 Leaning from group probability

In view of the advantages of group probability to protect the privacy of training data in data classification, using group probability to solve the privacy protection problem in transfer learning has attracted the attention of researchers [25, 26]. The existing algorithms only consider the knowledge in a source domain to target learning task, and the marginal probability difference. However, there is more than one source domain related to target domain. Therefore, it is natural that many transfer learning algorithms related to multiple source domains are proposed [27–29]. The multi-source transfer method extracts knowledge from data sets of two or more source domains for the learning task of target domain. Compared with the transfer learning method that only uses one source domain, it can increase the chance that transferring relevant knowledge from source domains to target domain and improve learning result. Today, transfer learning has been applied to COVID-19 recognition [30], law article prediction [31], the classification of histological images of colorectal cancer [32], human action recognition [33], cross-domain recommendations [34] and EEG signal analysis [35].

In this paper, we utilize the group probability and multi-source transfer learning theory, in the case of the application scenarios that the target domain has only group probability data with only a small amount of unlabeled data, and multiple source domains contain large amount of labeled data in each source domain, a new multi-source selection transfer learning algorithm with privacy-preserving (MultiSTLP) is proposed. The idea of MultiSTLP algorithm is to use the knowledge of group probability in target domain and the knowledge of labeled data in multiple source domains into the framework of support vector machine structure risk minimization, by constructing the similar distance term between target and each source domains. During the process, considering the marginal and conditional probability differences the knowledge of the representative dataset which is selected form source domains is transferred into target domain, and then the optimizable objective function is constructed. The theoretical proof of the objective function shows that the solution process is a quadratic programming problem with optimal solution. In the algorithm, the group probability protects the privacy of the target data, and the representative dataset of sources domain helps to reduce the size of training samples and improve the efficiency of algorithm training.

Compared with the previous works, the contributions of this paper are:

(1) A new multi-source selection transfer learning algorithm with privacy-preserving MultiSTLP is proposed, which utilizes samples of the representative dataset that is selected

from multiple sources and unlabeled group probability samples in target domain. MultiSTLP not only improves training efficiency, but also protects data privacy. The objective function of MultiSTLP can be transformed into a traditional standard quadratic programming problem and proved to have global optimal values through rigorous mathematical proof.

(2) By reducing the marginal and conditional probability differences, the knowledge of each source domain with similarity to target domain are transferred to the greatest extent, which effectively solves the negative transfer and improves the effect of algorithm. In addition, the representative dataset in source domains can make full use of high-quality samples, reducing the number of training samples and speeding up the algorithm training process.

(3) Extensive experiments have been carried out on real datasets, the experimental results show that the result of MultiSTLP is better than the state-of-art algorithms or at least comparable to them.

The rest of the paper is organized as follows. In Sect. 2, the related works of selective transfer learning support vector machine and group probability are briefly reviewed. The MultiSTLP is proposed in Sect. 3. In Sect. 4, we verify the effectiveness of MultiSTLP on four real-world datasets, and the experimental results are analyzed. The last section summarizes the conclusions of this paper and researches in the future.

2 Brief Review of Related Works

We briefly introduce the selective transfer learning support vector machine and group probabilities in this section. In the group probability introduction, we focus on the IC technology and the IC-SVM algorithm.

First of all, we start with the variable definitions of terminologies. For clarity, Table 1 lists the frequently used notations.

2.1 Selective Transfer Learning Support Vector Machine

For SVM, a lot of training samples is a prerequisite for achieving better training results. This not only requires a lot of manpower to label, but also a lot of time is consumed in the training phase, so the training efficiency of SVM is not satisfactory. In order to improve the efficiency of SVM training, a method of using training samples near the largest hyperplane to train SVM approximate the extreme point support vector machine (AESVM) was proposed in [36]. AESVM no longer needs all training samples to train the learning model. The training sample size can be greatly reduced, so that the training cost of the learning model is reduced.

On this basis, Li et al. [10] proposed a selective transfer learning support vector machine algorithm (STL-SVM), which uses AESVM to select representative dataset from source domain. STL-SVM first utilizes an improved maximum mean discrepancy (MMD) to calculate the weight vector of the importance of the sample in source domain relative to target domain; then AESVM method is applied to select a representative dataset and the weight of samples; finally, combined with the support vector to construct an objective function with the ability the transfer learning.

Given a source domain D_S containing n sample data, $D_S = \{(x_1^S, y_1^S), (x_2^S, y_2^S), \ldots, (x_n^S, y_n^S)\}$, $X_S = \{x_1^S, x_2^S, \ldots, x_n^S\}$, $Y_S = \{y_1^S, y_2^S, \ldots, y_n^S\}$, $Y_S \in \{1 - 1\}$. Similarly,for

Table 1 Notations and descriptions

Notation	Description
D_S/D_T	Source/target domain
D_{S_i}	S_i-th Source domain
X_S/X_T	Source/target sample set
Y_S/Y_T	Source/target class label set
$n_{S_i}{}'$	Number of labeled S_i-th source domain
M	Number of source domain
n	Number of target domain samples
w_s/b_s	Parameters of source linear classifier
w_t/b_t	Parameters of target linear classifier
p_k	Group probability
G_k	k-th Groups
K	Number of group
γ	Weight of source domains
β^{S_i}	Weight of representative data set in S_i-th source domain
v	Weight of samples

a target domain $D_T = \{(x_1^T, y_1^T), (x_2^T, y_2^T), \ldots, (x_m^T, y_m^T)\}$ with m samples, $X_T = \{x_1^T, x_2^T, \ldots, x_m^T\}$. The objective function of STL-SVM is shown in Eq. (1):

$$\begin{cases} \min\limits_{w_t, b_t} \frac{1}{2}||w_t||^2 + C_t \sum_{i=n+1}^{n+m} \xi_i^t + \frac{1}{2}||w_s||^2 + \frac{C_s}{n} \sum_{i=1}^{n} \beta_i \xi_i^s \\ \quad + \frac{\lambda}{2}||w_t - w_s||^2 \\ s.t.\ y_i^s(w_s^T \phi(x_i^s) + b_s) \geq 1 - \xi_i^s \\ y_i^t(w_t^T \phi(x_i^t) + b_t - \widetilde{w}_t^T \phi(x_i^t) - \widetilde{b}_t) \geq 0 \\ \xi_i^s \geq 0, \quad i = 1, 2 \ldots n \end{cases} \tag{1}$$

In Eq. (1), w_t and b_t represent the parameters in target domain, w_s and b_s represent the parameters in source domain, these parameters include knowledge in domains; $\widetilde{w}_t^T$ and $\widetilde{b}_t^T$ represent the knowledge obtained by SVM training only on dataset in target domain; $\phi(\cdot)$ is non-linear mapping function; ξ_i^t ($\xi_i^t \geq 0$) and ξ_i^s ($\xi_i^s \geq 0$) are the slack variables in target and source domains, respectively; n is the number of samples in source domain, n' is the number of samples in representative dataset calculated by AESVM; m is the number of samples in target domain; $\beta_i \in [\beta_1, \beta_2, \ldots, \beta_M]$ represents the weight value corresponding to each sample in representative data set; C_t ($C_t \geq 0$) and C_s ($C_s \geq 0$) are the degree of penalty error of the regularization coefficient in target and source domain, respectively; T represents the transposition of the matrix; $f(x_i) = \widetilde{w}_t^T \phi(x_i^t) + \widetilde{b}_t$ is the decision function of SVM classifier in target domain.

Solve the Eq. (1) to obtain the model parameters and, substitute them into the decision function Eq. (2) of STL-SVM:

$$f(x) = w_t^T \phi(x) + b_t \tag{2}$$

On the one hand, STL-SVM reduces the size of training samples in source domain through AESVM and accelerates the learning progress; on the other hand, it uses MMD and objective function construction principles to effectively solve the negative transfer problem that is easy to occur in transfer learning. Therefore, STL-SVM completes the knowledge transfer by

effectively fusing the knowledge of the source and target domains, so as to obtain a better classification effect. Experiments on artificial and real datasets show the effectiveness of STL-SVM. Compared with previous research work on transfer learning algorithms, the STL-SVM algorithm can better solve the negative transfer situation and the long training time of the classifier due to too many training samples in source domain. However, the problem with STL-SVM is that it only considers the marginal probability between domains and does not discuss the conditional probability; it only uses the knowledge in one source domain, when there are multiple source domains related to the target domain that will cause a waste of data resources. In addition, it does not have the ability to protect privacy of target data.

2.2 Group Probabilities

Given a dataset $\mathbf{X} = \{x_i, i = 1, \ldots, N\}$, x_i is i-th sample, N represents the number of samples, and the class label of samples is unknown. The group probability is defined as follows:

Assuming that the dataset $\mathbf{X}$ is divided into K groups, $G_k = \{X_{i,k}, i = 1, \ldots, N_k, k = 1 \ldots, K\}$, N_k represents the total number of samples in each group, and the group probability of each group G_k is known as p_k, which represents the probability that the sample is positive class in the group. In each group, we know the probability that the sample is a positive class, however the class label of each sample is unknown. p_k is called the group probability of dataset $\mathbf{X}$, which can effectively protect the privacy of dataset $\mathbf{X}$.

For the purpose of solving the difficulty of applying the traditional classification model directly to group probability, IC-SVM [22] first labels the group-probability data based on platt model of the inverse calibration technique (IC), then uses these labeled data to train the SVM. IC-SVM utilizes the sigmod function as an estimated SVM posterior probability output method:

$$p(y = 1|x) = 1/(1 + \exp(-Af(x) + B)) \tag{3}$$

In Eq. (3), the parameters A and B are obtained by the minimum cross entropy, x is the sample feature vector, y is a class label, $p(y = 1|x)$ indicates the probability that the sample is positive. Setting A $= 1$ and B $= 0$, the Eq. (3) can be converted into the following Eq. (4):

$$p = \sigma(y) = \frac{1}{1 + \exp(-y)} \tag{4}$$

Further deformation is as follows:

$$y = \sigma^{-1}(p) = -\log\left(\frac{1}{p} - 1\right) \tag{5}$$

In practice, it is difficult to obtain the class label of each sample, so the average values of the class label estimated in each group is approximated as the predicted value of sample, as in Eq. (6):

$$\forall i : \frac{1}{|G_i|} \sum_{j \in G_i} (w^T x_j + b) = \widetilde{y}_i \tag{6}$$

In Eq. (6), $|G_i|$ is number of group G_i, w and b are the parameters in classification hyperplane of SVM, which sets up a bridge between the group probability information and SVM. The optimization problem of the IC combined with the SVM theory can be expressed as follows:

$$\min_{w,b} \frac{1}{2}||w||^2 + C\sum_{i=1}^{K}(\xi_i + \xi_i^*)$$
$$s.t.\ \forall_i^K : \frac{1}{|G_i|}\sum_{j\in G_i}(w^T x_j + b) \geq \widetilde{y}_i - \varepsilon_i - \xi_i,$$
$$\forall_i^K : \frac{1}{|G_i|}\sum_{j\in G_i}(w^T x_j + b) \leq \widetilde{y}_i + \varepsilon_i + \xi_i^*,$$
$$\xi_i \geq 0, \xi_i^* \geq 0 \tag{7}$$

In Eq. (7), K denotes the number of group, ε_i is defined minimum required precision of the estimate. Equation (7) uses SVM classifier to conveniently process group probability information, which also provides theoretical support for the proposed multi-source transfer learning algorithm.

