

A general framework for transfer sparse subspace learning

Shizhun Yang · Ming Lin · Chenping Hou ·
Changshui Zhang · Yi Wu

Received: 21 April 2012 / Accepted: 7 July 2012 / Published online: 1 August 2012
© Springer-Verlag London Limited 2012

Abstract In this paper, we propose a general framework for transfer learning, referred to as transfer sparse subspace learning (TSSL). This framework is suitable for different assumptions on the divergence measures of the data distributions, such as maximum mean discrepancy, Bregman divergence, and K–L divergence. We introduce an effective sparse regularization to the proposed transfer subspace learning framework, which can reduce time and space cost obviously, and more importantly, which can avoid or at least reduce over-fitting problem. We give different solutions to the problems based on different distribution distance estimation criteria, and convergence analysis is also given. Comprehensive experiments on the text data sets and the face image data sets demonstrate that TSSL-based methods outperform existing transfer learning methods.

Keywords Transfer learning · Subspace learning · Sparse regularization · MMD · Bregman divergence

1 Introduction

The high dimensionality of data poses challenges to learning tasks such as the curse of dimensionality. A

common way to solve this problem is dimensionality reduction, which has attracted much attention in machine learning and data mining community in the past decades. In the literature, there are mainly two distinct ways for dimensionality reduction, that is, feature selection and feature extraction. In the former, subsets of features are selected directly. In the latter, new features are gained from their original features through algebraic transformation. Despite different motivations of these methods, they can all be interpreted in a unified Graph Embedding framework [1]. Subspace learning algorithms belong to the feature extraction issue. The most popular subspace learning methods include unsupervised principle component analysis (PCA) [2], supervised linear discriminant analysis (LDA) [2], maximum margin criterion (MMC) [3], and locality preserving projection (LPP) [4]. These algorithms project the data by linear transformation according to some optimization criteria. One of the key shortcomings of subspace learning is that new features are linear combinations of all original features. This means while subspace learning facilitates model interpretation and visualization by concentrating the information in a few features, the features themselves are still constructed using all features, hence are often hard to interpret [5]. Then, sparse subspace learning methods attempted to solve this problem effectively [6–9].

Conventional subspace learning algorithms for data mining and machine learning perform well under the assumption that training and testing samples are independent and identically distributed (i.i.d). Unfortunately, for many practical applications, this assumption is always violated, and this will deeply decrease the effect of conventional algorithms. Transfer learning aims to solve the problem when the training data from a source domain and the testing data from a target domain follow different

S. Yang (✉) · C. Hou · Y. Wu
Department of Mathematics and System Science,
National University of Defense Technology,
Changsha 410073, People's Republic of China
e-mail: yangshizhun@126.com

S. Yang · M. Lin · C. Zhang
State Key Laboratory of Intelligent Technology and Systems,
Tsinghua National Laboratory for Information Science
and Technology (TNList), Department of Automation,
Tsinghua University, Beijing 100084,
People's Republic of China

distributions or are represented in different feature spaces [10]. The key idea of transfer learning is that, although distributions between source and target domain are different, there must contend some common knowledge structures across domains. These common structures can be utilized as a bridge for knowledge transfer.

One fundamental motivation of transfer learning in the real applications is the so-called data sparsity problem in target domain, where data sparsity can be defined by a lack of useful labels or sufficient high-quality data in the training set. This sparsity problem in target domain will lead to an over-fitting model when training with conventional methods. The regularization framework is popular in machine learning to address various problems, for example, Tikhonov regularization [11], manifold regularization [12], graph Laplacian-based regularization [13], etc. To overcome the data sparsity problem, we enforce sparse regularization on the transfer learning framework.

The motivation of our work is as follows:

1. Though there are many subspace learning methods via sparse regularization methods, none of them is suitable for transfer learning paradigm, because most of them rely on the i.i.d assumption, which may be impractical in real applications. How to extend the traditional subspace learning methods to solve the transfer learning applications is significative.
2. Due to the fact that data sparsity problems happen in both transfer learning and subspace learning paradigm, how to overcome the over-fitting problem and which kind of sparse regularization to select is worthy of study.
3. There are lots of widely used transfer learning methods, suitable for different assumptions on the divergence measures of the data distributions, such as maximum mean discrepancy (MMD), Bregman divergence, and Kullback–Leibler (K–L) divergence, and most of recent algorithms depend on specific circumstances and applications. So how to unify them into a general framework is a challenging problem.

In this paper, we proposed a general framework for transfer learning, referred to as transfer sparse subspace learning (TSSL). The main contributions of this paper include the following:

1. We successfully extend the traditional subspace learning algorithms such as PCA, LDA, MMC, and LPP to solve transfer learning problems.
2. To deal with the considerable change between distributions of the source and target domains, TSSL minimized the distribution distance via two important criteria, that is, MMD and Bregman divergence. Indeed, TSSL provided a unified framework for handling any distribution distance estimation criterion.

3. We employ sparse regularization term on the transfer subspace learning framework to avoid or at least reduce the over-fitting problems and also reduce time and space cost obviously. We verify that the $L_{2,1}$ -norm regularization is an effective constraint on the transfer subspace learning procedure.

The rest of the paper is organized as follows. In Sect. 2, the previous related works are discussed, and the preliminaries including sparse subspace learning, $L_{2,1}$ -norm, transfer learning, MMD, Bregman divergence, and MMC are introduced. We presented our framework for transfer sparse subspace learning and corresponding solutions in Sect. 3. The experimental results on both text data sets and face data sets are discussed in Sect. 4. Finally, we draw a conclusion and discuss the future work.

2 Previous works and preliminaries

2.1 Sparse subspace learning

When the number of samples is smaller than the number of features, the subspace learning methods may fail, and it is necessary to control the model complexity according to the regularization theory. The most important regularization techniques include L_1 -norm, L_2 -norm, and the elastic net penalty. Recently, sparse subspace learning draws increasing interests, and many dimensionality reduction methods are extended to their sparse version. Zou et al. [6] proposed an elegant sparse PCA algorithm (SPCA) using “Elastic Net” framework for L_1 -penalized regression on regular principle components, solved very effectively using least angle regression (LARS). Moghaddam et al. [7] proposed a spectral bound framework for sparse subspace learning. Particularly, they proposed both exact and greedy algorithms for sparse PCA and sparse LDA [8]. Cai et al. [9] propose a unified sparse subspace learning framework, which builds the connection between regression and many popular graph-based subspace learning algorithms, for example, LDA, LPP, and NPE. Their sparse solutions can be effectively computed with a L_1 -norm regularization in the proposed framework.

Recently, sparse regularization has been widely investigated and also applied into subspace learning studies. L_1 -SVM was proposed to perform feature selection using the L_1 -norm regularization that tends to give sparse solution [14]. A hybrid huberized SVM (HHSVM) was proposed combining both L_1 -norm and L_2 -norm to form a more structured regularization [15]. Obozinsky et al. [16] and Argyriou et al. [17] have developed a similar model for $L_{2,1}$ -norm regularization to couple feature selection across tasks. Such regularization has close connections to group

lasso. On the basis of the motivation that the selected features by sparse subspace learning methods are independent and generally different for each dimension of the subspace, Gu et al. [18] proposed a joint framework based on using $L_{2,1}$ -norm on the projection matrix, which can do feature selection and subspace learning simultaneously.

None of these sparse subspace learning methods is suitable for transfer learning problems, since they ignore the fact that the distributions of source domain data and target domain data are different.

2.2 $L_{2,1}$ -norm

For a matrix $W \in R^{m \times d}$, the $L_{r,p}$ -norm is defined as follows:

$$\|W\|_{r,p} = \left(\sum_{i=1}^m \left(\sum_{j=1}^d |w_{ij}|^r \right)^{p/r} \right)^{1/p} = \left(\sum_{i=1}^m \|w^i\|_r^p \right)^{1/p} \tag{1}$$

where w^i is the i th row of W .

Then, $L_{2,1}$ -norm is defined in the following equation.

$$\|W\|_{2,1} = \sum_{i=1}^m \sqrt{\sum_{j=1}^d w_{ij}^2} = \sum_{i=1}^m \|w^i\|_2 \tag{2}$$

We can easily verify that $L_{2,1}$ -norm is rotational invariant for rows for any rotational matrix R , that is, $\|WR\|_{2,1} = \|W\|_{2,1}$.

We also give an intuitional explanation of $L_{2,1}$ -norm. First, we can compute the L_2 -norm of the rows w^i (corresponding to dimension i) and then compute L_1 -norm of the vector $b(W) = (\|w^1\|_2, \|w^2\|_2, \dots, \|w^m\|_2)$. The magnitudes of the components of the vector $b(W)$ indicate how important each dimension is. The $L_{2,1}$ -norm favors a small numbers of nonzero rows in the matrix W , thereby ensuring that dimensionality reduction will be achieved.

The $L_{2,1}$ -norm of a matrix was introduced [19] as rotational invariant L_1 -norm and used for multitask learning [16, 17]. Argyriou et al. [17] developed a non-convex multitask generalization of the $L_{2,1}$ -norm regularization that can be used to learn a few features common across multiple tasks. Obozinski et al. [16] proposed a type of joint regularization of the model parameters in order to couple feature selection across tasks. Liu et al. [20] consider the $L_{2,1}$ -norm regularized regression model for joint feature selection from multiple tasks, which can be derived in the probabilistic framework by assuming a suitable prior from the exponential family. One appealing feature of the $L_{2,1}$ -norm regularization is that it encourages multiple predictors to share similar sparsity patterns.

Motivated by previous research [16, 18, 20, 21], an $L_{2,1}$ -norm regularization is performed to select features across

all data points with joint sparsity, that is, each feature either has small scores for all data points or has large scores over all data points. In this paper, we also employ $L_{2,1}$ -norm regularization on the projection matrix W to enable subspace learning effectively.

2.3 Transfer learning

In the past decades, there are many transfer learning algorithms, which can be summarized into four cases, that is, instance-based transfer learning, parameter-based transfer learning and relational-knowledge transfer learning, and feature-based transfer learning. We refer to [22] for more information.

The instance-based transfer learning approaches reweighted some labeled data in the source domain for use in the target domain, the representative algorithms including KLIEP [23], TrAdaBoost [24], TransferBoost [25], TrAdaBoost.R2 [26], and MultiSourceTrAdaBoost [27].

