Pattern Recognition 44 (2011) 2358–2366

Contents lists available at ScienceDirect

Pattern Recognition

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

Transfer latent variable model based on divergence analysis

Xinbo Gao [a], Xiumei Wang [a], Xuelong Li [b,*], Dacheng Tao [c]

[a] School of Electronic Engineering, Xidian University, Xi'an 710071, P. R. China
[b] Center for OPTical IMagery Analysis and Learning (OPTIMAL), State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, Shaanxi, P. R. China
[c] School of Computer Engineering, Nanyang Technological University, 50 Nanyang Avenue, B/k N4, 639798, Singapore

ARTICLE INFO

Available online 19 June 2010

Keywords:
Dimensionality reduction
Latent variable model
Transfer learning
Bregman divergence

ABSTRACT

Latent variable models are powerful dimensionality reduction approaches in machine learning and pattern recognition. However, this kind of methods only works well under a necessary and strict assumption that the training samples and testing samples are independent and identically distributed. When the samples come from different domains, the distribution of the testing dataset will not be identical with the training dataset. Therefore, the performance of latent variable models will be degraded for the reason that the parameters of the training model do not suit for the testing dataset. This case limits the generalization and application of the traditional latent variable models. To handle this issue, a transfer learning framework for latent variable model is proposed which can utilize the distance (or divergence) of the two datasets to modify the parameters of the obtained latent variable model. So we do not need to rebuild the model and only adjust the parameters according to the divergence, which will adopt different datasets. Experimental results on several real datasets demonstrate the advantages of the proposed framework.

1. Introduction

With the increase of data acquisition capability, dimensionality reduction [2,4–9,17,22] plays more and more important role in machine learning and pattern recognition, which can catch the intrinsically low dimensional structure embedding in a very high-dimensional space. As the most classic dimensionality reduction algorithm, principal component analysis (PCA) was proposed by Hotelling [1]. PCA is a representative linear dimensionality reduction model and has been extensively used in the last few decades [10–13], and recent efforts are also made in various applications [18–20]. However, traditional PCA is inadequate for dealing with the nonlinear case. Therefore, many nonlinear dimensionality reduction methods have been proposed for the purpose of finding the nonlinear compact representation of the high-dimensional data [14–16]. Examples include locally linear embedding (LLE) [17,34], the Laplacian eigenmap (LE) algorithm [21], the Isomap algorithm [22,23]. LLE and LE obtain the nonlinear embedding structure with a locally linear technique. The Isomap algorithm aims to catch a global geometric framework for nonlinear dimensionality reduction. The approaches mentioned may work well if they are applied to appropriate datasets. However, if the number of the observed samples is relatively small or the samples are particularly noisy, neither of the linear dimensionality reductions nor the nonlinear embedding algorithms could get good performance.

To handle with the small sample size and noisy problem, probabilistic latent variable models (LVMs), which establish mappings from the low-dimensional latent variables space to the high-dimensional observed space, was developed as a popular framework used in dimensionality reduction. The LVMs estimate the joint density of the observed dataset through treating the latent coordinates and mappings as parameters. Representative LVMs include probabilistic principal component analysis (PPCA) [24] and Gaussian process latent variable model (GP-LVM) [25,26]. PPCA belongs to the linear probabilistic model and GP-LVM is a nonlinear generalization of PPCA. Both of them are conducted to tackle the problems of small sample size. In particularly, the GP-LVM attracts much more attentions for its nonlinear generalizing property than PPCA [27,28].

The LVMs work well under a necessary and strict assumption, that is, the training samples and testing samples are drawn from the same domain and obey the identical distribution. When they come from different domains, the training samples and testing samples would be not independent identical distribution (i.i.d.). Therefore, the performance of the latent variable model will be degraded because the parameters of the training model may not suit for the testing dataset. One attempt to this problem is applying a transfer learning framework to LVMs, which can effectively deal with the different domains problem, i.e., not i.i.d. [29–31]. The advantages of transfer learning have been widely demonstrated by Web-document classification [29] and language processing [32].

* Corresponding author.
E-mail address: xuelong_li@opt.ac.cn (X. Li).