3 Implementation of MultiSTLP

This section describes the multi-source transfer algorithm with group probability in detail. The algorithm framework is shown in Fig. 2. As shown in Fig. 2, the input information of MultiSTLP framework consists of two parts: label samples in target domains which contains unlabeled samples with only group probability information. For convenience, we only consider the binary classification problem (Fig. 3).

M source domains are defined as: $D_S = \{D_{S_i} = (x_j^{S_i}, y_j^{S_i})_{j=1}^{n_{S_i}}, i = 1, \ldots, M\}$, $x_j^{S_i}$ denotes j-th sample of S_i-th source domain, the corresponding class label is $y_j^{S_i}$, $n_{S_i}{}'$ is the number of samples in source domain, P_{S_i} means joint distribution. Analogously, target domain is $D_T = (x_i)_{i=1,\ldots,d}$, d is the number of group and joint distribution is P_T. As a class label, $p_k = |\{i \in G_k, y_i = 1\}|/|G_k|$ equals to $P(Y = 1|G_k)$ that is estimating probability of class label. $P_{S_i}(x^{S_i})$ and $P_T(x^t)$, $P_{S_i}(y^{S_i}|x^{S_i})$ and $P_T(y^t|x^t)$ are the marginal probability and conditional probability of source and target domains, respectively. Normally, $P_{S_i}(x^{S_i}) \neq P_T(x^t)$ and $P_{S_i}(y^{S_i}|x^{S_i}) \neq P_T(y^t|x^t)$. MultiSTLP not only considers the differences between source and target domains, but also transferability of source domain by simultaneously reducing the differences between the marginal and conditional probabilities.

3.1 Select Representative Data Set and Adapt the Probability of Partial Difference

Refer to reference [10], using the AESVM method to calculate the representative data set in source domain D_{S_i} and its corresponding weight vector $\beta_j^{S_i} \in [\beta_1^{S_i}, \beta_2^{S_i}, \ldots, \beta_{n_{S_i}}^{S_i}]$, the number of samples in representative data set is.

In order to effectively transfer knowledge from source domain that is similar to target domain, this paper adapts both marginal and conditional probability differences. we use MMD to calculate the weights of samples in source domain S_i on the marginal probability difference.

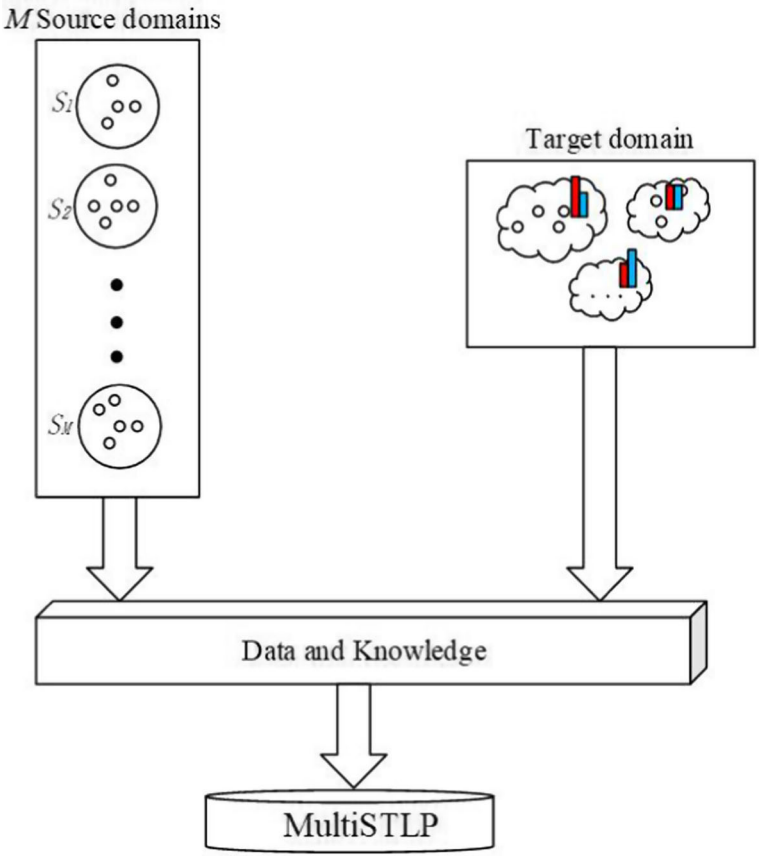

Fig. 2 Framework of MultiSTLP

$$\begin{aligned} &\min_{\alpha^{S_i}} \left\| \frac{1}{n_{S_i}} \sum_{j=1}^{n_{S_i}} v_i^{S_i} \phi(x_j^{S_i}) - \frac{1}{d} \sum_{j=1}^{d} \phi(x_j^t) \right\|_H^2 \\ &s.t.\ v_j^{S_i} \geq 0 \\ &i = 1, \ldots, m, \\ &j = 1, \ldots, n_{S_i} \end{aligned} \tag{8}$$

$\phi(x)$ denotes that feature is mapped to a regenerative kernel Hilbert space H, n_{S_i} is the number of the representative data set in source domain S_i, the number of group in target domain is d, n_{S_i} is also the dimension of v^{S_i}. The minimization problem of Eq. (8) is a standard quadratic programming problem and can be solved using many existing solvers. When constructing the objective function MultiSTLP, by using samples in each source domain we add the corresponding weights as in Eq. (9).

$$v_j^{S_i} = v_j^{S_i} \cdot x_j^{S_i} \cdot \beta_j^{S_i} \tag{9}$$

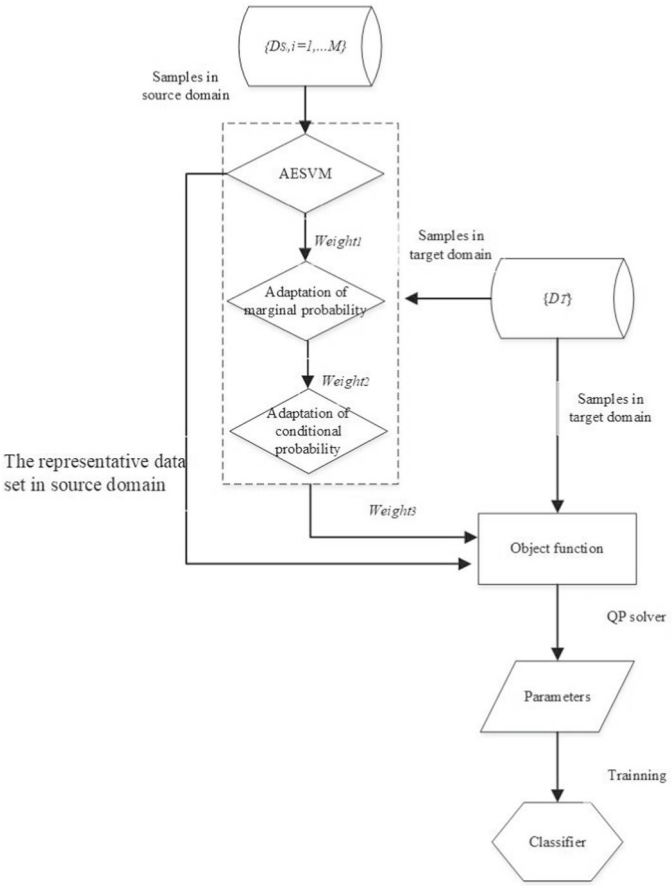

Fig. 3 Flowchart of MultiSTLP

In Eq. (9), v^{S_i} represents the sample vector after weighting samples of source domain S_i. For the convenience of subsequent calculations, set $x^{S_i} = v^{S_i}$, that is, the samples of source domain are weighted with v^{S_i}.

On the basis of the above calculation of marginal probability difference, calculate γ^{S_i} of source domain D_{S_i}, which reflects similarity between source and target domains. First, we learn the classifier $h^{S_i} : x \to y$, this ensures that the classifier learned on source domain with similar marginal probability distributions. Then, using h^{S_i} predicts unlabeled samples in target domain. $H^S = [h^{S_1}, \ldots, h^{S_M}]$ denotes M classifiers, $\gamma^S = [\gamma^{S_1}, \ldots, \gamma^{S_M}]^T$ is corresponding weight vector. Therefore, the goal of Eq. (10) is to find the optimal weight by minimizing the difference in prediction labels between two neighboring points in target domain.

$$\min_{\gamma:\gamma' e=1, \gamma \geq 0} \sum_{i,j=1}^{d} (H_i^S \gamma^S - H_j^S \gamma^S)^2 W_{ij} \tag{10}$$

H_i^S is the predicting result of i-th sample using H^S W_{ij} is a similarity parameter between two data samples of target domain. Equation (10) can be rewritten as the form of Eq. (11):

$$\min_{\varphi:\gamma' e=1,\gamma\geq 0} \sum_{i,j=1}^{d} (\gamma^S)^T (H^S)^T L H^S \gamma^S \tag{11}$$

In Eq. (11), $L = D - W$ is the graph Laplacian associated with the data of target domain, W is the similarity matrix, D is the diagonal matrix given by $D_{ii} = \sum_{j=1}^{M} W_{ij}$. The minimization problem of Eq. (11) is also a standard quadratic programming problem, which can be calculated using many existing solvers.

3.2 Construction of Object Function

On the basis of 3.1, we combine the structural risk minimization theory and the similarity distance minimization to construct the objective function of MultiSTLP as follows:

$$\begin{aligned}\min_{f_t f_s\in H_k} & \frac{1}{2M}\sum_{i=1}^{M}||f_{s_i}||^2 + \frac{1}{M}\sum_{i=1}^{M} C_{s_i}\sum_{j=1}^{n_{S_i}} l_{s_i}(f_{s_i}, y_j) + \frac{1}{2}||f_t||^2 \\ & + C_t\sum_{i=1}^{d} l_t(f_t, y_i) + \lambda\frac{1}{2M}\sum_{i=1}^{M} d(f_t, f_{s_i})\end{aligned} \tag{12}$$

f_s is decision function vector of M source domains, f_t is decision function in target domain. $||f_{s_i}||^2$ and $||f_t||^2$ are the structure risk terms controlling the complexity of the classifier in the source domain and the target domain, respectively. $||f||^2$ indicates L2-norm. C_{s_i} and C_t are the regularization coefficients in source domain S_i and target domain. $l()$ is convex non-negative loss function. $d()$ is used to quantify the diversity between source and target domains. λ is the trade-off parameter.