The parameter-based transfer learning approaches assumed that the source tasks and the target tasks share some parameters or prior distributions of the hyperparameters of the models. To discover the shared parameters or priors, the knowledge can be transferred across tasks. The representative algorithms include MI-IVM [28], GPDRTL [29], TLVM [30], etc.

The relational-knowledge transfer learning approaches assumed that some relationship among the data of the source and target domains is similar. Statistical relational learning techniques such as MLNs dominate this context [31, 32].

The feature-based transfer learning approaches include feature reweighting [33], feature replication [34], feature projection [35, 36], feature correlation [37], feature subsetting [38], feature extraction [39, 40], etc. The feature-based transfer learning aims to discover a shared feature space in which the data distributions across domains are close to each other. The shared feature space can be constructed in the original feature space [35, 36], or in the projected subspace [41, 42].

Our framework belongs to the feature-based transfer learning. So we focus on some previous representative feature-based algorithms as follows. Structured correspondence learning (SCL) [35] introduces the concept of *pivot features*, which possess high frequency and similar meaning in both auxiliary and target domains. Non-pivot features can be mapped to each other via the pivot features from the unlabeled data of both source and target domains. Blitzer et al. [36] proposed to use mutual information (MI) to choose the pivot features instead of using more heuristic criteria. MI-SCL tries to find some pivot features that have high dependence on the labels in the source domain. Pan et al. [41] exploited the maximum

mean discrepancy embedding (MMDE) method, originally designed for dimensionality reduction, to learn a low-dimensional space to reduce the difference of distributions between different domains for transductive transfer learning. However, MMDE may suffer from its computational burden. Thus, Pan et al. [42] further proposed an effective feature extraction algorithm, known as transfer component analysis (TCA), to overcome the drawback of MMDE.

Since the distribution divergence between the original source and target domains is large in transfer learning settings, the classification or regression function f may not generalize well in the target domain. So in this paper, we want to alleviate this difficulty by reducing the distribution distance across domains in a projected latent space. That is to say, our framework aims to employ subspace learning algorithm to discover the common shared projected subspace. The advantage of our method is that we do so in a regularization framework, which takes the distribution distance regularization and the sparse regularization into consideration. Also, our framework can avoid over-fitting problems.

2.4 Maximum mean discrepancy (MMD)

There are some criteria to estimate the distance between different distributions [41]. Many criteria are parametric because they need intermediate density estimation. MMD is a relevant criterion for comparing distributions based on reproducing kernel Hilbert space (RKHS) [43]. Given two domains X and Y , let $X = \{x_1, x_2, \dots, x_{n_1}\}$ and $Y = \{y_1, y_2, \dots, y_{n_2}\}$ be random variable sets with different distributions P and Q . The empirical estimation by MMD will be as follows:

$$\text{Dist}(X, Y) = \sup_{\|f\|_{\mathcal{H}} \leq 1} \left(\frac{1}{n_1} \sum_{i=1}^{n_1} f(x_i) - \frac{1}{n_2} \sum_{i=1}^{n_2} f(y_i) \right) \quad (3)$$

where $f: \mathcal{X} \rightarrow \mathcal{H}$ and H is a universal RKHS [44]. In a RKHS, function evaluation can be written as $f(x) = \langle \phi(x), f \rangle$, where $\phi(x): \mathcal{X} \rightarrow \mathcal{H}$ is a kernel-induced feature map, and the empirical estimation of MMD can be rewritten as:

$$\text{Dist}(X, Y) = \left\| \frac{1}{n_1} \sum_{i=1}^{n_1} \phi(x_i) - \frac{1}{n_2} \sum_{i=1}^{n_2} \phi(y_i) \right\|_H \quad (4)$$

On the basis of the MMD theory [43], the distance between distributions of two sets of samples is just the distance between the mean values of these two sets of samples in a RKHS.

In order to enable knowledge transfer for mismatching distribution data sets, Brian et al. [45, 46] explored a

feature extraction perspective, starting with the popular sparse coding approach, which learns a set of higher-order features for the data. They improved the original sparse coding technique by incorporating distribution distance regularization and the target data label information into the general objective function. Ren et al. [47] proposed a multiple kernel learning framework improved by MMD to solve transfer learning problems. The model not only utilizes the capacity of kernel learning to construct a non-linear hyperplane which maximizes the separation margin, but also reduces the distribution discrepancy between training and testing data simultaneously. Zhang et al. [48] proposed an approach, which performs multiple related clustering tasks simultaneously through domain adaptation. A shared subspace will be learned, where the gap of distributions via MMD among tasks is reduced, and the shared knowledge will be transferred through all tasks by exploiting the strengthened relation in the learned subspace. Uguroglu et al. [49] presented a method to identify variant and invariant features between two data sets. Unlike traditional feature-based transfer learning methods, rather than finding a projection of the feature space to maximize the similarity between source domain and target domain data via MMD, Duan et al. [50] proposed a cross-domain kernel learning framework called domain transfer multiple kernel learning (DTMKL), which simultaneously learns a kernel function and a robust classifier by minimizing both the structural risk functional and the distribution mismatch via MMD between source domain and target domain data.

Motivated by the success of MMD used for transfer learning problems, in this paper, we also employ MMD as one of the basic distribution distance estimation criteria, and our goal is to reduce distribution gap between the projected source and target domain subspace. The difference with the above algorithms is that we consider not only the distribution estimation but also the sparse regularization, which is important to handling over-fitting problem.

2.5 Bregman divergence

Definition [51] Let φ be a continuous-differentiable real-valued and strict convex function defined on a closed convex set Ω , and a Bregman distance function $d_\varphi(x, y)$ associated with the function φ is defined as follows:

$$d_\varphi(x, y) = \nabla \varphi(x) - \nabla \varphi(y) - \langle \nabla \varphi(y), (x - y) \rangle \quad (5)$$

for any points $x, y \in \Omega$, where $\nabla \varphi(y)$ is the gradient of φ evaluated at y .

According to the definition, different convex functions φ define different specific forms of Bregman divergences. Some frequently used divergences are all specific forms of Bregman divergences that are showed in Table 1 [52].

Table 1 Some frequently used divergences

Domain	$\varphi(x)$	$d_\varphi(x, y)$	Divergence
\mathbb{R}^d	$\ x\ ^2$	$\ x - y\ ^2$	Squared Euclidean distance
\mathbb{R}^d	$x^T A y$	$(x - y)^T A (x - y)$	Squared Mahalanobis distance
d -Simplex	$\sum_{j=1}^d x_j \log_2 x_j$	$\sum_{j=1}^d x_j \log_2 \left(\frac{x_j}{y_j}\right)$	Kullback–Leibler (K–L) divergence
\mathbb{R}_{++}	$-\log(x)$	$\frac{x}{y} - \log\left(\frac{x}{y}\right) - 1$	Itakura–Saito distance

There are some important properties of Bregman divergence. A Bregman divergence is not generally symmetric, that is, it does not always hold that $d_\varphi(x, y) = d_\varphi(y, x)$. The squared Euclidean distance and the squared Mahalanobis distance are two examples of symmetric Bregman divergences, and the Kullback–Leibler (K–L) divergence is asymmetric. A Bregman divergence $d_\varphi(x, y)$ is convex w.r.t. its left variate x , but non-convex w.r.t. its right variate y . More details about Bregman divergence can be found in [51].

Si et al. [53] presented a family of subspace learning algorithms based on Bregman divergence regularization, which transfers the knowledge gained in the source domain data to the target domain data. The transfer subspace learning (TSL) framework extends many classical subspace learning algorithms under transfer learning settings, such as TPCA, TLDA, TLPP, and TMFA. In the following work, Si et al. [54, 55] proposed the cross-domain discriminative Hessian Eigenmaps (CDHE) and cross-domain discriminative locally linear embedding (CDLLE), which incorporated Bregman divergence regularization. Wu et al. [56] proposed a scheme of learning Bregman distance function with side information. Gao et al. [57] proposed a transfer learning framework for latent variable model, which can utilize the Bregman divergence of the source and target domain data to modify the parameters of the obtained latent variable model. Zhang et al. [58] deal with multitask clustering-based Bregman divergence, which aims to improve performance of each single task and also discover the relationship between clusters of different tasks.

Motivated by these excellent works, we also unify the Bregman divergence into our general transfer learning framework. The advantage of our method is that we take both distribution estimation and the sparse regularization into consideration, so that our methods can be extended to more real applications.

2.6 Maximum margin criterion (MMC)

MMC [3] aims at maximizing the average margin between classes in the projected space. Therefore, the feature extraction criterion is defined as:

$$J = \frac{1}{2} \sum_{i=1}^C \sum_{j=1}^C p_i p_j d(C_i, C_j) \tag{6}$$

where C is the number of distinct classes and p_i, p_j are the prior probability of class i and class j , respectively; the interclass margin is defined as:

$$\begin{aligned} d(C_i, C_j) &= d(m_i, m_j) - s(C_i) - s(C_j) s(C_i) \\ &= tr(S_i), s(C_j) = tr(S_j) \end{aligned} \tag{7}$$

where m_i, m_j are the mean vectors of the class C_i and the class C_j and S_i, S_j are the covariance matrix of the class C_i and the class C_j . After simple mathematical operation, we can obtain the following formula:

$$J = tr(S_b - S_w) \tag{8}$$

The between-class scatter matrix S_b and the within-class scatter matrix S_w are defined as:

$$\begin{aligned} S_b &= \sum_{i=1}^C n_i (m_i - m)(m_i - m)^T \\ S_w &= \sum_{i=1}^C (X_i - m_i)(X_i - m_i)^T \end{aligned} \tag{9}$$

where n_i is the number of class C_i and m is the mean vector of all data. Then, the MMC can be formulated as:

$$\arg \max_{W \in \Omega} F(W) = \arg \max_{W \in \Omega} tr(W^T (S_b - S_w) W) \tag{10}$$

Obviously, we can get the optimal W by solving the generalized eigenvalue problem: $(S_b - S_w)W = \lambda W$. Therefore, W is composed of the first d largest eigenvectors of $S_b - S_w$.

The number of clusters is predefined as c , $F \in R^{n \times c}$ is the indicator matrix, $F_{ij} = 1/\sqrt{l_j}$ if x_i belong to j th cluster, and $F_{ij} = 0$ otherwise, where l_j is the number of samples in j th cluster. We can easily verify the following equations:

$$S_w = X(I - FF^T)X^T; \quad S_b = XF F^T X^T \tag{11}$$

Different with LDA, we need not calculate the inverse of S_w , which allows us to avoid the small sample size problem easily. Due to the advantage of MMC method, we employ it as an example to test the efficiency of our

framework. Of course, our framework is suitable for any subspace learning methods.