0031-3203/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2010.06.013

However, the transfer learning for dimensionality reduction always focuses on the determined dimensionality reduction methods [31], such as PCA, locality preserving projections (LPP) [3], marginal Fisher analysis (MFA) [33], etc. There is little effort of transfer learning that has been made for the probabilistic dimensionality reduction, i.e., PPCA, GP-LVM. The transfer learning framework for determined dimensionality reduction methods pays attention to updating the projection matrix which is the parameter in linear dimensionality reduction algorithms. It cannot used to deal with probabilistic dimensionality reduction algorithms for two reasons. On the one hand, the framework does not consider the noises in the datasets. In cross-domain tasks, the distributions of the datasets are different and the noises may be different too, so the transfer learning framework should be take the noises into consideration. On the other hand, in some probabilistic dimensionality reduction algorithm, such as GP-LVM, there is no projection matrix but hyper-parameters of kernel function chosen in the model. Therefore, the transfer learning for GP-LVM should be focused on how to update the hyper-parameters. Furthermore, the framework of transfer learning for determined dimensionality reduction methods cannot be directly introduced into the LVMs for it only pays attention to updating the projection matrix.

For this purpose, we present a transfer learning framework based on divergence analysis for LVMs, i.e., the transfer latent variable model learning (TLVM), which combines the advantages of the LVMs and transfer learning. The divergence defines a general distance which could measure the margin between two datasets, i.e., the training dataset and the testing dataset. With the divergence term, the LVMs do not need to be retrained when confront with the cross-domain tasks, while we just need to adjust the parameters according to the degree of divergence. Therefore, just like LVMs, the TLVMs can handle the sparse/small training sample size well. Besides, compared with the traditional LVMs, TLVMs also have the following advantages: (a) through transferring the knowledge and priori information of the training dataset to the parameters, the transfer learning can speed up the learning process and meanwhile improve the accuracy of the model; (b) the transfer learning framework can be used for cross-domain learning tasks.

To evaluate the performance of the TLVMs for reconstruction and recognition in cross-domain, handwritten digits datasets and several face datasets, e.g., (Olivetti Research Laboratory) [46], Yale [47] and YaleB [48] are used in the experiments. The extensive experimental results validate the effectiveness and efficiency of the proposed TLVM in cross-domain reconstruction and recognition tasks. For handwritten digits datasets, the training samples are drawn from one digit, while the testing datasets come from the other digits. For the face datasets, the training date set is drawn from Yale data and the testing samples come from ORL and YaleB datasets. Experimental results of two datasets confirm the significant performance on the different domains of the proposed method.

The remainders of this paper are organized as follows. Section 2 introduces the background and related works. Section 3 details the transfer learning with LVMs. Section 4 presents the experimental results on different databases. The final one is conclusion.

2. Background

In this section, we will give a brief review for the latent variable model (LVM), especially for the Gaussian process latent variable model (GP-LVM).

2.1. Latent variable model

More formally, let $Y=[y_1,\ldots,y_N]^T$ be the matrix denoting a set of N observed samples, i.e., the high-dimensional dataset for training. Each object y_i is described by a D-dimensional feature vector with $y_i \in R^D$. We denote the latent variables as $X=[x_1,\ldots,x_N]^T$, where the column vector x_i is the low-dimensional representation for the sample y_i with $x_i \in R^L$, $L<D$.

The latent variable model is a powerful approach to dimensionality reduction by estimating probabilistic density. It is one of the major issues in pattern recognition and machine learning [35,36]. By defining the mapping function from the latent variables to the observed samples as $f:X\rightarrow Y$, a joint distribution over visible and latent variables can be obtained. Through maximizing the joint density of the observed dataset Y, we can determine the latent coordinates X and the parameters of the mapping function f.

2.2. Gaussian process latent variable model

Similar to LVMs, the GP-LVM models joint density of the observed dataset Y with the latent variables X and the mapping function f. Different from the traditional LVMs, the output of mapping function obeys Gaussian process, i.e., $f(x)\sim GP(m(x), k(x,x'))$ which represents that the mapping function f is specified by its mean function $m(x)$ and covariance function $k(x, x')$ [37]. Generally, we define the mean function $m(x)=0$.