Equation (12) consists of three items, the first term $(\frac{1}{2M}\sum_{i=1}^{M}||f_{s_i}||^2 + \frac{1}{M}\sum_{i=1}^{M} C_{s_i}\sum_{j=1}^{n_{S_i}} l_{s_i}(f_{s_i}, y_j))$ refers to the knowledge learning from source domains. The second term $(\frac{1}{2}||f_t||^2 + C_t\sum_{i=1}^{d} l_t(f_t, y_i))$ denotes the knowledge learning from target domain. The third term $(\lambda\frac{1}{2M}\sum_{i=1}^{M} d(f_t, f_{s_i}))$ is that guarantees good generalization performance by minimizing the differences between each source and target domains.

In further, $\frac{\lambda}{2M}\sum_{i=1}^{M}||\mathbf{w}_t - \gamma^{S_i}\mathbf{w}_{s_i}||^2$ is utilized to quantify the diversity between domains. So, Eq. (12) can be rewritten into Eq. (13).

$$\begin{aligned}\min_{w_t,b_t,w_s,b_s} & \frac{1}{2}||\mathbf{w}_t||^2 + C_t\sum_{i=1}^{d} l_t(\mathbf{w}_t^T\varphi(\mathbf{x}) + b_t, y_i) \\ & + \frac{1}{M}\sum_{i=1}^{M}||w_{s_i}||^2 + \frac{1}{M}\sum_{i=1}^{M} C_{s_i}\sum_{j=1}^{n_{S_i}} l_{s_i}(\mathbf{w}_{s_i}^T\varphi(\mathbf{x}) + b_{s_i}, y_j) \\ & + \frac{\lambda}{2M}\sum_{i=1}^{M}||w_t - \gamma^{S_i} w_{s_i}||^2\end{aligned} \tag{13}$$

In Eq. (13), we chose two different hinge loss functions in source and target domains: $l_s(f(x_i), y_i) = \max\{0, 1 - y_i f(x_i)\}$ and $l_t(f(x_i), y_i) = \max\{0, |f(x_i) - \widetilde{y}_i| - \varepsilon\}$. Therefore,

Eq. (13) can be formulated as an optimization problem:

$$
\begin{aligned}
&\min_{w_t,b_t,w_s,b_s} \frac{1}{2}||\mathbf{w}_t||^2 + C_t \sum_{i=1+\sum_j^M n_{S_j}}^{\sum_j^M n_{S_j}+d} (\xi_i+\xi_i^*) \\
&\quad + \frac{1}{2M}\sum_{i=1}^{M}||\mathbf{w}_{s_i}||^2 + \frac{1}{2M}\sum_{i=1}^{M} C_{s_i} \sum_{j=1}^{n_{S_i}} \xi_j^{s_i} \\
&\quad + \frac{\lambda}{2M}\sum_{i=1}^{M}||\mathbf{w}_t - \gamma^{S_i}\mathbf{w}_{s_i}||^2 \\
&s.t. \\
&y_j^{S_i}(\mathbf{w}_{s_i}^T\varphi(\mathbf{x}_j^{S_i}) + b_{s_i}) \geq 1 - \xi_j^{s_i},\ j = 1,\ldots,n_{S_i}\ S_i = 1,\ldots,M \\
&\forall_{i=1}^{d} : \frac{1}{|G_i|}\sum_{j\in G_i}(\mathbf{w}_t^T\varphi(\mathbf{x}_j) + b_t)) \geq \widetilde{y}_i - \varepsilon_i - \xi_i, \\
&\forall_{i=1}^{d} : \frac{1}{|G_i|}\sum_{j\in G_i}(\mathbf{w}_t^T\varphi(\mathbf{x}_j) + b_t)) \leq \widetilde{y}_i + \varepsilon_i + \xi_i^*
\end{aligned} \tag{14}
$$

In Eq. (14), $\xi_j^{s_i}$, ξ_i and ξ_i^* are slack variables; the first constraint guarantees that each source domain is classified as accurately as possible; the second and three constraints control estimating class probability of G_i in target domain to approximate p_i. ε_i is the estimated minimum precision of $\widetilde{y}_i$ that satisfies the following function:

$$
p_i - \varepsilon \leq \frac{1}{1+\exp(-\widetilde{y}_i)} \leq p_i + \varepsilon \tag{15}
$$

According to [22], ε_i is set to be $\varepsilon_i = \frac{\tau}{p_i(1-p_i)}$, p_i is the group probability $P(Y=1|G_k)$, ε is a very small positive constant.

3.3 Theorems Related to the Objective Function

Theorem 1 *The dual problem of Eq.* (14) *is a QP problem as shown in Eq.* (16).

$$
\begin{aligned}
&\min_{\beta} \frac{1}{2}\beta^T\widetilde{\mathbf{K}}\beta + \widetilde{\mathbf{e}}^T\beta \\
&s.t.\ \mathrm{f}^T\beta = 0 \\
&\beta = [\alpha^{s_1}, \alpha^{s_2}, \ldots, \alpha^{s_M}, \alpha, \alpha^*]^{\mathrm{T}}, \\
&0 \leq \beta \leq \Big[\underbrace{C_{s_1},\ldots,C_{s_1}}_{n_{S_1}},\ldots,\underbrace{C_{s_M},\ldots,C_{s_M}}_{n_{S_M}},\underbrace{C_t,\ldots,C_t}_{d},\underbrace{C_t,\ldots,C_t}_{d}\Big], \\
&\mathbf{f}^{\mathrm{T}} = \Big[y_1^{S_1},\ldots,y_{n_{S_1}}^{S_1},\ldots,y_1^{S_M},\ldots,y_{n_{S_M}}^{S_M},\underbrace{1,\ldots,1}_{d},\underbrace{-1,\ldots,-1}_{d}\Big],
\end{aligned}
$$

$$\widetilde{\mathbf{e}} = \left[\underbrace{0, \ldots, 0}_{n_{S_1}}, \ldots, \underbrace{0, \ldots, 0}_{n_{S_M}}, \varepsilon - \widetilde{y}, \varepsilon + \widetilde{y} \right]$$

$$\widetilde{\mathbf{K}} = \begin{bmatrix} \frac{M+\lambda}{1+2\lambda M}\mathbf{K}_{s_1,s_1} + \frac{\lambda}{M}, \ldots, \frac{M+\lambda}{1+2\lambda M}\mathbf{K}_{s_1,s_M} + \frac{\lambda}{M}, \frac{\lambda}{1+2\lambda M}\mathbf{K}_{s_1,t}, -\frac{\lambda}{1+2\lambda M}\mathbf{K}_{s_1,t} \\ \cdots \\ \frac{M+\lambda}{1+2\lambda M}\mathbf{K}_{s_M,s_1} + \frac{\lambda}{M}, \ldots, \frac{M+\lambda}{1+2\lambda M}\mathbf{K}_{s_M,s_M} + \frac{\lambda}{M}, \frac{\lambda}{1+2\lambda M}\mathbf{K}_{s_M,t}, -\frac{\lambda}{1+2\lambda M}\mathbf{K}_{s_M,t} \\ \frac{\lambda}{1+2\lambda M}\mathbf{K}^T_{s_1,t}, \ldots, \frac{\lambda}{1+2\lambda M}\mathbf{K}^T_{s_M,t}, \frac{M+\lambda}{1+2\lambda M}\mathbf{K}_{t,t} - \frac{\lambda}{1+2\lambda M}\mathbf{K}_{t,t} \\ -\frac{\lambda}{1+2\lambda M}\mathbf{K}^T_{s_1,t}, \ldots, -\frac{\lambda}{1+2\lambda M}\mathbf{K}^T_{s_M,t}, -\frac{M+\lambda}{1+2\lambda M}\mathbf{K}_{t,t}, \frac{M+\lambda}{1+2\lambda M}\mathbf{K}_{t,t} \end{bmatrix}_{\left(\sum_{i\in M} n_{S_i}+2d\right)\times\left(\sum_{i\in M} n_{S_i}+2d\right)}$$

$$\begin{aligned} \mathbf{K}_{s_i,s_i} &= (y_j^{S_i} y_q^{S_i} k(x_j^{S_i} x_q^{S_i}))_{j,q=1,2,\ldots,n_{S_i}} \\ \mathbf{K}_{s_i,t} &= \left(\frac{\widetilde{y}_j^{S_i}}{|G_k|} \sum\nolimits_{q\in G_k} k(\mathbf{x}_j, \mathbf{x}_q) \right)_{j=1,\ldots,n_{S_i}, q=1,\ldots,d}, \\ \mathbf{K}_{t,t} &= \left(\frac{1}{|G_i||G_j|} \sum\nolimits_{i'\in G_i} \sum\nolimits_{j'=G_j} k(\mathbf{x}_{i'}, \mathbf{x}_{j'}) \right)_{i,j=1,\ldots,d}. \end{aligned} \tag{16}$$

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

Theorem 2 *The quadratic form of the optimization problem of Eq.* (16) *is a standard convex quadratic programming problem.*

The proof of Theorem 1 can be seen in "Appendix 2".

It is clear from the above results that the optimization problem in Eq. (16) for training can be transformed into a convex QP problem and can be directly solved by the traditional SVM solutions. Simultaneously, Eq. (16) is a convex quadratic programming problem, the KKT condition is also a sufficient condition, and thus the obtained solution is the optimal solution.

According to the results obtained by Eq. (16), the results of the optimal solution are as follows:

$$\begin{aligned} \mathbf{w}_t^* = & \frac{\lambda M}{1+2\lambda M} \sum_{i=1}^{M} \sum_{j=1}^{n_{S_i}} \widetilde{\alpha}_j^{s_i} \gamma^{S_i} y_j^{S_i} \varphi(x_j^{S_i}) \\ & + \frac{M+\lambda}{1+2\lambda M} \sum_{i=1+\sum_{j=1}^{M} n_{S_j}}^{\sum_{j=1}^{M} n_{S_j}+d} \frac{\widetilde{\alpha}_i - \widetilde{\alpha}_i^*}{|G_i|} \sum_{j\in G_i} \varphi(x_j) \end{aligned} \tag{17}$$

$$\begin{aligned} b_t^* = & y_i - \frac{\lambda M}{1+2\lambda M} \sum_{i'=1}^{M} \gamma^{S_{i'}} \sum_{j=1}^{n_{S_i}} \frac{\widetilde{\alpha}_j^{s_j} y_j}{|G_i|} \sum_{q\in G_i} k(x_j, x_q) \\ & - \frac{\lambda + M}{1+2\lambda M} \sum_{i'=1}^{M} \gamma^{S_{i'}} \sum_{j=1+\sum_{l\in M} n_{S_l}}^{\sum_{l\in M} n_{S_l}+d} \frac{\widetilde{\alpha}_j - \widetilde{\alpha}_j^*}{|G_j||G_i|} \sum_{j'\in G_j} \sum_{q\in G_i} k(x_{j'}, x_q) \end{aligned} \tag{18}$$

Finally, the decision function of MultiSTLP is expressed as follows:

$$f(x) = \mathbf{w}_t \varphi(x) + b_t \tag{19}$$

As we can see from Eqs. (17) and (18), the results contain both information of M source and target domains: such as $\mathbf{w}_t^*$, $\frac{\lambda M}{1+2\lambda M} \sum_{i=1}^{M} \sum_{j=1}^{n_{S_i}} \widetilde{\alpha}_j^{s_i} \gamma^{S_i} y_j^{S_i} \varphi(x_j^{S_i})$ is the knowl-