3 Transfer sparse subspace learning (TSSL)

3.1 Problem statement and notations

In a transfer learning setting, we denote the source domain data as $\mathcal{D}_S = \{(x_1, z_1), (x_2, z_2), \dots, (x_{n_1}, z_{n_1})\}$, $x_i \in \mathbb{R}^m$, $i = 1, 2, \dots, n_1$ and z_i is the corresponding label. Similarly, we denote the target domain data as $\mathcal{D}_T = \{x_{n_1+1}, x_{n_1+2}, \dots, x_{n_1+n_2}\}$, and we assumed $x_i \in \mathbb{R}^m$, $i = n_1 + 2, \dots, n_1 + n_2$. Denote $X = [x_1, x_2, \dots, x_n] \in \mathbb{R}^{m \times n}$, where $n = n_1 + n_2$. Let $P(X_S)$ and $Q(X_T)$ be the marginal distribution of X_S and X_T , respectively, and usually $P(X_S) \neq Q(X_T)$.

3.2 General framework

3.2.1 Subspace learning framework

A subspace learning algorithm finds a low-dimensional subspace \mathbb{R}^d , where samples from different classes can be well separated or a specific redundancy is minimized. The objective function of subspace learning framework is as follows:

$$W = \arg \min_{W \in \Omega} F(W) \quad (12)$$

Whatever the objective is, to approximate the transformation from \mathbb{R}^m to \mathbb{R}^d , the linear function can be used, $y_i = W^T x_i$, where $i = 1, 2, \dots, n_1 + n_2$, where y_i is the low-dimensional representation of the samples x_i , $W \in \mathbb{R}^{m \times d}$, $x_i \in \mathbb{R}^m$, $y_i \in \mathbb{R}^d$. Denote the low-dimensional samples as:

$$\begin{aligned} Y &= Y_S \cup Y_T, Y_S = \{y_1, y_2, \dots, y_{n_1}\}, Y_T \\ &= \{y_{n_1+1}, y_{n_1+2}, \dots, y_{n_1+n_2}\} \end{aligned}$$

Let the probability density for the source domain data and the target domain data in the projected subspace W be $p(y_s)$ and $q(y_t)$, respectively, and in general $p(y_s) \neq q(y_t)$.

In this paper, we use the MMC as the subspace learning method. Of course, our framework can be easily extended to all of other subspace learning methods such as PCA, LDA, and LPP.

3.2.2 Incorporate sparse regularization

The regularization principals can deal with various machine learning problems [53], and there are some advantages of incorporating sparse regularization to the subspace learning framework. First, it can make the subspace more succinct and simpler, and the calculation will be more effective. Parsimony is especially an important

factor when the dimension of the original samples is very high. Second, it can control the importance of original dimensions and decrease the influence brought by possible over-fitting problem. Third, it provides a good interpretation of the subspace and thus reveals an explicit relationship between the objective of the model and the given variables. It is important because we can understand the problem better by learning which kind of dimension plays more important role. The objective function of sparse subspace learning framework incorporated with sparse regularization is as follows:

$$W = \arg \min_{W \in \Omega} F(W) + \alpha \Phi(W) \quad (13)$$

We enforce the sparsity penalty on the projection matrix W and encourage the rows of W to be zeroed as much as possible. The intuition behind this is that we expect the source domain and target domain data to only depend on a subset of the latent dimensions. The zero-valued rows of W remove the influence of the corresponding latent dimensions.

In this paper, we enforced $L_{2,1}$ -norm regularization on the W , the rows of W will be zero as much as possible. This lets us automatically discover the dimensionality of the latent space. Furthermore, in our transfer learning setting, we assume the source domain and the target domain share some information, and this regularization will favor representing this shared information in a common latent dimension space.

3.2.3 Incorporate distribution divergence regularization

The subspace learning framework incorporated with sparse regularization works well when source domain data and target domain data are independent and identically distributed (i.i.d). But in transfer learning setting, we relax this assumption, that is, $P(X_S) \neq Q(X_T)$. Then, the distributions of low-dimensional projected data are also different, that is, $P(Y_S) \neq Q(Y_T)$. So we should take the difference between $P(Y_S)$ and $Q(Y_T)$ into consideration and ensure they are close to each other in the projected subspace. Let $\text{Dist}(Y_S, Y_T)$ be the distance estimation of the different distributions between the source and target domains in the projected subspace. Then, we can get the general framework for transfer sparse subspace learning as follows:

$$W = \arg \min_{W \in \Omega} F(W) + \alpha \Phi(W) + \beta \text{Dist}(Y_S, Y_T) \quad (14)$$

In this paper, we use the MMD criterion and the Bregman divergence criterion as the two main regularizations. Of course, our framework can be easily extended to all of other distribution divergence regularizations such as Kullback–Leibler (K–L) divergence, β divergence, Jensen–Shannon divergence, and χ^2 divergence.

3.3 MMD-based regularization

3.3.1 Reformulation of MMD regularization

Pan et al. [41] developed a transfer learning technique for learning in a latent space, called MMDE. MMDE embeds the data from both domains into a common low-dimensional latent subspace. The key idea is to formulate it as a kernel learning problem using the kernel trick, $K_{ij} = K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$, and to learn the Kernel matrix defined on all the data:

$$K = \begin{bmatrix} K_{Y_S, Y_S} & K_{Y_S, Y_T} \\ K_{Y_T, Y_S}^T & K_{Y_T, Y_T} \end{bmatrix} \in R^{(n_1+n_2) \times (n_1+n_2)} \tag{15}$$

where K_{Y_S, Y_S} , K_{Y_T, Y_T} and K_{Y_S, Y_T} are the Gram matrix defined on the source domain D_S , target domain D_T , and cross-domains, respectively. Y_S and Y_T are the low-dimensional representations of source domain data X_S and target domain data X_T , respectively. Then, we can get the new formulation of minimizing the distance (measured by MMD) between the two domains as:

$$\text{Dist}(Y_S, Y_T) = \text{tr}(KL) \tag{16}$$

where $L = [L_{ij}] \geq 0$ with

$$L_{ij} = \begin{cases} \frac{1}{n_1^2} & \text{when } x_i, x_j \in X_S \\ \frac{1}{n_2^2} & \text{when } x_i, x_j \in X_T \\ -\frac{1}{n_1 n_2} & \text{otherwise} \end{cases} \tag{17}$$

There are several limitations of MMDE. First, it is transductive and cannot handle out-of-domain samples. Second, the resultant kernel learning problem has to be solved by expensive SDP complexity solvers. Third, the obtained K has to be processed by PCA. This may discard potential useful information in K .

To overcome the limitations of MMDE above, Pan proposed a new feature extraction method, TCA, for transfer learning [42]. It learns a set of transfer components in a RKHS such that when projecting domain data onto the latent space spanned by the transfer components, the distance between domains can be reduced. According to the empirical kernel map [59], $K = (KK^{-1/2})(K^{-1/2}K)$, the projection matrix $\tilde{W} \in \mathbb{R}^{(n_1+n_2) \times m}$ transforms the empirical kernel map features to an m -dimensional space (where $m \ll n_1 + n_1$). The new resultant kernel matrix is as follows:

$$\tilde{K} = (KK^{-1/2}\tilde{W})(\tilde{W}^TK^{-1/2}K) = KWW^TK$$

where $W = K^{-1/2}\tilde{W}$. The distance between the two domains can be formulated as:

$$\begin{aligned} \text{Dist}(Y_S, Y_T) &= \text{tr}(\tilde{K}L) = \text{tr}((KWW^TK)L) \\ &= \text{tr}(W^TKLKW) \end{aligned} \tag{18}$$

3.3.2 Reformulation of sparse regularization

In this paper, we incorporate $L_{2,1}$ -norm to the objective as the sparse regularization.

Let $\Phi(W) = \|W\|_{2,1} = \sum_{i=1}^m \sqrt{\sum_{j=1}^d w_{ij}^2} = \sum_{i=1}^m \|w^i\|_2$, $D = (d_{ij}) \in R^{m \times m}$ be the diagonal matrix with the i th diagonal element, $d_{ii} = 1/(2\|w^i\|_2)$, $i = 1, 2, \dots, m$, where w^i is the i th row of W . It can be easily verified that [21]

$$\min \|W\|_{2,1} \Leftrightarrow \min \text{tr}(W^TDW) \tag{19}$$

For more details, please refer to [21].

3.3.3 Transfer sparse subspace learning-based MMD (TSSL_MMD)

Besides reducing the distance between two marginal distributions, we should also preserve data properties that are useful for the target supervised learning task. In this part, we select the MMC as the base subspace learning method. As we know, the formulation of MMC is equally as follows:

$$\arg \min_{W^TW=I} F(W) = \arg \min_{W^TW=I} \text{tr}(W^T(S_w - S_b)W) \tag{20}$$

To avoid the rank deficiency of the denominator in the generalized eigenvalue decomposition, a regularization term $W^TW = I$ is needed.

Then, we can get the final optimization formulation of the transfer sparse subspace learning-based MMD (TSSL_MMD):

$$\begin{aligned} \arg \min_{W^TW=I} & \text{tr}(W^T(S_w - S_b)W) + \alpha \text{tr}(W^TDW) \\ & + \beta \text{tr}(W^TKLKW) \end{aligned} \tag{21}$$

The first term of Eq. (21) is to learn a shared subspace, the second term is to handle data sparsity and over-fitting problems, and the last term is to reduce the gap of distributions via MMD among domains in the reduced subspace. By minimizing Eq. (21), the shared knowledge will be transferred through domains by exploiting the strengthened relation in the learned subspace.

3.3.4 Solution for TSSL_MMD

According to Eq. (11), we can reformulate the TSSL_MMD as follows:

$$\begin{aligned} \arg \min_{\substack{W^TW=I \\ F^TF=I}} & \text{tr}(W^TX(I - 2FF^T)X^TW) + \alpha \text{tr}(W^TDW) \\ & + \beta \text{tr}(W^TKLKW) \end{aligned} \tag{22}$$

We can see that there are three different variables that should be optimized, that is, W , D , and F . It is difficult to compute them simultaneously. We alternately optimize them, and then we can get W , D , and F .