If a Gaussian process prior is imposed on the mapping function f, the likelihood of a set of observations can be denoted as

$$P(Y|X,\theta)=\prod_{d=1}^{D}\frac{1}{(2\pi)^{N/2}|K|^{1/2}}\exp\left(-\frac{1}{2}tr(y_{:,d}{}^{T}K^{-1}y_{:,d})\right). \quad (1)$$

The covariance, or the linear kernel, is given by $K=XX^T+\theta_{white}E$, where θ_{white} denotes noise variance and E is an identity matrix. It is also possible to consider a nonlinear kernel here, e.g., the RBF kernel, i.e.,

$$k(x_i,x_j)=\theta_{rbf}\exp\left(-\frac{\gamma}{2}(x_i-x_j)^T(x_i-x_j)\right)+\theta_{white}\delta_{ij}, \quad (2)$$

where $k(x_i,x_j)$ is the entry in the ith row and the jth column of the covariance matrix, K and δ_{ij} is the Kronecker delta function. θ_{rbf} and γ denote RBF process variance and the kernel width, respectively. The hyper-parameter θ is collection of the kernel parameters θ_{rbf}, γ and θ_{white}, i.e., $\theta=[\theta_{rbf}, \gamma, \theta_{white}]$.

As shown in Eq. (1), the GP-LVM represents the high-dimensional dataset Y with a Gaussian process mapping function f from the latent variables X. The GP-LVM can achieve good results under the assumption that all the observed data come from the same domain. When the testing samples are drawn from another domain, the assumption that all the samples are i.i.d. would be not satisfied. Then the model needs to be rebuilt which leads to additional learning and training. In order to avoid the case and utilize the information in training dataset for learning the testing samples, we introduce a transfer learning framework in the following section, which can modify the parameters of the model according to the distances between the training and testing samples.

3. Transfer learning framework for the latent variable model

In this section, a transfer learning framework will be incorporated with the LVMs. This framework is based on the Bregman divergence, which can address the problem aforementioned issue. We will firstly give a brief introduction of Bregman divergence, then proposed TLVM in detail.

3.1. Bregman divergence

Bregman divergences define a generalization distance to measure the discrepancy between distributions [38,39]. It has

been testified to be effective and efficient in clustering [40] and nearest neighbor retrieval [41].

Definition 1. *Bregman divergence between vectors*: Let $\Phi:\Omega\rightarrow R$ be a strictly convex function defined on a closed convex set Ω, and Φ is continuously differentiable. Then Bregman divergence associated with Φ for vectors p, $q\in\Omega$ is

$$d_\Phi(p,q)=\Phi(p)-\Phi(q)-(\nabla\Phi(q),(p-q)), \tag{3}$$

where $\nabla\Phi(q)$ denotes the first order difference of the Φ at point p.

The Bregman divergence can be interpreted as the distance between a function and its first-order Taylor expansion, as shown in Fig. 1.

Besides the vectors, Bregman divergences can also be utilized to measure the distance between matrices, functions, and distributions [42]. In this paper, the LVMs will be extended to deal with the cross-domain tasks, therefore, the probability density functions of different domains are also different. Let $p(y)$ and $q(y)$ represent the probability density functions of the training samples Y and testing samples Y_t, respectively. The Bregman divergences defined between the distributions are given as follows:

Definition 2. *Bregman divergence between distributions*: The Bregman divergence between two given distribution functions $p(y)$ and $q(y)$ under a certain measure μ can be obtained as follows:

$$D_\Phi(p,q)=\int d_\Phi(p(y),q(y))\,d\mu(y). \tag{4}$$

The definition of $d_\Phi(\cdot,\cdot)$ is given in Eq. (3). There are several commonly used examples of Bregman divergence which are defined according to the different function Φ [43].

Examples.