Table 2 MultiSTLP algorithm training

Steps of MultiSTLP Algorithm
Input: Labeled source domains $D_S = \{D_{S_i} = (x_j^{S_i}, y_j^{S_i})_{j=1}^{n_{S_i}}, i = 1, \ldots, M\}$, the number of sample in D_{S_i} is n_{S_i}. An unlabeled training dataset $D_T = (x_i)_{i=1,..,d}$ in the target domain, the group probability is $\{(G_k, p_k)\}_{k=1}^{d}$.
Output: $f(x) = w_t^T \phi(x) + b_t$
Step1 Compute the output of Inverse Calibration in the target domain by using Eq. (3);
Step2 Calculate the representative data set and corresponding weight vector according to [10];
Step3 Compute Eq. (8) to obtain weight vector of samples in each source domain;
Step4 Compute Eq. (9) to obtain weighting sample set of each source domain;
Step5 Compute Eq. (11) to get the weight vector of each source domain γ^S=[$\gamma^{S_1}, \gamma^{S_2}, \ldots, \gamma^{S_M}$];
Step6 Compute Lagrange multiplierβ by solving Eq. (16) with a QP solver;
Step7 Obtain the optimal value of w_t by computing Eq. (17);
Step8 Compute Eq. (18) to get the optimal value of b_t;
Step9 Output the decision function in Eq. (19).

edge that is learned from source domains, otherwise the knowledge from target domain is $\frac{M+\lambda}{1+2\lambda M} \sum_{i=1+\sum_{j=1}^{M} n_{S_j}}^{\sum_{j=1}^{M} n_{S_j}+d} \frac{\widetilde{\alpha}_i - \widetilde{\alpha}_i^*}{|G_i|} \sum_{j \in G_i} \varphi(x_j)$.

3.4 Training MultiSTLP

According to Sects. 3.1–3.3, the training process of MultiSTLP is now summarized and described in Table 2.

4 Experimental Results

In this section, for the purpose of testing the generalization performance of MultiSTLP algorithm, we compare MultiSTLP with the benchmark algorithms on four real-world datasets 20-Newsgroups, TRECVID video detection, sentiment analysis, and Email spam. In the experiments, without loss of generality, we only consider the binary classification problem.

4.1 Experimental Environment and Evaluation Criteria

For the fairness of experiments, a 5-fold cross-validation strategy is selected, and we repeat the strategy twice as the final comparison results. In the experiments, we will run 10 times, the average value of classification accuracy, recall, precision, training time with their standard

deviations are recorded. The representation of classification accuracy is as follows:

$$Accuracy = \frac{|\{x|x_t \in D_t \cap f(x_i) = y_t\}|}{|\{x|x_t \in D_t\}|}$$

D_t represents datasets in the target domain, y_t is true tag category, $f(x_t)$ is the result of classifying x_t using the learned classifier.

The recall is expressed as follows:

$$\text{Re}call = \frac{TP}{TP + FN}$$

The precision is:

$$\text{Pr}\,ecision = \frac{TP}{TP + FP}$$

TP represents the number of positive class samples that are accurately classified as positive classes by the classifier; FP is the number of negative class samples that are incorrectly classified as positive classes; and FN is the number that indicates that the positive class samples are incorrectly classified as negative classes.

For each algorithm, a Gaussian kernel function is selected in the form $k(x_i, x_j) = \exp(-||x_i - x_j||)/2\sigma^2$. The parameters C_t, C_s and λ of the proposed MultiSTLP are determined by searching the grid $\{10^{-4}10^{-3}10^{-2}, 10^{-1}, 10, 10^1, 10^2, 10^3, 10^4\}$. For baseline algorithms, the default parameter settings in their literatures are adopted in our experiments. The hardware setting of all experiments are as follows: Intel Core (TM), 3.6 GHz, 8 GB, Windows 10 operating system.

The following state-of-the-art baseline algorithms are selected as the comparison algorithms for MultiSTLP.

(1) TrGNB [25] integrates the transfer learning and group probability information into naive Bayesian framework, the knowledge transfer is completed in the process of solving the naive Bayes by using the maximum posterior probability method. Compared with TrGNB, the advantages of our model are as follows: the marginal and conditional probability between source and target domains are considered; the knowledge in more than one source domains is transferred.
(2) IC-SVM [22] is based on the framework of traditional SVM classifier, combined with Inverse Calibration technology (IC) to construct an optimization function of classifier for class labels, which have no the ability of privacy protection and transfer knowledge compared with MultiSTLP.
(3) ARTL [17] learns the adaptive classifier by simultaneously optimizing the structural risk function, the joint distribution matching between domains and the manifold consistency behind the marginal distribution. The differences from the proposed method is that only one source domain can be used.
(4) STL-SVM [10] compared with the proposed method, which have no the ability of privacy protection and transfer knowledge of multi-source.
(5) TSVM-GP [26] integrates transfer term and group probability information into a support vector machine (SVM) to improve the classification accuracy. This method considers only a single source domain and marginal probability compared to the proposed method.
(6) SVM [35] is traditional support vector model, which has no ability to learn across domains.

Table 3 The statistics of 20-Newsgroups

Domains	Comments	Training	Testing	Positive (%)	Feature
Books (B)	6465	2000	4465	50	30,000
DVDs (D)	5586	2000	3586	50	30,000
Electronics (E)	7681	2000	5681	50	30,000
Kitchen (K)	7945	2000	5945	50	30,000

4.2 Datasets

20-Newsgroups [8], TRECVID 2005 [24], Sentiment analysis [20], and Email spam [21] are commonly used in transfer learning applications, so all experiments in this paper are performed on these datasets.

(1) 20-Newsgroups

The 20-Newsgroups dataset are divided into 4 top categories: comp, rec, sci and talk, which contains about 20,000 documents, and each top category can also be divided into four subcategories with detailed information as shown in Table 3. According to the construction of task group in [8, 17], two top categories are randomly selected from top categories, one of which is positive class and the other is negative class. Each task group is specifically: comp vs rec, comp vs sci, comp vs talk, rec vs sci, rec vs talk, and sci vs talk.

(2) TRECVID 2005

TRCVID 2005 contains approximately 86 h of video programs and consists of 74,523 video shots. Each shot is represented by a video frame as a keyframe, and each keyframe is depicted by a 273-dimensional feature vector. All shots are manually labeled with 39 semantic categories. These semantics cover a variety of types, including outdoor scenes, indoor scenes, news types, and generally common objects. TRCVID video are from CNN_ENG, NBC_ENG, MSNBC_ENG, CCTV_CHN, NTDTV_CHN and LBC_ARB 6 channels, 13 news programs. Each channel represents a domain, except LBC containing 3 news programs, the other channels contain 2 news programs. The source domains datasets are selected from 3 English channels and 2 Chinese channels, and the target domain dataset is selected from the Arabic dataset.

(3) Sentiment analysis

The sentiment analysis dataset contains four different comments of Amazon products: books, DVDs, electronics, and kitchen, which represent four domains Books (B), DVDs (D), Electronics (E), and Kitchen (K). Each comment contains product name, comment title, date, location, and comment content. We will evaluate the product with a rating of 3 stars (0–5 stars) or more as a positive example, a product with a rating of less 3 stars as a negative example, and discard if a fuzzy evaluation is found. In the every domain, there are 2000 labeled instances, and about 4000 unlabeled instances, where the number of positive and negative instances is substantially the same. The dataset details are shown in Table 4.

(4) Email spam Email spam dataset was released by ECML/PKDD 2006,see Table 5 for details. It contains a set of 4000 publicly available labeled emails ($U4$) as well as three email sets (*each has* 2500 *emails*) annotated by three different users ($U1$, $U2$ *and* $U3$).

Table 4 The statistics of sentiment analysis dataset

Domains	Comments	Training	Testing	Positive (%)	Feature
Books (B)	6465	2000	4465	50	30,000
DVDs (D)	5586	2000	3586	50	30,000
Electronics (E)	7681	2000	5681	50	30,000
Kitchen (K)	7945	2000	5945	50	30,000

Table 5 The statistics of spam dataset

Domain	Number	Positive	Negative	Feature
U1	4000	2000	2000	206,908
U2	2500	1250	1250	206,908
U3	2500	1250	1250	206,908
U4	2500	1250	1250	206,908

Table 6 Description of source and target domains on TRECVID dataset

Domain	Source domains					Target domain
Channel	CNN_ENG	MSNBS_ENG	NBC_ENG	CCTV_CHN	NTDTV_CHN	LBC_ARB
# Keyframes	11,025	8905	9322	10,896	6481	15,272

Therefore, the data distributions of the three user-annotated email sets and the publicly available email set are different from each other, in which one half of the emails are non-spam and the other half are spam. Since the spam and non-spam in four email subsets have been differentiated, their distributions are relevant but different.

4.3 Analysis of Experimental Results

In this section, the experimental results of MultiSTLP and six benchmark algorithms on real datasets are analyzed and compared.

TRECVID 2005 dataset We utilize two Chinese channels CCTV_CHN(CC), three English channels CNN_ENG(CN), NBC_ENG(NB), MSNBC_ENG(MS) and NTDTV _CHN(NT) as the source domains, and LBC_ARB(L) as the target domain. The details are shown in Table 6. The four transfer learning methods in the benchmarks can only use one source domain, so in the experiment, one of these source domains is randomly selected as the training dataset, and the MultiSTLP algorithm uses the datasets in all source domains simultaneously.

20-Newsgroups dataset: in the experiment, we constructed 3 source domains and one target domain. The details are shown in Table 7. For the four single source domain transfer learning algorithms TrGNB, ARTL, STL-SVM and TSVM-GP randomly select one of the source domains for training, and MultiSTLP can simultaneously use the datasets of the three source domains for training.

Table 7 Description of source and target domains on 20-Newsgroups dataset

Domain	Source domains	Target domain
rec vs sci(r vs s)	rec.autos & sci.crypt	rec.sport.hockey & sci.space
	rec.motorcycles & sci.electronics	
	rec.sport.baseball & sci.med	
com vs sci(c vs s)	comp.graphics & rec.autos	comp.sys.mac.hardware & rec.sport.hockey
	comp.os.ms-windows.misc & rec.motorcycles	
	comp.sys.ibm.pc.hardware & rec.sport.baseball	
sci vs com(s vs c)	sci.crypt & comp.graphics	sci.space & comp.sys.mac.hardware
	sci.electronics & comp.os.ms-windows.misc	
	sci.med & comp.sys.ibm.pc.hardware	

Table 8 Description of source and target domains on Sentiment analysis datase

Domain	Source domains			Target domain
Sentiment dataset	Books	DVDs	Electronics (E)	Kitchen (K)
# Sentiment	6465	5586	7681	7945

Table 9 Description of source and target domains on email spam dataset

Domain	Source domains			Target domain
Emails dataset	U1	U2	U3	U4
#emails	2500	2500	2500	2500

Sentiment analysis dataset: in this dataset, we use Books, DVDs, and Electronics to construct three source domains, and Kitchen as the target domain. The details of source and target domains are shown in Table 8.