Step 1 Fixing W , compute F , the second and the third term are fixed, so the optimization problem becomes

$$\arg \max_{F^T F=I} \text{tr}(F^T X^T W W^T X F)$$

Clearly, we can use spectral decomposition technique to solve this problem. That is, the optimal F is formed by the eigenvectors corresponding to the m largest eigenvalues of the matrix $X^T W W^T X$.

Step 2 Fixing F , compute W and D , we notice that there are still two variables to be optimized. We use nesting optimization technique.

Step 2.1 Fixing W , compute D , we can easily update D as follows:

$$D = (d_{ii}), \quad \text{where } d_{ii} = 1 / \left(2 \|w^i\|_2 \right), \quad i = 1, 2, \dots, m$$

Step 2.2 Fixing D , compute W , the optimization problem becomes

$$\arg \min_{W^T W=I} \text{tr}(W^T (X(I - 2FF^T)X^T + \alpha D + \beta K L K) W)$$

We can also use spectral decomposition technique to solve this problem. Moreover, the optimal W is formed by the eigenvectors corresponding to the m smallest eigenvalues of the matrix $X(I - 2FF^T)X^T + \alpha D + \beta K L K$, where $m \ll n_1 + n_2 - 1$.

The iteration procedure is repeated until the algorithm converges. We also give the algorithm convergence analysis in the next section.

The main algorithm is presented in Table 5.

3.3.5 Convergence analysis of TSSL_MMD

In this section, we will prove that TSSL_MMD monotonically decreases the objective of the problem in Eq. (21). Firstly, we give the lemma from [21].

Lemma 1 For any nonzero vector $w, v \in \mathbb{R}^d$, the following inequality holds.

$$\|w\|_2 - \frac{\|w\|_2^2}{2\|v\|_2} \leq \|v\|_2 - \frac{\|v\|_2^2}{2\|v\|_2}$$

Theorem 1 The algorithm will monotonically decrease the objective of the problem in Eq. (21) in each iteration and will converge to the optimum of the problem.

Proof It can be easily verified that optimizing Eq. (21) is equivalent to solving Eq. (22). As seen in algorithm, when fixing D as D_t , we can compute W and F . In the t iteration, we should solve the following problem:

$$W_{t+1}, F_{t+1} = \arg \min_{W^T W=I, F^T F=I} \text{tr}(W^T X(I - 2FF^T)X^T W) + \alpha \text{tr}(W^T D_t W) + \beta \text{tr}(W^T K L K W)$$

Then we can get the following equation:

$$\begin{aligned} & \text{tr}(W_{t+1}^T X(I - 2F_{t+1}F_{t+1}^T)X^T W_{t+1}) + \alpha \text{tr}(W_{t+1}^T D_t W_{t+1}) \\ & + \beta \text{tr}(W_{t+1}^T K L K W_{t+1}) \leq \text{tr}(W_t^T X(I - 2F_t F_t^T)X^T W_t) \\ & + \alpha \text{tr}(W_t^T D_t W_t) + \beta \text{tr}(W_t^T K L K W_t) \end{aligned}$$

Since

$$\|W\|_{2,1} = \sum_{i=1}^m \|w^i\|_2, \quad d_{ii} = 1 / \left(2 \|w^i\|_2 \right), \quad i = 1, 2, \dots, m$$

w^i is the i th row of W . The above inequality indicates that

$$\begin{aligned} & \text{tr}(W_{t+1}^T X(I - 2F_{t+1}F_{t+1}^T)X^T W_{t+1}) + \alpha \sum_{i=1}^m \frac{\|w_{t+1}^i\|_2^2}{2\|w_t^i\|_2} \\ & + \beta \text{tr}(W_{t+1}^T K L K W_{t+1}) \leq \text{tr}(W_t^T X(I - 2F_t F_t^T)X^T W_t) \\ & + \alpha \sum_{i=1}^m \frac{\|w_t^i\|_2^2}{2\|w_t^i\|_2} + \beta \text{tr}(W_t^T K L K W_t) \end{aligned} \quad (23)$$

According to Lemma 1, for each i , we have

$$\|w_{t+1}^i\|_2 - \frac{\|w_{t+1}^i\|_2^2}{2\|w_t^i\|_2} \leq \|w_t^i\|_2 - \frac{\|w_t^i\|_2^2}{2\|w_t^i\|_2}$$

Then the following inequality holds

$$\sum_{i=1}^m \|w_{t+1}^i\|_2 - \frac{\|w_{t+1}^i\|_2^2}{2\|w_t^i\|_2} \leq \sum_{i=1}^m \|w_t^i\|_2 - \frac{\|w_t^i\|_2^2}{2\|w_t^i\|_2} \quad (24)$$

Combining Eqs. (23) and (24), we can get the following result:

$$\begin{aligned} & \text{tr}(W_{t+1}^T X(I - 2F_{t+1}F_{t+1}^T)X^T W_{t+1}) + \alpha \sum_{i=1}^d \|w_{t+1}^i\|_2 \\ & + \beta \text{tr}(W_{t+1}^T K L K W_{t+1}) \leq \text{tr}(W_t^T X(I - 2F_t F_t^T)X^T W_t) \\ & + \alpha \sum_{i=1}^d \|w_t^i\|_2 + \beta \text{tr}(W_t^T K L K W_t) \end{aligned}$$

That is to say

$$\begin{aligned} & \text{tr}(W_{t+1}^T X(I - 2F_{t+1}F_{t+1}^T)X^T W_{t+1}) + \alpha \|W_{t+1}\|_{2,1} \\ & + \beta \text{tr}(W_{t+1}^T K L K W_{t+1}) \leq \text{tr}(W_t^T X(I - 2F_t F_t^T)X^T W_t) \\ & + \alpha \|W_t\|_{2,1} + \beta \text{tr}(W_t^T K L K W_t) \end{aligned}$$

This inequality indicates the algorithm will monotonically decrease the objective of the problem in Eq. (21) in each iteration. Besides, since the three items in Eq. (21) are convex functions and the objective function has lower bounds, such as zero, the above iteration will converge to the optimum solution. In the following experiment section, we can see that our algorithm converges fast. \square

3.4 Bregman divergence–based regularization

3.4.1 Reformulation of Bregman divergence regularization

On the basis of the definition above, we can give the Bregman divergence–based regularization, which measures the distance between $P(Y_S)$ and $Q(Y_T)$.

$$\text{Dist}(Y_S, Y_T) = \int d(p(y_s), q(y_t))d\mu \tag{25}$$

where $d\mu$ is the Lebesgue measure. The right side is also called the U -divergence on the subspace \mathbb{R}^d . When we set $\varphi(y) = y^2$, the regularization item reduces to squared Euclidean distance form:

$$\text{Dist}(Y_S, Y_T) = \int d(p(y_s), q(y_t))d\mu = \int (p(y_s), q(y_t))^2 dy \tag{26}$$

Then, we use kernel density estimation (KDE) technique [60] to estimate the distribution $P(Y_S)$ and $Q(Y_T)$ in the projected subspace W , that is,

$$\begin{aligned} p(y_s) &= (1/n_1) \sum_{i=1}^{n_1} G_{\Sigma_1}(y - y_i), q(y_t) \\ &= (1/n_2) \sum_{j=n_1+1}^{n_1+n_2} G_{\Sigma_2}(y - y_j) \end{aligned} \tag{27}$$

For two arbitrary Gaussian kernels, we have

$$\int G_{\Sigma_1}(y - y_i)G_{\Sigma_2}(y - y_j) = G_{\Sigma_1+\Sigma_2}(y_i - y_j) \tag{28}$$

Then, we can get the discrete form of Bregman divergence:

$$\begin{aligned} \text{Dist}(Y_S, Y_T) &= \frac{1}{n_1^2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_1} G_{\Sigma_{11}}(y_j - y_i) \\ &+ \frac{1}{n_2^2} \sum_{i=n_1+1}^{n_1+n_2} \sum_{j=n_1+1}^{n_1+n_2} G_{\Sigma_{22}}(y_j - y_i) \\ &- \frac{1}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=n_1+1}^{n_1+n_2} G_{\Sigma_{12}}(y_j - y_i) \end{aligned} \tag{29}$$

where

$$\Sigma_{11} = \Sigma_1 + \Sigma_1, \quad \Sigma_{12} = \Sigma_1 + \Sigma_2, \quad \Sigma_{22} = \Sigma_2 + \Sigma_2.$$

3.4.2 Transfer sparse subspace learning–based Bregman divergence (TSSL_BD)

In this section, we presented a Bregman divergence–based regularization $\text{Dist}(Y_S, Y_T)$, which measures the distribution difference of samples drawn from different domains in a projected subspace. We also employ the $L_{2,1}$ -norm as the sparse regularization, and the MMC as the base subspace learning method. Then, we can get the final optimization formulation of the transfer sparse subspace learning–based Bregman divergence (TSSL_BD)

$$\begin{aligned} \arg \min_{W^T W=I} & \text{tr}(W^T(S_w - S_b)W) + \alpha \text{tr}(W^T DW) \\ & + \beta \text{Dist}(Y_S, Y_T). \end{aligned} \tag{30}$$

3.4.3 Solution for TSSL_BD

The convexity of TSSL_BD method depends on all of the three terms of the objective function. The convexity of the first term $F(W)$ depends on a particular subspace learning method, such as PCA, MMC, and LPP. The convexity of the second term $\Phi(W)$ depends on the selection of sparse regularization, such as L_1 -norm, L_2 -norm, and $L_{2,1}$ -norm. The convexity of the third term $\text{Dist}(Y_S, Y_T)$ depends on the selection of the distribution estimation criteria of the different data sets. So it is not easy to give a general convexity of TSSL problem theoretically, and it is problem dependent.

Because the Eq. (30) is not convex, the solution can be obtained by the gradient decent algorithm, that is,

$$\begin{aligned} W \leftarrow & W - \eta(\partial_W \text{tr}(W^T(S_w - S_b)W) + \alpha \partial_W \text{tr}(W^T DW) \\ & + \beta \partial_W \text{Dist}(Y_S, Y_T)) \end{aligned} \tag{31}$$

where η is the learning rate and ∂_W is the gradient with respect to W .

Next, we calculate the derivative of the three terms one by one.