Email Spam dataset: for the MultiSTLP algorithm, three personal email datasets are used as three source domains, and the public email dataset is used as target domain; the other four single source domain transfer learning algorithms randomly select one of the three personal email. The detailed information is shown in Table 9.

Tables 10, 11, 12 and 13 show the average classification accuracies, average recall and average precision with their standard deviations of all the benchmarking classifiers on different transfer learning tasks. From these results, we can draw the following conclusions:

(1) In terms of the average classification accuracy, it can be seen from Table 10 that the transfer learning algorithms TrGNB, ARTL, STL-SVM, TSVM-GP and MultiSTLP have better classification results than the non-transfer learning algorithms SVM and IC-SVM. This is because only a small amount of data set with probability information in target domain is not enough to train a reliable learning model, and the transfer learning algorithms can use the knowledge in a large amount of labeled data in source domain to assist target domain to create classification task, so the trained model is better.

In addition, in the transfer learning algorithms TrGNB, ARTL, STL-SVM and TSVM-GP only transfer the knowledge in one source domain, and only consider the marginal probability difference between data between domains, without considering the conditional probability difference. On the one hand, this resulted in insufficient knowledge to be transferred. On the other hand, the large difference between the transferred knowledge and the data in target domain resulted in a negative transfer phenomenon, which harmed the learning effect.

The MultiSTLP algorithm proposed in this paper makes up for the above problems. It not only transfers the knowledge of multiple source domain, but also adapts the marginal probability and conditional probability, and the classification effect is also better. Therefore, the average accuracy of the MultiSTLP algorithm proposed in this article on the four data sets 20-Newsgroups, TRECVID 2005, sentiment analysis and email spam is better than the comparison algorithms, which are 92.45%, 91.16%, 89.25% and 95.05%, respectively.

(2) The average recall in Table 11 show that MultiSTLP has certain advantages compared with non-transfer learning algorithms (SVM, IC-SVM) and single source transfer learning algorithms (TrGNB, ARTL, STL-SVM and TSVM-GP) on all transfer learning tasks.
(3) The average precision in Table 12, it is can be seen that MultiSTLP is better than benchmark algorithms. On the four data sets 20-Newsgroups, TRECVID 2005, senti-

Table 10 Comparison of average classification accuracy with standard deviation on real-world four transfer datasets

Datasets		SVM	IC-SVM	TrGNB	ARTL	STL-SVM	TSVM-GP	MultiSTLP
20-Newsgroups	r vs s	80.14(2.81)	81.66(2.82)	91.45(1.32)	86.35(1.55)	88.87(1.47)	90.67(1.56)	**94.12(1.23)**
	c vs s	74.24(1.98)	79.24(2.01)	92.28(1.11)	82.65(1.32)	81.86(1.36)	88.82(1.45)	**95.32(1.11)**
	s vs c	75.35(2.48)	77.73(2.25)	90.11(1.56)	85.38(1.48)	86.15(1.55)	87.97(1.58)	**93.22(1.01)**
Average		76.58(2.42)	79.54(2.36)	91.48(1.33)	84.79(1.45)	85.63(1.46)	89.15(1.53)	**92.45(1.30)**
TRECVID 2005	CN vs L	79.16(2.87)	80.29(2.65)	91.53(1.98)	86.91(2.17)	87.32(2.51)	93.15(1.69)	**94.32(1.52)**
	MS vs L	77.87(2.13)	78.26(2.27)	89.35(1.89)	83.88(2.12)	84.62(2.42)	90.55(1.76)	**92.57(1.64)**
	NB vs L	73.97(2.49)	74.65(2.59)	86.53(1.85)	80.27(1.96)	82.74(2.01)	88.71(1.81)	**89.76(1.59)**
	CC vs L	74.53(2.01)	75.72(1.96)	87.62(1.76)	81.32(2.78)	83.79(1.81)	90.95(1.53)	**91.54(1.44)**
	NT vs L	69.69(1.93)	70.66(1.87)	84.81(1.65)	78.58(1.76)	79.85(1.73)	86.98(1.58)	**87.62(1.32)**
Average		75.04(2.29)	75.91(2.27)	87.97(1.83)	82.19(2.16)	83.66(2.10)	90.27(1.68)	**91.16(1.50)**
Sentiment analysis	B vs K	78.46(1.77)	79.37(1.83)	90.21(1.87)	88.75(2.01)	91.85(1.93)	89.87(1.71)	**92.53(1.64)**
	D vs K	75.18(1.69)	77.25(1.72)	86.37(1.75)	84.66(2.46)	87.95(1.98)	84.54(1.68)	**88.65(1.55)**
	E vs K	77.59(2.11)	78.27(2.05)	84.56(1.98)	80.93(2.15)	85.11(2.26)	81.75(1.85)	**86.58(1.76)**
Average		77.08(1.86)	72.30(1.87)	87.05(1.87)	84.78(2.21)	88.30(2.06)	85.39(1.75)	**89.25(1.65)**

Bold represents the results of algorithm proposed in this paper

Table 11 Comparison of average recall with standard deviation on four real-world transfer datasets

Datasets		SVM	IC-SVM	TrGNB	ARTL	STL-SVM	TSVM-GP	MultiSTLP
20-Newsgroups	r vs s	70.87(2.73)	71.22(2.61)	75.12(2.42)	74.24(2.54)	71.66(3.43)	76.01(2.31)	**78.54(2.15)**
	c vs s	62.52(3.32)	63.68(3.26)	70.43(2.65)	69.12(2.87)	68.12(3.54)	71.11(2.45)	**73.22(2.02)**
	s vs c	74.24(4.05)	75.78(3.97)	72.81(2.73)	71.67(2.95)	72.33(3.65)	73.64(2.65)	**75.66(2.42)**
Average		69.21(3.37)	70.23(3.28)	72.79(2.60)	71.68(2.79)	70.70(3.54)	73.57(2.47)	**75.81(2.20)**
TRECVID 2005	CN vs L	60.34(3.65)	61.32(3.32)	71.33(2.85)	66.15(3.25)	70.26(3.17)	71.45(2.73)	**72.86(2.66)**
	MS vs L	70.27(2.64)	71.12(2.34)	75.45(2.42)	74.23(2.53)	73.52(2.92)	75.46(2.42)	**76.65(2.36)**
	NB vs L	75.68(3.86)	75.97(3.75)	73.57(2.53)	70.57(2.76)	72.62(2.58)	73.73(2.51)	**74.53(2.32)**
	CC vs L	73.53(3.19)	74.15(3.04)	78.23(2.35)	74.15(3.05)	75.36(2.44)	77.28(2.47)	**79.87(2.23)**
	NT vs L	78.65(2.48)	79.26(2.35)	81.89(2.12)	79.85(2.24)	80.14(2.23)	82.57(2.11)	**83.75(2.02)**
Average		71.69(3.16)	72.36(2.96)	76.09(2.45)	72.99(2.77)	74.38(3.25)	76.10(2.45)	**77.53(2.32)**
Sentiment analysis	B vs K	62.45(3.45)	63.98(3.23)	73.23(2.43)	71.65(2.66)	72.49(2.79)	72.98(2.47)	**74.78(2.32)**
	D vs K	61.98(3.11)	62.76(3.08)	70.36(2.03)	70.25(2.19)	68.17(3.11)	70.01(2.12)	**72.63(1.98)**
	E vs K	50.76(3.85)	51.52(3.63)	68.65(2.42)	67.37(2.76)	64.56(2.92)	67.95(2.53)	**70.21(2.25)**
Average		58.40(3.47)	59.42(3.31)	70.75(2.29)	69.76(2.54)	68.41(2.94)	70.31(2.37)	**72.54(2.18)**

Bold represents the results of algorithm proposed in this paper

Table 12 Comparison of average precision with standard deviation on four real-world transfer datasets

Datasets		SVM	IC-SVM	TrGNB	ARTL	STL-SVM	TSVM-GP	MultiSTLP
20-Newsgroups	r vs s	76.87(2.71)	77.18(2.59)	82.66(2.12)	80.12(2.33)	81.61(2.43)	81.96(2.11)	**83.76(2.01)**
	c vs s	69.25(2.26)	70.12(2.31)	83.45(2.25)	78.84(1.78)	77.66(2.85)	82.84(2.65)	**84.94(2.11)**
	s vs c	72.64(3.23)	73.76(3.35)	82.46(2.32)	77.75(2.05)	80.51(2.74)	80.14(2.13)	**83.76(2.02)**
Average		72.92(2.40)	73.69(2.75)	82.86(2.23)	78.90(2.05)	79.93(2.67)	81.65(2.30)	**84.15(2.05)**
TRECVID 2005	CN vs L	65.32(3.12)	66.21(2.92)	77.75(1.45)	73.65(2.02)	75.84(1.96)	76.23(1.53)	**79.57(1.33)**
	MS vs L	74.64(2.55)	75.36(2.53)	83.42(1.31)	80.82(2.11)	83.41(2.13)	82.56(1.36)	**84.74(1.23)**
	NB vs L	71.25(2.68)	72.52(2.43)	79.65(1.26)	77.57(1.99)	78.87(1.97)	80.67(1.37)	**80.36(1.22)**
	CC vs L	78.38(2.56)	79.73(2.45)	86.43(1.42)	83.76(2.15)	83.86(1.92)	85.32(1.55)	**87.21(1.31)**
	NT vs L	84.26(2.44)	85.32(2.36)	89.65(1.15)	85.95(1.93)	85.49(1.81)	86.46(1.25)	**89.82(1.12)**
Average		74.77(2.67)	75.83(2.54)	83.38(1.32)	80.35(2.04)	81.49(1.96)	82.25(1.41)	**84.34(1.24)**
Sentiment analysis	B vs K	65.65(2.21)	66.36(2.32)	76.56(1.32)	73.75(1.68)	74.29(2.11)	75.35(1.38)	**77.75(1.28)**
	D vs K	64.52(2.38)	65.55(2.23)	72.45(1.28)	70.72(2.11)	69.25(1.87)	71.84(1.42)	**73.54(1.22)**
	E vs K	54.88(2.55)	55.48(2.45)	68.72(1.24)	69.18(1.87)	67.68(2.32)	67.27(1.51)	**69.67(1.12)**
Average		61.68(2.38)	62.46(2.33)	72.58(1.28)	71.22(1.89)	70.41(2.10)	71.49(1.47)	**73.65(1.25)**

Bold represents the results of algorithm proposed in this paper

Table 13 Comparison of average training time (s) with standard deviation on four real-world transfer datasets