First, it is simple that the derivative of $\text{tr}(S_w - S_b)W$ with respect to W is

$$\partial_W \text{tr}(W^T(S_w - S_b)W) = 2(S_w - S_b)W \tag{32}$$

Second, due to the fact that D is related with W as follows:

$$d_{ii} = 1 / \left(2\|w^i\|_2, \|w^i\|_2 = \sum_{j=1}^d w_{ij}^2 \right)$$

So the derivative of $\text{tr}(W^T DW)$ with respect to W is as follows:

$$\begin{aligned} \frac{\partial \text{tr}(W^T DW)}{\partial W} &= \frac{\partial \left(\sum_{i=1}^d \sum_{j=1}^m w_{ji}^2 d_{ij} \right)}{\partial w_{j'i'}} DW \\ &= 2 \sum_{i=1}^d \sum_{j=1}^m d_{j'i'} w_{j'i'} + \sum_{i=1}^d \sum_{j=1}^m w_{j'i'}^2 \frac{\partial d_{j'i'}}{\partial w_{j'i'}} \\ &= 2 \sum_{i=1}^d \sum_{j=1}^m d_{j'i'} w_{j'i'} \\ &+ \sum_{i=1}^d \sum_{j=1}^m w_{j'i'}^2 \frac{w_{j'i'}}{-2 \left(\sqrt{\sum_{j=1}^d w_{j'i'}^2} \right)^3} \\ &= \sum_{i=1}^d \sum_{j=1}^m d_{j'i'} w_{j'i'} = DW \end{aligned} \tag{33}$$

Table 2 Data description: the 20-newsgroups data sets (the number of the document is inside the parentheses)

Data set	The training procedure				The testing procedure	
	Source (labeled)	Source (labeled)	Target (unlabeled)	Target (unlabeled)	Target (to be predicted)	Target (to be predicted)
C2R	com.graphics (973)	rec.sport.baseball (994)	com.windows.x (100)	rec.sport.hockey (100)	com.windows.x (888)	rec.sport.hockey (899)
S2T	sci.electronics (984)	talk.politics.misc (775)	sci.space (100)	talk.religion.misc (100)	sci.space (887)	talk.religion.misc (528)
C2S	com.graphics (973)	sci.space (987)	com.windows.x (100)	sci.electronics (100)	com.windows.x (888)	sci.electronics (884)
T2R	rec.sport.baseball (994)	talk.politics.misc (775)	rec.sport.hockey (100)	talk.religion.misc (100)	rec.sport.hockey (899)	talk.religion.misc (528)
C2T	com.graphics (973)	talk.politics.misc (775)	com.windows.x (100)	talk.religion.misc (100)	com.windows.x (888)	talk.religion.misc (528)
R2S	rec.sport.baseball (994)	sci.space (987)	rec.sport.hockey (100)	sci.electronics (100)	rec.sport.hockey (899)	sci.electronics (884)

Third, similar to [53], we can get the derivative of $\text{Dist}(Y_S, Y_T)$ with respect to W :

$$\begin{aligned}
& \sum_{i=1}^{n_1+n_2} \frac{\text{Dist}(Y_S, Y_T)}{\partial y_i} \frac{\partial y_i}{\partial W} \\
&= \sum_{i=1}^{n_1} \frac{\text{Dist}(Y_S, Y_T)}{\partial y_i} \frac{\partial y_i}{\partial W} + \sum_{i=n_1+1}^{n_1+n_2} \frac{\text{Dist}(Y_S, Y_T)}{\partial y_i} \frac{\partial y_i}{\partial W} \\
&= \frac{2}{n_1^2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_1} G_{\Sigma_{11}}(y_i - y_j) (\Sigma_{11})^{-1} (y_j - y_i) x_i^T \\
&\quad + \frac{2}{n_2^2} \sum_{i=n_1+1}^{n_1+n_2} \sum_{j=n_1+1}^{n_1+n_2} G_{\Sigma_{22}}(y_i - y_j) (\Sigma_{22})^{-1} (y_j - y_i) x_i^T \\
&\quad - \frac{2}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=n_1+1}^{n_1+n_2} G_{\Sigma_{12}}(y_j - y_i) (\Sigma_{12})^{-1} (y_j - y_i) x_i^T \\
&\quad - \frac{2}{n_1 n_2} \sum_{i=n_1+1}^{n_1+n_2} \sum_{j=1}^{n_1} G_{\Sigma_{12}}(y_j - y_i) (\Sigma_{12})^{-1} (y_j - y_i) x_i^T
\end{aligned} \tag{34}$$

So we can obtain a solution W of TSSL_BD in an iterative way.

Similar to Sect. 3.3.4, after obtaining the projection matrix W , we can predict the labels of target domain data. In the training stage, the labels of target domain are blind to the subspace learning methods. One reference image for each test class is preserved, so that the classification can be done in the testing stage. Then, we adopt the nearest neighbor classifier to predict the labels of remaining test images in the selected subspace. The main algorithm is presented in Table 6.

In this section, we do not give any theoretical convergence analysis, and instead, we give some experimental results in the next section. Experimental results demonstrate that we can get a fast convergence.

4 Experimental results

4.1 Data sets descriptions

4.1.1 20-Newsgroups data sets

The 20-newsgroups data set collects approximately 20,000 documents across 20 different newsgroups. It is widely used to test the performance of text mining algorithms. We employed the conventional strategy to construct the data sets, that is, ignored the headers, removed stop words, and selected the top words by mutual information. In order to make the 20-newsgroups data set suitable for our transfer learning problem setting, we reorganize the 20 subcategories and put them in related but different domains. The preprocessing strategy of 20-newsgroups data set is similar to [61]. We reorganize the 20 subcategories into 6 source and target domain pairs. Within each domain pair, the texts are from only two top categories. And within each domain in the pair, positive instances consist of some subcategories in one top category, while negative instances consist of some other subcategories in the other top category. In each data set, we randomly selected 100 target documents as the training samples, and the remaining were used as the testing samples. Detail settings of the 20-newsgroups data set are shown in Table 2.

4.1.2 Face image data sets

To the best of our knowledge, no public face data sets are constructed for transfer learning. In this paper, similar to [53], we build a set of data sets based on the existing face data sets, for example, ORL [62], YALE [2], FERET [63], and UMIST [64]. ORL face database [62] contains 10 images for each of the 40 human subjects, which are taken

Table 3 Data description: the face image data sets (the number of the images is inside the parentheses)

Date sets	The source domain data set (labeled)	The target domain data set (unlabeled)
O2Y	ORL (400)	YALE (165)
O2F	ORL (400)	FERET (600)
O2U	ORL (400)	UMIST (564)
Y2F	YALE (165)	FERET (600)
Y2U	YALE (165)	UMIST (564)
F2U	FERET (600)	UMIST (564)

at different times, varying the lighting, facial expressions, and facial details. The original images (with 256 gray levels) have size 92×112 , which are resized to 32×32 for efficiency. The YALE face database [2] consists of 165 images from 15 individuals, and each has 11 images with different facial expressions or configurations. The FERET face database [63] contains 13,539 face images collected from 1565 individuals, where images are photographed with different sizes, poses, illuminations, and facial expressions. The UMIST face database [64] consists of 564 images of 20 people with different races, genders, and appearances, covering a range of poses from profile to

frontal views. For simplicity, we randomly select 100 individuals, each of which has 6 images for FERET. All images are used from the other three face databases. We generated six new data sets for transfer learning settings by mixing some of them together. Detail settings of the new face image data sets are shown in Table 3.

4.2 Experimental setup

For the text data sets, we compare our transfer sparse subspace learning methods, TSSL_MMD and TSSL_BD, with the following typical methods:

1. SVM, use linear support vector machine to train a classifier in the original space of source domain, and then directly apply this classifier to the testing target data;
2. PCA [2], first apply PCA to get a latent space of source and target domain and then use SVM to train a classifier for the testing target data;
3. MMDE [41], TCA [42], SCL [35], and TSL [53], some classical transfer learning methods, first get the common latent space of source and target domain, and then use SVM to train a classifier for the testing target data.

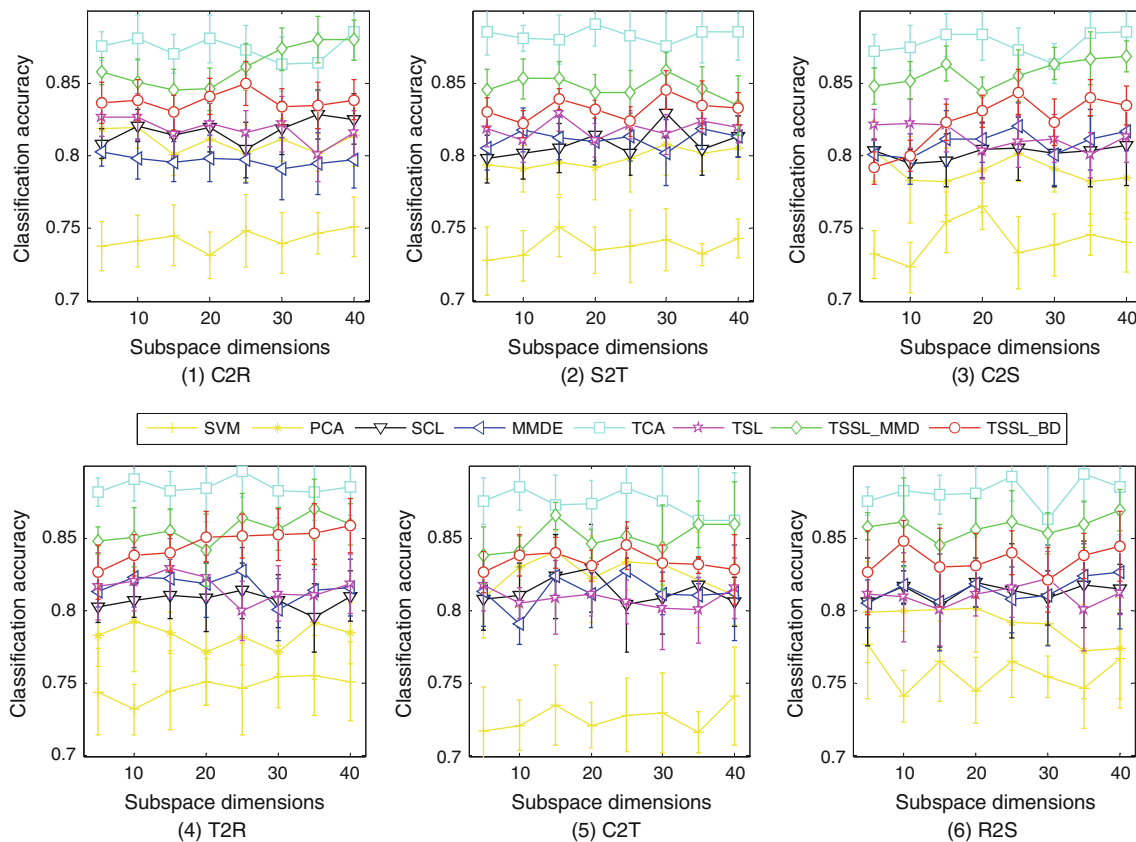


Fig. 1 The performances of various methods based on different subspace dimensions on the text data sets

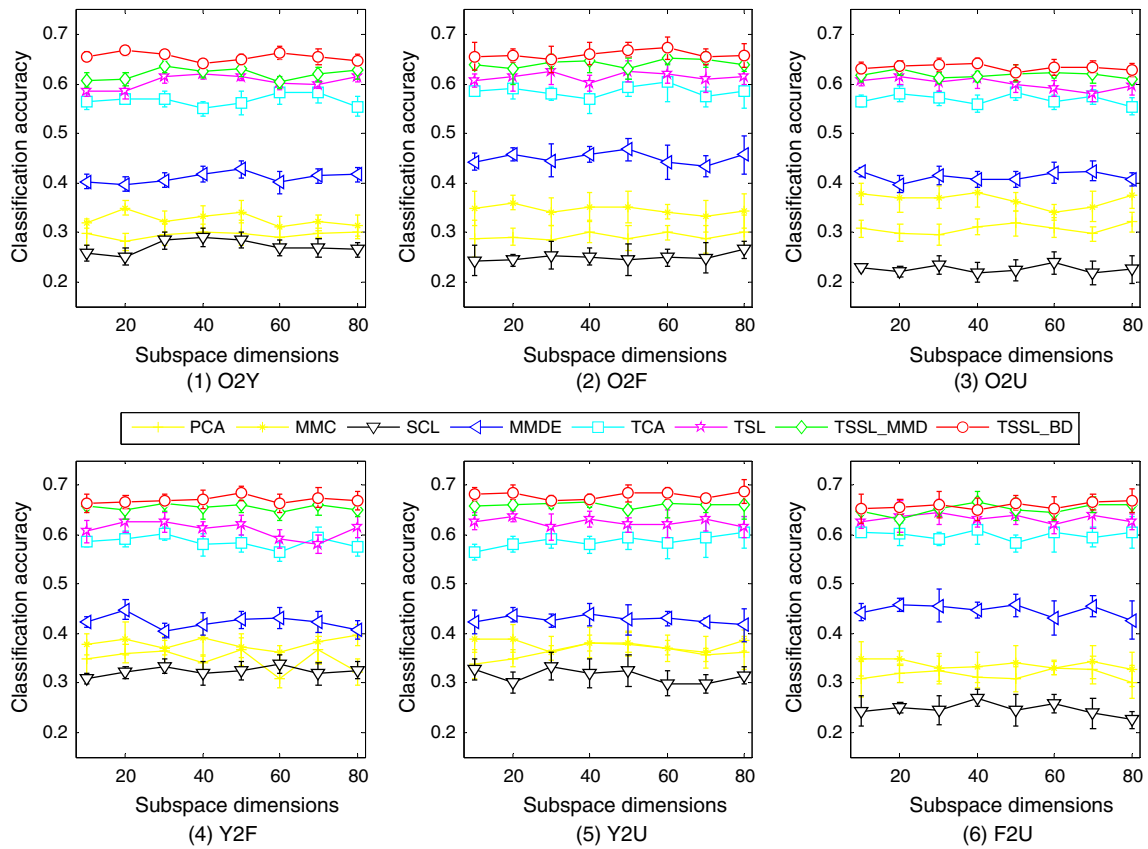


Fig. 2 The performances of various methods based on different subspace dimensions on the face image data sets

Table 4 The classification accuracies of various methods (the standard deviations are inside the parentheses)

Methods	C2R	S2T	C2S	T2R	C2T	R2S
Text data sets						
SVM	74.32(2.36)	75.21(3.58)	71.25(6.01)	76.25(1.98)	77.69(2.05)	78.01(3.22)
PCA	82.14(4.01)	79.26(1.02)	80.32(2.36)	78.23(7.98)	81.29(2.55)	80.00(2.87)
SCL	80.68(1.08)	79.65(3.58)	82.25(5.25)	79.25(2.14)	80.56(2.55)	81.99(1.99)
MMDE	81.58(2.58)	82.25(1.78)	80.25(2.33)	79.36(2.55)	80.33(2.64)	82.36(2.99)
TCA	82.19(5.88)	82.69(4.66)	79.20(3.55)	81.55(6.22)	80.56(4.81)	84.58(6.11)
TSL	81.02(2.01)	83.56(5.22)	81.32(2.69)	80.65(6.32)	81.00(3.55)	83.02(2.71)
TSSL_MMD	85.02(2.05)	86.01(2.51)	83.99(6.05)	85.45(3.42)	84.02(6.00)	86.91(2.19)
TSSL_BD	83.99(2.06)	84.06(5.02)	84.02(6.33)	84.36(2.65)	83.01(6.03)	85.32(2.36)
	O2Y	O2F	O2U	Y2F	Y2U	F2U
Face image data sets						
PCA	29.02(3.06)	28.36(3.25)	31.02(3.57)	32.06(2.15)	33.67(3.85)	30.48(2.15)
MMC	34.06(3.51)	35.28(3.49)	38.48(4.02)	37.29(2.59)	38.48(6.15)	36.01(3.16)
SCL	25.05(6.56)	24.58(3.15)	22.15(3.49)	30.15(3.49)	31.09(2.55)	25.02(3.41)
MMDE	41.09(3.01)	45.25(2.01)	43.12(3.09)	42.28(4.21)	43.15(2.01)	45.28(3.65)
TCA	50.36(3.36)	52.14(2.16)	54.15(3.16)	52.19(3.16)	52.59(2.65)	54.16(6.13)
TSL	59.16(2.03)	61.29(3.12)	60.39(2.98)	62.35(3.21)	62.65(2.68)	63.16(2.98)
TSSL_MMD	63.02(3.49)	64.06(5.06)	61.06(2.69)	66.01(3.64)	66.00(6.01)	64.36(2.58)
TSSL_BD	66.03(6.69)	65.69(3.67)	63.69(3.22)	67.00(6.01)	68.69(2.88)	65.01(6.00)

Table 5 Transfer sparse subspace learning based on MMD (TSSL_MMD)

Input: The source domain data $D_S = \{(x_1, z_1), (x_2, z_2), \dots, (x_{n_1}, z_{n_1})\}$; the target domain data $D_T = \{x_{n_1+1}, x_{n_1+2}, \dots, x_{n_1+n_2}\}$, $X = [x_1, x_2, \dots, x_n] \in R^{m \times n}$, $n = n_1 + n_2$, balance parameters α, β

Output: The labels of the target domain data

Initialize: Construct kernel matrix K form the source and target domain data based on Eq. (15), matrix L based on Eq. (17), set $t = 0$. Initialize $D_t \in R^{m \times m}$ as an identity matrix

Stage one: Computing Projection matrix W , alternatively update F , W and D until convergence

- 1: Fixing W_t , compute F_{t+1} . The optimal F_{t+1} is formed by the eigenvectors corresponding to the m largest eigenvalues of the matrix $X^T W_t W_t^T X$
- 2: Fixing F , compute W_{t+1} and D_{t+1} . We use nesting optimization technique
 - 2.1: Fixing W_t , compute $D_{t+1}.D_{t+1} = (d_{ii})$, where $d_{ii} = 1 / \left(2 \|w_{t+1}^i\|_2 \right)$
 - 2.2: Fixing D_t , compute W_{t+1} . The optimal W_{t+1} is formed by the eigenvectors corresponding to the m smallest eigenvalues of the matrix $X(I - 2F_t F_t^T)X^T + \alpha D_t + \beta K L K$

Stage two: Predict the labels of the target domain data

- 1: Get the new low-dimensionality representations of the original data via $Y = W^T X$, $Y = Y_S \cup Y_T$, $Y_S = \{y_1, y_2, \dots, y_{n_1}\}$, $Y_T = \{y_{n_1+1}, y_{n_1+2}, \dots, y_{n_1+n_2}\}$
- 2: Learn a classifier or regressor, $f: y_i \rightarrow z_i$, $i = 1, 2, \dots, n_1$
- 3: Use the learned classifier or regressor to predict the labels of the target domain data, $z_i = f(x_i)$, $i = n_1 + 1, n_1 + 2, \dots, n_1 + n_2$

Table 6 Transfer sparse subspace learning based on BD (TSSL_BD)

Input: The source domain data $D_S = \{(x_1, z_1), (x_2, z_2), \dots, (x_{n_1}, z_{n_1})\}$; the target domain data $D_T = \{x_{n_1+1}, x_{n_1+2}, \dots, x_{n_1+n_2}\}$, $X = [x_1, x_2, \dots, x_n] \in R^{m \times n}$, $n = n_1 + n_2$, balance parameters α, β

Output: The labels of the target domain data

Initialize: Construct kernel matrix K form the source and target domain data based on Eq. (15), matrix L based on Eq. (17). Initialize W as MMC projection matrix, set $\eta(k) = \eta(0)/k$

Stage one: Computing Projection matrix W by the gradient decent algorithm

Use source domain and target domain data to learn the projection matrix W based on Eq. (30–34)

Stage two: Predict the labels of the target domain data

- 1: Get the new low-dimensionality representations of target domain data via $Y_T = W^T X$, $Y_T = \{y_{n_1+1}, y_{n_1+2}, \dots, y_{n_1+n_2}\}$
- 2: Use KNN classifier to predict the labels of the target domain data, $z_i = f(x_i)$, $i = n_1 + 1, n_1 + 2, \dots, n_1 + n_2$

For the face image data sets, we compare our transfer sparse subspace learning methods, TSSL_MMD and TSSL_BD, with the following typical methods:

1. PCA [2], MMC [3], first apply PCA or MMC to get a latent space of source domain W , and then project the target domain data into this latent space by W , and then use KNN to find labels of unlabeled target domain data;
2. MMDE [41], TCA [42], SCL [35], and TSL [53], some classical transfer learning methods, first get the common latent space of source and target domain, and then use KNN to find labels of unlabeled target domain data.

In this section, we use classification accuracy to measure the classification performance. We run 10 repetitions and report the means and the standard deviations of all methods.