Datasets		SVM	IC-SVM	TrGNB	ARTL	STL-SVM	TSVM-GP	MultiSTLP
20-Newsgroups	r vs s	1.22(0.14)	0.04(0.07)	3.29(1.13)	8.75(1.32)	9.25(1.28)	3.26(1.14)	**3.21(1.16)**
	c vs s	1.18(0.15)	0.06(0.09)	3.16(1.14)	8.65(1.33)	9.11(1.22)	3.02(1.03)	**2.15(1.17)**
	s vs c	1.46(0.15)	0.08(0.11)	3.38(1.12)	8.84(1.34)	9.36(1.25)	3.47(1.13)	**3.32(1.18)**
TRECVID 2005	CN vs L	24.67(1.22)	1.16(0.82)	86.57(1.35)	97.46(1.98)	100.57(2.11)	86.58(1.43)	**85.75(1.42)**
	MS vs L	20.86(1.19)	1.12(0.78)	82.84(1.33)	93.25(1.89)	94.23(1.96)	80.47(1.39)	**76.74(1.23)**
	NB vs L	21.35(1.21)	1.13(0.81)	84.45(1.33)	94.33(1.92)	96.54(1.98)	82.36(1.41)	**81.27(1.19)**
	CC vs L	23.47(1.25)	1.14(0.81)	85.86(1.34)	95.45(1.97)	98.76(2.08)	84.54(1.42)	**82.53(1.22)**
	NT vs L	19.43(1.17)	1.09(0.77)	81.53(1.32)	91.86(1.88)	92.52(1.94)	79.55(1.38)	**79.28(1.17)**
Sentiment analysis	B vs K	13.34(0.88)	0.43(0.12)	18.57(1.15)	21.37(1.25)	23.46(1.34)	18.44(1.18)	**18.15(1.24)**
	D vs K	13.16(0.92)	0.41(0.13)	18.36(1.13)	20.87(1.28)	21.75(1.32)	18.27(1.16)	**17.86(1.23)**
	E vs K	13.53(1.02)	0.45(0.15)	18.63(1.16)	21.65(1.31)	23.66(1.33)	18.68(1.15)	**18.26(1.25)**
Email spam	U1vsU4	1.73(0.12)	0.06(0.02)	4.25(1.23)	6.25(1.67)	7.87(1.86)	3.86(1.12)	**3.24(2.11)**
	U2vsU4	1.75(0.13)	0.05(0.02)	4.37(1.25)	6.56(1.75)	8.11(2.15)	3.87(1.15)	**3.62(2.25)**
	U3vsU4	1.72(0.11)	0.06(0.03)	4.36(1.24)	6.47(1.73)	7.94(2.01)	3.65(1.03)	**3.34(2.18)**

Bold represents the results of algorithm proposed in this paper

ment analysis and email spam, the average precisions are 84.15%, 84.34%, 73.65% and 92.97%, respectively.

(4) In terms of training time shown in Table 13, it can also be seen that MultiSTLP has obvious advantage over transfer learning algorithms TrGNB, ARTL, STL-SVM and TSVM-GP in training time, because of selecting representative data set from source domain. IC-SVM needs less training time than other six 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. Even though, SVM is effective in training time, its classification accuracy is not prominent on transfer learning problems.

In summary, through the comparative analysis of experimental results, we can see that the algorithm proposed in this paper is effective and efficient. It also shows the rationality and effectiveness of the proposed algorithm.

Finally, to test the differences between MultiSTLP and benchmark algorithms with similar classification results, the Wilcoxon signed rank test is applied to these methods. According to the contents of Table 10, the average classification accuracy of all algorithms is shown.

In Table 10, we can see the average classification accuracy of all algorithms on real datasets. The results of the Wilcoxon test on real-world datasets 20-Newsgroups, TRECVID 2005, Sentiment analysis and Email spam are discussed below.

20-Newsgroups: the classification accuracy of MultiSTLP is only 0.97% higher than TrGNB; therefore, when using MultiSTLP and TrGNB to classify three cross-domain tasks, each task is repeated 10 times, the values of W^+ and W^- are $+143$ and -24, respectively. For the bilateral test of $\alpha = 0.05$, when $n = 30$, by querying the distribution table of the Wilcoxon signed rank test, $T^{0.025} = 137$. Because of $W^+ > T^{0.025}$, H_0 was accepted: there was no significant difference in the classification results between the two methods.

TRECVID 2005: the classification accuracy of MultiSTLP increases 0.89% by comparison with TSVM-GP; therefore, when using MultiSTLP and TSVM-GP to classify 5 cross-domain tasks, each task is repeated 10 times, the values of W^+ and W^- are $+576$ and -89, respectively. For the bilateral test of $\alpha = 0.05$, when $n = 50$, by querying the distribution table of the Wilcoxon signed rank test, $T^{0.025} = 434$. Because of $W^+ > T^{0.025}$, H_0 was accepted: there was no significant difference in the classification results between the two methods.

Sentiment analysis The classification accuracy of MultiSTLP is only 0.95% higher than STL-SVM; therefore, when using MultiSTLP and STL-SVM to classify three cross-domain tasks, each task is repeated 10 times, the values of W^+ and W^- are $+169$ and -29, respectively. For the bilateral test of $\alpha = 0.05$, when $n = 30$, by querying the distribution table of the Wilcoxon signed rank test, $T^{0.025} = 137$. Because of $W^+ > T^{0.025}$, H_0 was accepted: there was no significant difference in the classification results between the two methods.

Email spam: compared with TrGNB and ARTL, the classification accuracy of MultiSTLP increases 0.83 and 0.36%. When MultiSTLP, TrGNB and ARTL are used to classify three cross-domain tasks, each task was repeated 10 times. For MultiSTLP and TrGNB, the values of W^+ and W^- are $+159$ and -47, respectively. For the bilateral test of $\alpha = 0.05$, when $n = 30$, by querying the distribution table of the Wilcoxon signed rank test, $T^{0.025} = 137$. Because of $W^+ > T^{0.025}$, H_0 was accepted: there was no significant difference in the classification results between the two methods. Similarly, for MultiSTLP and TrGNB, the values of W^+ and W^- are 182 and -26, $W^+ > T^{0.025}$, H_0 is accepted: there is no significant difference in the classification results of the two methods.

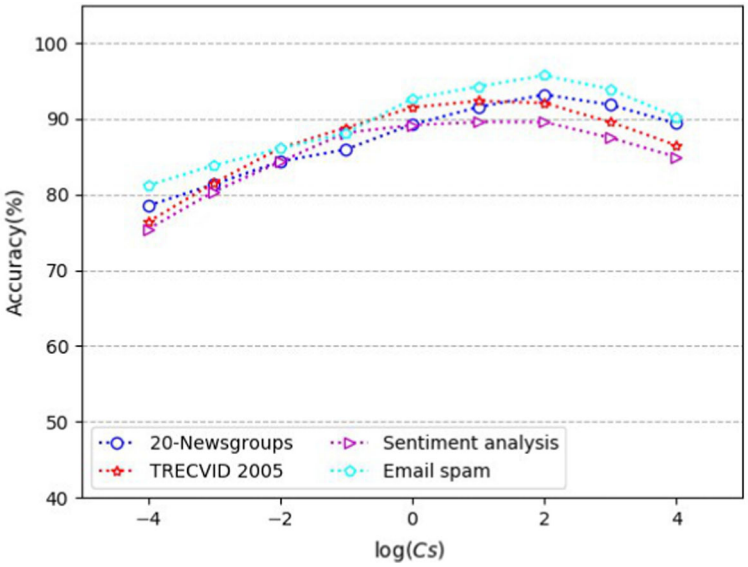

Fig. 4 Sensitivity of parameter C_t for MultiSTLP

4.4 Parameter Sensitivity Analysis

In TrGNB, ARTL and TSVM-GP, they performed sensitivity analysis of parameters with large influence on performance of algorithm: TrGNB analyzed the difference parameter between source and target domains; MMD regularization parameter and manifold regularization parameter were analyzed to determine the influence of parameters on classification performance for ARTL; TSVM-GP analyzed the regularization coefficient of source domain, the regularization coefficient of source of target domains, and the trade-off term. Like them, we analyze the sensitivity of three parameters: regularization coefficient of target domain C_t, regularization coefficient of source domain C_t and trade-off coefficient λ in objective function of MultiSTLP, which illustrates their influence on the classification performance in this section. For each parameter, we fix the other two parameters at the optimal values determined by cross-validation, and then observe the effect of parameter with different values on the classification result. The experimental results are shown in Figs. 4, 5 and 6.

From the results of Figs. 4, 5 and 6, the following conclusions can be drawn:

(1) From Figs. 4 and 5, MultiSTLP is considerably sensitive to regularization parameters $C_s$and C_t with a wide range. This denotes that it is critical to determine the value of parameter by some effective strategies.
(2) In Fig. 6, we can see that shows that MultiSTLP is sensitive to λ. When λ approaches 1, MultiSTLP achieves the best classification performance. As λ is too small, the distribution difference between source and target domains is ignored, so the classification performance will be poor. When it is too large, the distribution difference between source and target domains will theoretically be larger, but this will also reduce the knowledge of source domains that can be transferred to target domain, and the classification performance is also poor.

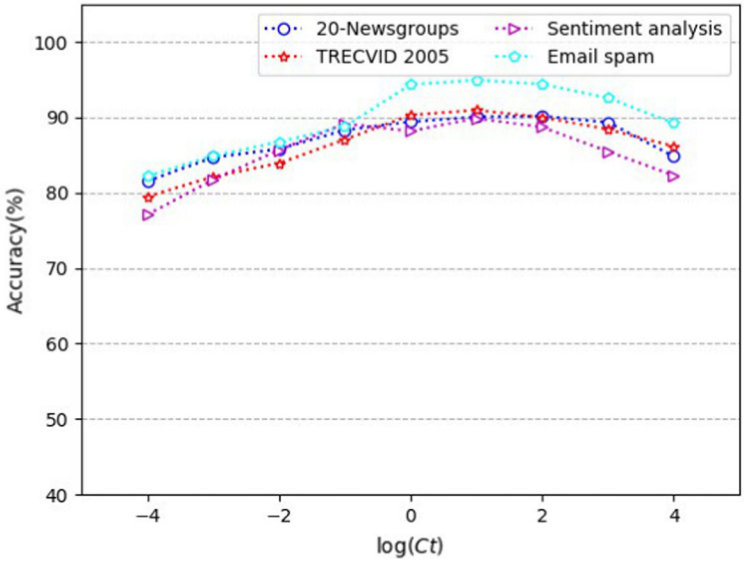

Fig. 5 Sensitivity of parameter C_s for MultiSTLP

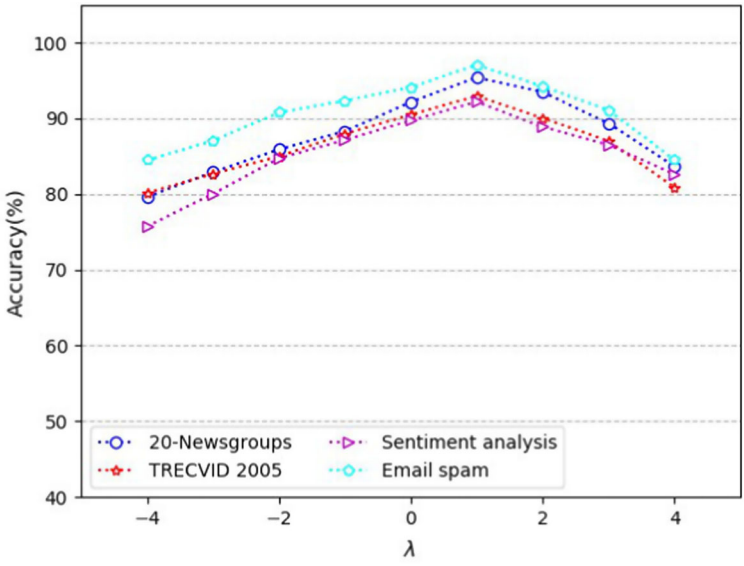

Fig. 6 Sensitivity of parameter λ for MultiSTLP

5 Conclusion

Aiming at the current hot data privacy protection problem in machine learning, we propose a MultiSTLP by combining group probability information with transfer learning. MultiSTLP first uses AESVM to select representative dataset in each source domain; secondly, based on minimizing the marginal probability difference calculate the weight of samples of

representative dataset; then, according to conditional probability difference calculates the weight of source domains; finally, the group probability knowledge in the target domain and the weighted knowledge of representative dataset from multiple source domains are combined into the support vector machine structure risk minimization framework, the objective function of MultiSTLP is proposed and proved theoretically. MultiSTLP not only improves training efficiency and result, but also protects data privacy. The effectiveness of the classifier obtained by training MultiSTLP is demonstrated on experiments utilizing four real-world datasets 20-Newsgroups, TRECVID 2005, Sentiment analysis and Email spam. Although the experimental results show that the MultiSTLP algorithm has advantages over the benchmark algorithms, it is still a problem worthy of further study in terms of training efficiency and domain similarity.

Acknowledgements This work was supported by Information Technology Research Center,Beijing Institute of Remote Sensing Equipment,the Second Academy of China Aerospace Science Industry Corp.

Appendix 1

Proof of Theorem 1 By using the Lagrangian optimization theorem, we can obtain the following Lagrangian function for Eq. (1.1):

$$
\begin{aligned}
L(w_t, w_s, b_t, b_s, \xi, \xi^*, \xi^s, \alpha, \alpha^*, \alpha^s, r, r^*, r^s) &= \frac{1}{2}||\mathbf{w}_t||^2 + \frac{1}{2M}\sum_{i=1}^{M}||\mathbf{w}_{s_i}||^2 \\
&+ C_t \sum_{i=\sum_{j=1}^{M} n_j+1}^{\sum_{j=1}^{M} n_j+d} (\xi_i + \xi_i^*) + \frac{1}{2M}\sum_{i=1}^{M} C_{s_i} \sum_{j=1}^{n_{S_j}} \xi_j^{s_i} \\
&+ \frac{1}{2M}\sum_{i=1}^{M}||\mathbf{w}_t - \gamma^{S_i}\mathbf{w}_{s_i}||^2 - \frac{1}{2M}\sum_{i=1}^{M}\sum_{j=1}^{n_{S_j}} r_j^{s_i}\xi_j^{s_i} - C_t \sum_{i=\sum_{j=1}^{M} n_j+1}^{\sum_{j=1}^{M} n_j+d} r_i\xi_i - C_t \sum_{i=\sum_{j=1}^{M} n_j+1}^{\sum_{j=1}^{M} n_j+d} r_i^*\xi_i^* \\
&- \sum_{i=1}^{M}\sum_{j=1}^{n_{S_i}} \alpha_j^{s_i}(y_j^{S_i}(\mathbf{w}_{s_i}^T\varphi(\mathbf{x}_j^{S_i}) + b_{s_i}) - 1 + \xi_j^{s_i}) \\
&- \sum_{i=\sum_{j=1}^{M} n_j+1}^{\sum_{j=1}^{M} n_j+d} \alpha_i \left(\frac{1}{|G_i|}\sum_{j\in G_i}(\mathbf{w}_t^T\varphi(\mathbf{x}_j) + b_t) \right) - \widetilde{y}_i + \varepsilon_i + \xi_i)) \\
&- \sum_{i=\sum_{j=1}^{M} n_j+1}^{\sum_{j=1}^{M} n_j+d} \alpha_i^* \left(\widetilde{y}_i + \varepsilon_i + \xi_i^* - \frac{1}{|G_i|}\sum_{j\in G_i}(\mathbf{w}_t^T\varphi(\mathbf{x}_j) + b_t) \right)
\end{aligned} \tag{1.1}
$$

where $\alpha^{s_i} = (\alpha_1^{s_i}, \alpha_2^{s_i}, \ldots, \alpha_{n_{s_i}}^{s_i})$, $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_d)$, $\alpha^* = (\alpha_1^*, \alpha_2^*, \ldots, \alpha_d^*)$, $r^{s_i} = (r_1^{s_i}, r_2^{s_i}, \ldots, r_{n_{s_i}}^{s_i})$, $r = (r_1, r_2, \ldots, r_d)$ and $r^* = (r_1^*, r_2^*, \ldots, r_d^*)$ are Lagrange multipliers. Then the following equations can be considered as the necessary conditions of the optimal solution:

$$\frac{\partial L}{\partial \xi_j^{s_i}} = 0 \Rightarrow \sum_{i=1}^{M} \sum_{j=1}^{n_{S_i}} (r_j^{s_i} + \alpha_j^{n_{S_i}}) = C_{s_i} \tag{1.2}$$

$$\frac{\partial L}{\partial \xi_j^{(*)}} = 0 \Rightarrow \sum_{i=\sum_{j=1}^{M} n_j+1}^{\sum_{j=1}^{M} n_j+d} (\alpha_i^{(*)} + r_i^{(*)}) = C_t \tag{1.3}$$

$$\frac{\partial L}{\partial \mathbf{w}_{s_i}} = 0 \Rightarrow \frac{1}{M} \sum_{i=1}^{M} \mathbf{w}_{s_i} - \frac{\lambda}{M} \sum_{i=1}^{M} (\mathbf{w}_t - \gamma^{S_i} \mathbf{w}_{s_i}) - \sum_{i=1}^{M} \sum_{j=1}^{n_{S_i}} \alpha_j^{s_i} y_j^{S_i} \varphi(x_j^{S_i}) = 0 \tag{1.4}$$

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{w}_t} = 0 \Rightarrow \mathbf{w}_t + \frac{\lambda}{\mathrm{M}} \sum_{i=1}^{M} (\mathbf{w}_t - \gamma^{S_i} \mathbf{w}_{s_i}) - \sum_{i=\sum_{j=1}^{M} n_j+1}^{\sum_{j=1}^{M} n_j+d} \alpha_i \frac{1}{|G_i|} \sum_{j \in G_i} \varphi(\mathbf{x}_j) \\ + \sum_{i=\sum_{j=1}^{M} n_j+1}^{\sum_{j=1}^{M} n_j+d} \alpha_i^* \frac{1}{|G_i|} \sum_{j \in G_i} \varphi(\mathbf{x}_j) = 0 \end{aligned} \tag{1.5}$$

$$\frac{\partial L}{\partial b_{s_i}} = 0 \Rightarrow \sum_{i=1}^{M} \sum_{j=1}^{n_{S_i}} \alpha_j^{s_i} y_j^{S_i} = 0 \tag{1.6}$$

$$\frac{\partial L}{\partial b_t} = 0 \Rightarrow \sum_{i=\sum_{j=1}^{M} n_j+1}^{\sum_{j=1}^{M} n_j+d} (\alpha_i - \alpha_i^*) = 0 \tag{1.7}$$

Substituting Eqs. (1.2)–(1.7) into Eq. (1.1) by simplification, and we can obtain the dual of Eq. (1.8).

$$\min_{\beta} \frac{1}{2} \beta^T \widetilde{\mathbf{K}} \beta + \widetilde{\mathbf{e}}^T \beta$$

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

$$\beta = [\alpha^{s_1}, \alpha^{s_2}, \ldots, \alpha^{s_M}, \alpha, \alpha^*]^{\mathrm{T}},$$

$$0 \le \beta \le \left[\underbrace{C_{s_1}, \ldots, C_{s_1}}_{n_{S_1}}, \ldots, \underbrace{C_{s_M}, \ldots, C_{s_M}}_{n_{S_M}}, \underbrace{C_t, \ldots, C_t}_{d}, \underbrace{C_t, \ldots, C_t}_{d} \right],$$

$$\mathbf{f}^{\mathrm{T}} = \left[y_1^{S_1}, \ldots, y_{n_{S_1}}^{S_1}, \ldots, y_1^{S_M}, \ldots, y_{n_{S_M}}^{S_M}, \underbrace{1, \ldots, 1}_{d}, \underbrace{-1, \ldots, -1}_{d} \right],$$

$$\widetilde{\mathbf{e}} = \left[\underbrace{0, \ldots, 0}_{n_{S_1}}, \ldots, \underbrace{0, \ldots, 0}_{n_{S_M}}, \varepsilon - \widetilde{y}, \varepsilon + \widetilde{y} \right]$$

$$\widetilde{\mathbf{K}} = \begin{bmatrix} \frac{M+\lambda}{1+2\lambda M}\mathbf{K}_{s_1,s_1} + \frac{\lambda}{M}, \ldots, \frac{M+\lambda}{1+2\lambda M}\mathbf{K}_{s_1,s_M} + \frac{\lambda}{M}, \frac{\lambda}{1+2\lambda M}\mathbf{K}_{s_1,t}, -\frac{\lambda}{1+2\lambda M}\mathbf{K}_{s_1,t} \\ \cdots \\ \frac{M+\lambda}{1+2\lambda M}\mathbf{K}_{s_M,s_1} + \frac{\lambda}{M}, \ldots, \frac{M+\lambda}{1+2\lambda M}\mathbf{K}_{s_M,s_M} + \frac{\lambda}{M}, \frac{\lambda}{1+2\lambda M}\mathbf{K}_{s_M,t}, -\frac{\lambda}{1+2\lambda M}\mathbf{K}_{s_M,t} \\ \frac{\lambda}{1+2\lambda M}\mathbf{K}_{s_1,t}^T, \ldots, \frac{\lambda}{1+2\lambda M}\mathbf{K}_{s_M,t}^T, \frac{M+\lambda}{1+2\lambda M}\mathbf{K}_{t,t} - \frac{\lambda}{1+2\lambda M}\mathbf{K}_{t,t} \\ -\frac{\lambda}{1+2\lambda M}\mathbf{K}_{s_1,t}^T, \ldots, -\frac{\lambda}{1+2\lambda M}\mathbf{K}_{s_M,t}^T, -\frac{M+\lambda}{1+2\lambda M}\mathbf{K}_{t,t}, \frac{M+\lambda}{1+2\lambda M}\mathbf{K}_{t,t} \end{bmatrix}_{\left(\sum_{i\in M} n_{S_i}+2d\right)\times\left(\sum_{i\in M} n_{S_i}+2d\right)}$$

$$\mathbf{K}_{s_i,s_i} = (y_j^{S_i} y_q^{S_i} k(x_j^{S_i} x_q^{S_i}))_{j,q=1,2,\ldots,n_{S_i}}$$

$$\mathbf{K}_{s_i,t} = \left(\frac{\widetilde{y}_j^{S_i}}{|G_k|} \sum\nolimits_{q\in G_k} k(\mathbf{x}_j, \mathbf{x}_q) \right)_{j=1,\ldots,n_{S_i}, q=1,\ldots,d},$$

$$\mathbf{K}_{t,t} = \left(\frac{1}{|G_i||G_j|} \sum\nolimits_{i'\in G_i} \sum\nolimits_{j'=G_j} k(\mathbf{x}_{i'}, \mathbf{x}_{j'}) \right)_{i,j=1,\ldots,d}. \tag{1.8}$$