In TSSL_MMD method, the most important factor is the selection of kernel function. On the basis of the well-known observation that the linear kernel is often adequate

for high-dimensional text data [41], we also employ the linear kernel function for 20-newsgroup data. On the basis of the fact that the manifold assumption of the face image data set is much stronger than the text data, we employ Laplacian kernel function for face image data.

In TSSL_BD method, we initialized W as the MMC projection matrix and empirically set the learning rate as $\eta(k) = \eta(0)/k$. The intuition is that in the forepart, we give large step sizes for iterations because the initial of W is far away from the optimal solution, and in the afterward, we give small step sizes for iterations and let it gradually approach to the optimal solution.

4.3 Overall comparison results

In this section, we perform three groups of experiments. The first group is the classification accuracies of various methods on both of the text data and face image data. The second group is the convergence property of TSSL methods. The last group is the sensitivity of the regularization parameters.

Fig. 3 The convergence property of TSSL methods

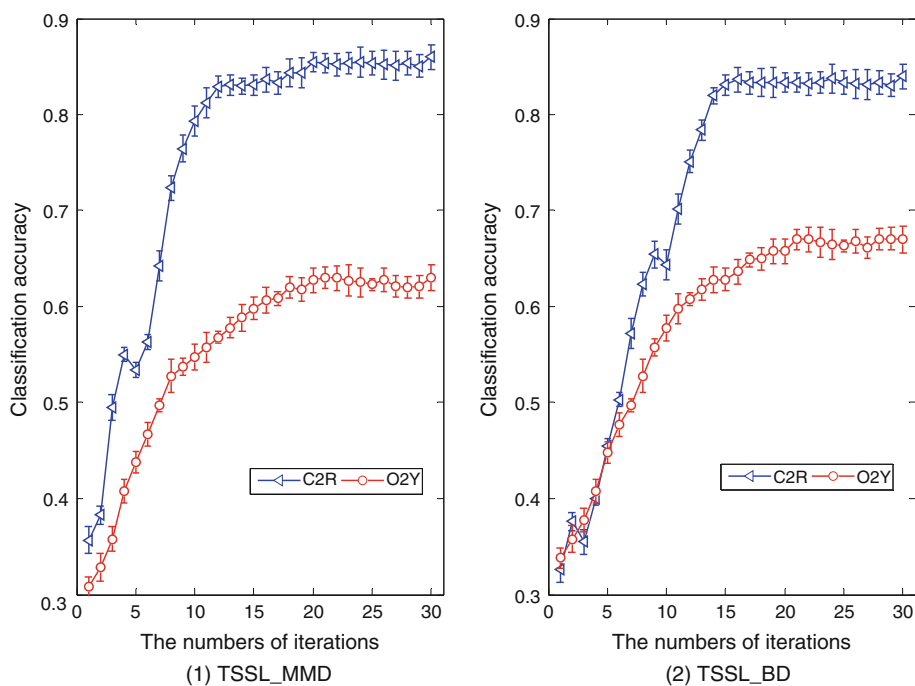
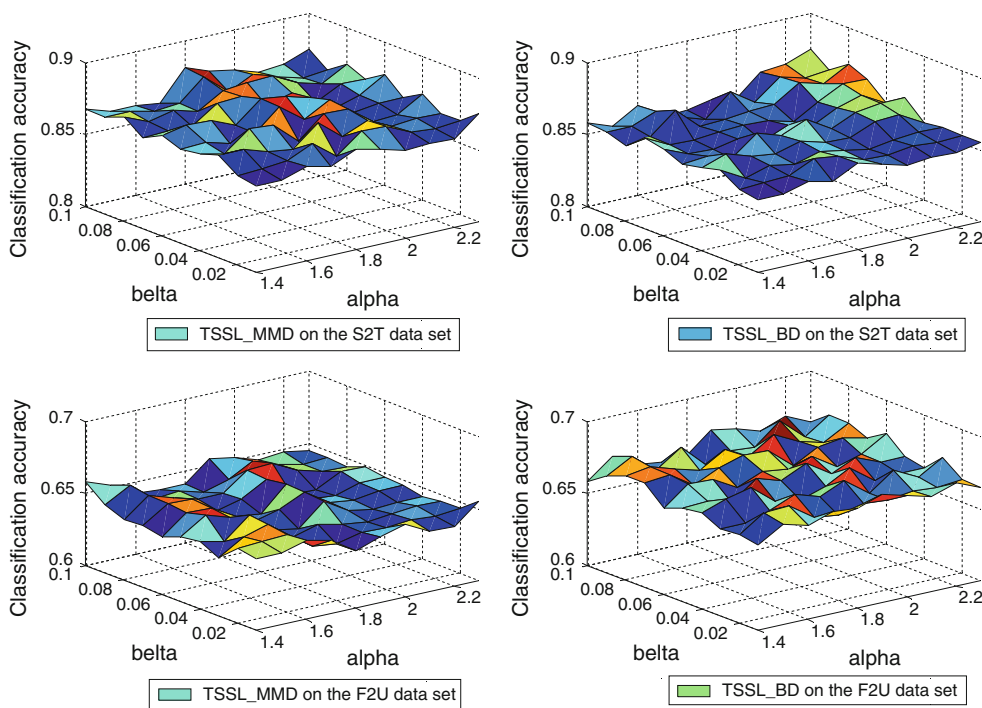


Fig. 4 Sensitivity analysis of the TSSL parameters



4.3.1 Classification accuracies of various methods

Firstly, on the text data, the number of dimensionality of the subspace varies from 5 to 40, and on the face image data, the number of dimensionality of the subspace varies from 10 to 80. Figures 1 and 2 show the performances of various methods on the text data sets and the face image data sets separately.

Secondly, we fix the dimensionality of the subspace at 20 on the text data and fix the dimensionality of subspace at 40 on the face image data, and then we can get the classification accuracies of various methods in Table 4.

From Figs. 1 and 2 and Tables 4, 5 and 6, we can see that our TSSL methods can outperform other methods on all of the text data sets and the face image data sets. There are

several observations. The numbers showed in bold are the best results among the eight methods.

1. The SVM method on text data sets and PCA or MMC methods on the face image data sets perform worse than other transfer learning methods because neither of them takes the distribution difference of source and target domain into consideration. The worst method on the face image data set is SCL, maybe SCL is developed for Natural Language Processing (NLP), and it cannot be suitable for image data well. MMC performs better than PCA because it considers more discriminant structure of face image data set, and that is why we select MMC as the base subspace learning method in the following methods.
2. On some text data sets, such as C2R and C2T, PCA sometimes performs as well as traditional transfer learning methods. This is because these two data sets may have more similarities than others, so they need not transfer to each other.
3. The representative transfer learning methods, MMDE, TCA, and TSL, perform much better than PCA, but worse than our TSSL methods. The reason is that the previous methods only consider the distribution difference of source and target domain data, but in our TSSL methods, we also enforce sparse regularization on the objective function to get better results, and then it can transfer more useful information from source domain to target domain.
4. There is another observation that TSSL_MMD is better than TSSL_BD on the text data, but worse on the face image data. Maybe, the MMD distance criterion is more suitable for text data, and Bregman divergence distance criterion is more suitable for face image data. This is an open problem, and we will focus on it in the future.
5. The classification accuracy results on the face image data are lower than text data sets because there is less commonality between the source and target domains on the face image data sets.

4.3.2 Convergence property

In this section, we test the convergence property of our TSSL methods. We select one text data set C2R and one face image data set O2Y as the basic databases to the experiment.

The results in Fig. 3 showed that both of TSSL_MMD and TSSL_BD can converge fast and the number of iteration is <20 .

4.3.3 Sensitivity analysis of the TSSL parameters

There are two parameters α and β in TSSL methods to discuss. In intuition, when we set α larger, then the sparse

regularization favors larger numbers of zero rows in the projection matrix W , which makes the source and target domains share more information in the common subspace. But if we set α too large, the most rows of W will be zeros, which is not suitable for transfer either. Similarly, when we set β larger, the distribution difference between source and target domains will be smaller. But if we set β too large, there will be less information can be transferred from source domain to target domain.

In a word, we should select duly parameters to make transfer learning more effectively. We first determine two parameters α and β of TSSL method by grid search and then change them within certain ranges.

The main procedure is as follows. Firstly, we fix $\alpha = 1$ and search for the best β value based on the validation set in the range of $[10^{-5}, 10^5]$. Then, we fix β and search the best value in the range of $[10^{-1}, 10^1]$. Finally, we let α vary from 1.4 to 2.3, let β vary in the range of $[10^{-2}, 10^{-1}]$. The classification accuracies with different α and β on the S2T data set and the F2U data set are shown in Fig. 4. As seen from Fig. 4, when the two parameters are changed within a certain range, the performance of TSSL changes within a certain range.

5 Conclusion and discussion

In this paper, we proposed a general framework for transfer sparse subspace learning, which is suitable for different assumptions on the divergence measures of the data distributions, such as MMD, Bregman divergence, and K–L divergence. To overcome over-fitting problems, we employ sparse regularization on the objective function and give different solutions to the problems based on different distribution distance estimation criteria. Experiments on both text data sets and face image data sets verify the efficiency and effectiveness of the proposed TSSL methods.

In the future, we plan to extend our TSSL framework to a multi-source-domain setting. We also plan to consider the heterogeneous transfer learning (HTL), which is a more challenging transfer learning setting. HTL considers not only the difference distributions between the source and target domains but also the different feature spaces between them.

There is another profound insight direction. In this paper, we used unsupervised divergence estimation terms and ignored the labeled target domain data. In future work, we will find another divergence estimation regularization that can make full use of the supervised information of target domain data and finally improve the accuracy of distribution divergence estimation.

Acknowledgments We would like to thank Sinno Jialin Pan and Sisi for providing the code of transfer component analysis and transfer subspace learning. We would like to express our appreciations to the editors and reviewers for their contributions in improving the quality of our paper. We gratefully acknowledge the supports from National Natural Science Foundation of China, under Grant No. 60975038 and Grant No. 61005003.