□

Appendix 2

Proof of Theorem 2 The matrix $\widetilde{\mathbf{K}}$ can be decomposed into $\widetilde{\mathbf{K}} = \widetilde{\mathbf{K}}_1 + \widetilde{\mathbf{K}}_2 + \widetilde{\mathbf{K}}_3 + \widetilde{\mathbf{K}}_4$. Among them, $\widetilde{\mathbf{K}}_1$, $\widetilde{\mathbf{K}}_2$, $\widetilde{\mathbf{K}}_3$ and $\widetilde{\mathbf{K}}_4$ are as follows.

$$\widetilde{\mathbf{K}}_1 = \frac{\lambda}{1+2\lambda M} \begin{bmatrix} \mathbf{K}_{s_1,s_1}, \ldots, \mathbf{K}_{s_1,s_M}, \mathbf{K}_{s_1,t}, -\mathbf{K}_{s_1,t} \\ \cdots \\ \mathbf{K}_{s_M,s_1}, \ldots, \mathbf{K}_{s_M,s_M}, \mathbf{K}_{s_M,t}, -\mathbf{K}_{s_M,t} \\ \mathbf{K}_{s_1,t}^T, \ldots, \mathbf{K}_{s_M,t}^T, \mathbf{K}_{t,t}, -\mathbf{K}_{t,t} \\ -\mathbf{K}_{s_1,t}^T, \ldots, -\mathbf{K}_{s_M,t}^T, -\mathbf{K}_{t,t}, \mathbf{K}_{t,t} \end{bmatrix}_{\left(\sum_{i\in M} n_{S_i}+2d\right)\times\left(\sum_{i\in M} n_{S_i}+2d\right)}$$

$$\widetilde{\mathbf{K}}_2 = \frac{M}{1+2\lambda M} \begin{bmatrix} \mathbf{K}_{s_1,s_1}, \ldots, \mathbf{K}_{s_1,s_M}, 0, 0 \\ \cdots \\ \mathbf{K}_{s_M,s_1}, \ldots, \mathbf{K}_{s_M,s_M}, 0, 0 \\ 0, \ldots, 0, 0, 0 \\ 0, \ldots, 0, 0, 0 \end{bmatrix}_{\left(\sum_{i\in M} n_{S_i}+2d\right)\times\left(\sum_{i\in M} n_{S_i}+2d\right)}$$

$$\widetilde{\mathbf{K}}_3 = \frac{\lambda}{M} \begin{bmatrix} 1, \ldots, 1, 0, 0 \\ \cdots \\ 1, \ldots, 1, 0, 0 \\ 0, \ldots, 0, 0, 0 \\ 0, \ldots, 0, 0, 0 \end{bmatrix}_{\left(\sum_{i\in M} n_{S_i}+2d\right)\times\left(\sum_{i\in M} n_{S_i}+2d\right)}$$

$$\widetilde{\mathbf{K}}_4 = \frac{M}{1+2\lambda M}\begin{bmatrix} 0,\ldots,0,0,0 \\ \cdots \\ 0,\ldots,\mathbf{K}_{t,t},-\mathbf{K}_{t,t} \\ 0,\ldots,-\mathbf{K}_{t,t},\mathbf{K}_{t,t} \end{bmatrix}_{\left(\sum_{i\in M} n_{S_i}+2d\right)\times\left(\sum_{i\in M} n_{S_i}+2d\right)}$$

For $\widetilde{\mathbf{K}}$, setting

$$\begin{aligned} Q_1 = \sqrt{\frac{M}{1+2\lambda M}}\Bigg(& y_1^{S_1}x_1^{S_1},\ldots,y_{n_{s_1}}^{S_1}x_{n_{s_1}}^{S_1},\ldots,y_1^{S_M}x_1^{S_M},\ldots,y_{n_{s_M}}^{S_M}x_{n_{sM}}^{S_M}, \\ & \frac{1}{|G_1|}\sum_{i\in G_1}x_i,\ldots,\frac{1}{|G_d|}\sum_{i\in G_d}x_i, \\ & -\frac{1}{|G_1|}\sum_{i\in G_1}x_i,\ldots,-\frac{1}{|G_d|}\sum_{i\in G_d}x_i\Bigg), \end{aligned}$$

it is symmetric and positive semidefinite matrix, so $\widetilde{\mathbf{K}}_1 = Q_1^T Q_1$ is symmetric and positive semidefinite matrix, too. Like this, $\widetilde{\mathbf{K}}_2$, $\widetilde{\mathbf{K}}_3$ and $\widetilde{\mathbf{K}}_4$ are symmetric and positive semidefinite matrix, thus $\widetilde{\mathbf{K}}$ is symmetric and positive semidefinite matrix. Therefore, Eq. (16) is a standard convex quadratic programming problem. Theorem 2 is hold. □

References

1. Jordan MI, Mitchell TM (2015) Machine learning: trends, perspectives, and prospects. Science 349(6245):255–260
2. Barbu A, She Y, Ding L et al (2017) Feature selection with annealing for computer vision and big data learning. IEEE Trans Pattern Anal Mach Intell 39(2):272–286
3. Jingmei L, Weifei W, Di X (2020) An intrusion detection method based on active transfer learning. Intell Data Anal 24(2):363–383
4. Stefano R, Zied M, Francesco M, Alberto C (2021) Emotion recognition from speech: an unsupervised learning approach. Int J Comput Intell Syst 14(1):23–35
5. Li D, Liu J, Yang Z et al (2021) Speech emotion recognition using recurrent neural networks with directional self-attention. Expert Syst Appl 173(3):114683–114694
6. Jr A, Sp A, Bkk B et al (2021) Unsupervised multi-sense language models for natural language processing tasks. Neural Netw 142:397–409
7. Pintas JT, Fernandes L, Garcia A (2021) Feature selection methods for text classification: a systematic literature review. Artif Intell Rev 1(6):2568–2573
8. Day O, Khoshgoftaar TM (2017) A survey on heterogeneous transfer learning. J Big Data 4(1):29–54
9. Weiss K, Khoshgoftaar TM, Wang DD (2016) A survey of transfer learning. J Big Data 3(1):1–40
10. Li J, Wu W, Xue D (2020) Research on transfer learning algorithm based on support vector machine. J Intell Fuzzy Syst 38(10):1–16
11. Wang R, Zhou J, Jiang H et al (2021) A general transfer learning-based Gaussian mixture model for clustering. Int J Fuzzy Syst 23:776–793
12. Yunxin L, Kunling H, Danlei X et al (2020) A transfer learning method using speech data as the source domain for micro-Doppler classification tasks. Knowl-Based Syst 209:106449–106461
13. Deng Z, Wang Z, Tang Z et al (2021) A deep transfer learning method based on stacked autoencoder for cross-domain fault diagnosis. Appl Math Comput 408:126318–126332
14. Peipei J, LiaiLun C, Min-Feng W (2021) Transfer learning based recurrent neural network algorithm for linguistic analysis. ACM Trans Asian Low Resour Lang Inf Process 20(3):1–40
15. Gao J, Fan W, Jiang J et al (2009) Knowledge transfer via multiple model local structure mapping. In: International conference on knowledge discovery & data mining, pp 283–291
16. Quanz B, Huan J (2009) Large margin transductive transfer learning. In: ACM conference on information and knowledge management, pp 1327–1336

17. Long M, Wang J, Ding G et al (2014) Adaptation regularization: a general framework for transfer learning. IEEE Trans Knowl Data Eng 26(5):1076–1089
18. Li M, Dai Q (2018) A novel knowledge-leverage-based transfer learning algorithm. Appl Intell 48(8):2355–2372
19. Wang Y, Gu QQ, Brown D (2018) Differentially private hypothesis transfer learning. In: Proceedings of European conference on machine learning and principles and practice of knowledge discovery in databases, pp 811–826
20. Xie LY, Baytas I M, Lin KX et al (2017) Privacy-preserving distributed multi-task learning with asynchronous updates. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, pp 1195–1204
21. Sarpatwar K, Shanmugam K, Ganapavarapu VS et al (2019) Differentially private distributed data summarization under covariate shift. In: Proceedings of advances in neural information processing systems, pp 14432–14442
22. Rping S (2010) SVM classifier estimation from group probabilities. In: Proceedings of 27th ICML, pp 911–918
23. Quadrianto N, Smola AJ, Caetano TS et al (2009) Estimating labels from label proportions. J Mach Learn Res 10:2349–2374
24. Pengjiang Q, Zhaohong D et al (2015) A novel privacy-preserving probability transductive classifiers from group probabilities based on regression model. J Intell Fuzzy Syst 53(6):91–103
25. Jinmei L, Weifei W, Di X (2020) Transfer Naive Bayes algorithm with group probabilities. Appl Intell 50(1):61–73
26. Tongguang N, Xiaoqing G et al (2018) Scalable transfer support vector machine with group probabilities. Neurocompting 2018:570–582
27. Peng G, Weifei W, Jingmei L (2021) Multi-source fast transfer learning algorithm based on support vector machine. Appl Intell 2021:1–15
28. Jingmei L, Weifei W, Di X, Peng G (2019) Multi-source deep transfer neural networks algorithm. Sensors 19(18):1090–1112
29. Gu Q, Dai Q (2021) A novel active multi-source transfer learning algorithm for time series forecasting. Appl Intell 51(2):1–25
30. Li C, Yang Y, Liang H et al (2021) Transfer learning for establishment of recognition of COVID-19 on CT imaging using small-sized training datasets. Knowl Based Syst 10228:106849–106861
31. Chen YS, Chiang SW, Wu ML (2021) A few-shot transfer learning approach using text-label embedding with legal attributes for law article prediction. Appl Intell 2021:1–19
32. Ohata EF, Chagas J, Bezerra GM et al (2021) A novel transfer learning approach for the classification of histological images of colorectal cancer. J Supercomput 1:1–26
33. Abdulazeem Y, Balaha HM, Bahgat WM, Badawy M (2021) Human action recognition based on transfer learning approach. IEEE Access 9:82058–82069
34. Zhang, He M (2021) CRTL: context restoration transfer learning for cross-domain recommendations. IEEE Intell Syst 36(4):65–72
35. Wan Z, Yang R, Huang M et al (2021) A review on transfer learning in EEG signal analysis. Neurocomputing 421:1–14
36. Nandan M, Khargonekar PP, Talathi SS (2013) Fast SVM training using approximate extreme points. J Mach Learn Res 15(1):59–98