References

1. Yan S, Xu D, Zhang B, Zhang H, Yang Q, Lin S (2007) Graph embedding and extensions: a general framework for dimensionality reduction. *IEEE Trans Pattern Anal Mach Intell* 29(1):40–51
2. Belhumeur P, Hespanha J, Kriegman D (1997) Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Trans Pattern Anal Mach Intell* 19(7):711–720
3. Li H, Jiang T, Zhang K (2006) Effective and robust feature extraction by maximum margin criterion. *IEEE Trans Neural Netw* 17(1):157–165
4. He X, Niyogi P (2003) Locality preserving projections. In: Proceedings of the annual conference on advances in neural information processing systems (NIPS-03)
5. Zhang Y, d'Aspremont A, Ghaoui L (2010) Sparse PCA: convex relaxations, algorithms and applications, handbook on semidefinite, cone and polynomial optimization
6. Zou H, Hastie T, Tibshirani R (2004) Sparse principle component analysis. Technical report, Statistics Department, Stanford University
7. Moghaddam B, Weiss Y, Avidan S (2005) Spectral bounds for sparse PCA: exact and greedy algorithms. In: Proceedings of the annual conference on advances in neural information processing systems (NIPS-05)
8. Moghaddam B, Weiss Y, Avidan S (2006) Generalized spectral bounds for sparse LDA. In: Proceedings of the 23rd international conference on Machine learning (ICML-06), pp 641–648
9. Cai D, He X, Han J (2007) Spectral regression: a unified approach for sparse subspace learning. In: Proceedings of 2007 international conference on data mining (ICDM-07), Omaha
10. Caruana R (1997) Multitask learning. *Mach Learn* 28(1):41–75
11. Tikhonov AN (1963) Regularization of incorrectly posed problems. *Soviet Math Dokl* 4:1624–1627
12. Belkin M, Niyogi P, Sindhvani V (2006) Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. *J Mach Learn Res* 7:2399–2434
13. Ando RK, Zhang T (2006) Learning on graph with Laplacian regularization, advances in neural information processing systems (NIPS-06), vol 19. MIT Press, Cambridge, pp 25–33
14. Bradley P, Mangasarian O (1998) Feature selection via concave minimization and support vector machines. In: Proceedings of the 15th international conference on machine learning (ICML-98)
15. Wang L, Zhu J, Zou H (2007) Hybrid huberized support vector machines for microarray classification. In: Proceedings of the 24th international conference on machine learning (ICML-07)
16. Obozinski G, Taskar B, Jordan M (2006) Multi-task feature selection. Technical report, Department of Statistics, University of California, Berkeley
17. Argyriou A, Evgeniou T, Pontil M (2007) Multi-task feature learning. In: Proceedings of the annual conference on advances in neural information processing systems (NIPS-07), pp 41–48
18. Gu Q, Li Z, Han J (2011) Joint feature selection and subspace learning. In: The 22nd international joint conference on artificial intelligence (IJCAI-11), Barcelona
19. Ding C, Zhou D, He X, Zha H (2006) R1-PCA: rotational invariant l1-norm principal component analysis for robust subspace factorization. In: Proceedings of the 23rd international conference on machine learning (ICML-06)
20. Liu J, Ji S, Ye J (2009) Multi-task feature learning via effective L2,1-norm minimization. In: The conference on uncertainty in artificial intelligence (UAI-09)
21. Nie F, Huang H, Cai X, Ding C (2010) Effective and robust feature selection via joint l2,1-norms minimization. In: Proceedings of the annual conference on advances in neural information processing systems (NIPS-10)
22. Pan SJ, Yang Q (2010) A survey on transfer learning. *IEEE Trans Knowl Data Eng* 22(10):1345–1359
23. Sugiyama M, Nakajima S, Kashima H, Buenau PV, Kawanabe M (2008) Direct importance estimation with model selection and its application to covariate shift adaptation. In: Proceedings of the 20th annual conference on neural information processing systems (NIPS-08), Vancouver
24. Dai W, Yang Q, Xue G, Yu Y (2007) Boosting for transfer learning. In Proceedings of the 24th international conference on machine learning (ICML-07), New York, pp 193–200
25. Eaton E, desJardins M (2009) Set-based boosting for instance level transfer. In Proceedings of the 2009 IEEE international conference on data mining workshops (ICDMW-09), Washington, pp 422–428
26. Pardoe D, Stone P (2010) Boosting for regression transfer. In: Proceedings of the 27th international conference on Machine learning (ICML-10), pp 863–870
27. Yao Y, Doretto G (2010) Boosting for transfer learning with multiple sources. In: The 24th IEEE conference on computer vision and pattern recognition (CVPR-10), pp 1855–1862
28. Lawrence ND, Platt JC (2004) Learning to learn with the informative vector machine. In: Proceedings of the 21st international conference on machine learning (ICML-04). ACM, Banff
29. Tong B, Gao J, Thach N, Suzuki E (2011) Gaussian process for dimensionality reduction in transfer learning. In: Proceedings of the 11th SIAM international conference on data mining (SDM-11), pp 783–794
30. Gao X, Wang X, Li X, Tao D (2011) Transfer latent variable model based on divergence analysis. *Pattern Recogn* 44(10–11):2358–2366
31. Mihalkova L, Mooney RJ (2008) Transfer learning by mapping with minimal target data. In: Proceedings of the AAAI-2008 workshop on transfer learning for complex tasks, Chicago
32. Davis J, Domingos P (2008) Deep transfer via second-order markov logic. In: Proceedings of the AAAI-2008 workshop on transfer learning for complex tasks, Chicago
33. Arnold A, Nallapati R, Cohen W (2007) A comparative study of methods for transductive transfer learning. In: Proceedings of the seventh IEEE international conference on data mining workshops (ICDMW-07), Washington, pp 77–82
34. Daum'e H III (2007) Frustratingly easy domain adaptation. The association for computational linguistics (ACL-2007)
35. Blitzer J, Dredze M, Pereira F. Biographies, bollywood, boomboxes and blenders: domain adaptation for sentiment classification. In: Association for computational linguistics, Prague
36. Blitzer J, McDonald R, Pereira F (2006) Domain adaptation with structural correspondence learning. In: Proceedings of the 2006 conference on empirical methods in natural language processing (EMNLP-06), Association for Computational Linguistics, Stroudsburg, pp 120–128
37. Krupka E, Tishby N (2007) Incorporating prior knowledge on features into learning. In: Proceedings of the 11th international conference on artificial intelligence and statistics, San Juan
38. Satpal S, Sarawagi S (2007) Domain adaptation of conditional probability models via feature subsetting. In: Proceedings of the 11th European conference on principles and practice of knowledge discovery in databases (PKDD-2007), Berlin, pp 224–235

39. Tu W, Sun S (2011) Transferable discriminative dimensionality reduction. In: Proceedings of the ICTAI, pp 865–868
40. Tu W, Sun S (2012) Subject transfer framework for EEG classification. *Neurocomputing* 82:109–116
41. Pan SJ, Kwok JT, Yang Q (2008) Transfer learning via dimensionality reduction. In: Proceedings of the 23rd AAAI conference on artificial intelligence, Chicago (AAAI-08), Illinois, pp 677–682
42. Pan SJ, Tsang IW, Kwok JT, Yang Q (2009) Domain adaptation via transfer component analysis. In: Proceedings of the 21st international joint conference on artificial intelligence (IJCAI-09), Pasadena
43. Borgwardt K, Gretton A, Rasch M, Kriegel H, Schölkopf B, Smola A. Integrating structured biological data by kernel maximum mean discrepancy. In: Proceedings of the 14th international conference on intelligent systems for molecular biology, pp 49–57
44. Steinwart I (2001) On the influence of the kernel on the consistency of support vector machines. *J Mach Learn Res* 2:67–93
45. Quanz B, Huan J, Mishra M (2011) Knowledge transfer with low-quality data: a feature extraction issue. In: Proceedings of the IEEE international conference on data engineering (ICDE-11), Hannover
46. Quanz B, Huan J (2009) Large Margin Transductive Transfer Learning. In: Proceedings of the 18th ACM conference on information and knowledge management (CIKM-09), Hong Kong, pp 1327–1336
47. Ren J, Liang Z, Hu S (2010) Multiple kernel learning improved by MMD. *ADMA* (2):63–74
48. Zhang Z, Zhou J (2012) Multi-task clustering via domain adaptation. *Pattern Recogn* 45(1):465–473
49. Uguroglu S, Carbonell J (2011) Feature selection for transfer learning. *ECML/PKDD* 3:430–442
50. Duan L, Tsang I, Xu D (2012) Domain transfer multiple kernel learning. *IEEE Trans Pattern Anal Mach Intell* 34(3):465–479
51. Bregman L (1967) The relaxation method of finding the common points of convex sets and its application to the solution of problems in convex programming. *USSR Comput Mathe Mathe Phys* 7:200–217
52. Zhang J, Zhang C (2010) Multitask Bregman clustering. In: Proceedings of the 25th AAAI conference on artificial intelligence, Chicago (AAAI-10), pp 655–660
53. Si S, Tao D, Geng B (2010) Bregman divergence-based regularization for transfer subspace learning. *IEEE Trans Knowl Data Eng* 22(7):929–942
54. Si S, Tao D, Chan K (2010) Evolutionary cross-domain discriminative hessian eigenmaps. *IEEE Trans Image Process* 19(4):1075–1086
55. Si S, Tao D, Wang M, Chan K (2012) Social image annotation via cross-domain subspace learning. *Multimed Tools Appl* 56(1):91–108
56. Wu L, Hoi S, Jin R, Zhu J, Yu N (2012) Learning Bregman distance functions for semi-supervised clustering. *IEEE Trans Knowl Data Eng* 24(3):478–491
57. Gao X, Wang X, Li X, Tao D (2011) Transfer latent variable model based on divergence analysis. *Pattern Recogn* 44(10–11):2358–2366
58. Zhang J, Zhang C (2011) Multitask Bregman clustering. *Neurocomputing* 74(10):1720–1734
59. Schölkopf B, Smola A, Müller K-R (1998) Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput* 10(5):1299–1319
60. Parzen E (1962) On estimation of a probability density function and mode. *Ann Math Stat* 33(3):1065–1076
61. Dai W, Yang Q, Xue G-R, Yu Y (2009) EigenTransfer: a unified framework for transfer learning. In: Proceedings of the 26th international conference on machine learning (ICML-09)
62. <http://www.zjucadcg.cn/dengcai/Data/FaceData.html>
63. Phillips JP, Moon H, Rizvi SA, Rauss PJ (2000) The FERET evaluation methodology for face-recognition algorithms. *IEEE Trans Pattern Anal Mach Intell* 22(10):1090–1104
64. <http://images.ee.umist.ac.uk/danny/database.html>