(a) Let $\Phi(y)=y^2$, the divergence is the measure of energy

$$D_\Phi(p,q)=\|p(y),q(y)\|^2_{L^2(\mu)}. \tag{5}$$

(b) Let $\Phi(y)=-\log y$, the divergence is *Itakura–Saito* distortion

$$D_\Phi(p,q)=\int\left(\log\frac{q(y)}{p(y)}+\frac{p(y)}{q(y)}-1\right)d\mu(y). \tag{6}$$

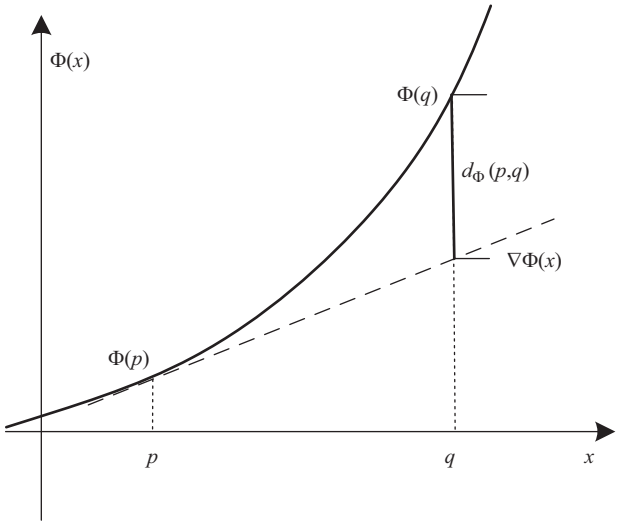

Fig. 1. The Bregman divergence between p and q.

(c) Let $\Phi(y)=y\log y+(1-y)\log(1-y)$, the divergence is *Bernoulli* entropy

$$D_\Phi(p,q)=\int\left(p(y)\log\frac{q(y)}{p(y)}+(1-p(y))\log\frac{1-p(y)}{1-q(y)}\right)d\mu(y). \tag{7}$$

(d) Let $\Phi(y)=y\log y$, the divergence is the *generalized Kullback–Leibler* divergence

$$D_\Phi(p,q)=\int\left(p(y)\log\frac{q(y)}{p(y)}-p(y)+q(y)\right)d\mu(y). \tag{8}$$

If $p(y)$and $q(y)$ represent the probability density functions, that is, $\int p(y)\,dy=1$ and $\int q(y)\,dy=1$, the divergence will be the *Kullback–Leibler* (KL) divergence. In general, KL divergence is a special case of Bregman divergences.

As shown in examples, Bregman divergences are generalizations of the squared Euclidean distance that they all share similar properties, such as non-negative and not symmetric. In the next section, the Bregman divergence will be utilized to measure the distance between two distributions.

3.2. Transfer learning framework for the latent variable model

According to Eq. (1), the likelihood function for training dataset Y can be presented as

$$P(Y|X)=\frac{1}{(2\pi)^{DN/2}|K|^{D/2}}\exp\left(-\frac{1}{2}tr(K^{-1}YY^T)\right). \tag{9}$$

The corresponding negative log-likelihood is then

$$L(X,\theta)=\frac{DN}{2}\ln 2\pi+\frac{D}{2}\ln|K|+\frac{1}{2}tr(K^{-1}YY^T), \tag{10}$$

where the latent variables X and hyper-parameter θ are considered as the parameters in the GP-LVM, which need to be determined by estimating the maximum-likelihood. The maximum-likelihood estimator for the hyper-parameter θ and the latent variables X is given by using the scaled conjugate gradient [44],

$$\{X,\theta\}=\underset{X,\theta}{\operatorname{argmin}}\{L(X,\theta)\}. \tag{11}$$

The model performs well when the samples from training and testing datasets are i.i.d. Let $Y_t=[y_{t1},\ldots,y_{tM}]^T$ denote the testing dataset and $X_t=[x_{t1},\ldots,x_{tM}]^T$, the corresponding variables for $Y_{testing}$ in low-dimensional space. If Y and Y_t are drawn from different distributions, the samples in two datasets will not be i.i.d. Sometimes, even the samples were drawn from the same kind of distributions, such as Gaussian distribution, the mean values and covariance matrices of training samples and testing samples would not be uniform, e.g., the two Gaussian distributions shown in Fig. 2, which have different mean values and covariance matrices.

Therefore a regularization based on Bregman divergence will be required for measuring the distance between Y and Y_t. Then a new transfer learning framework for the LVMs can be established by using the regularization,

$$\{X,\theta\}=\underset{X,\theta}{\operatorname{argmin}}\{L(X,\theta)+D(p(Y)\|p(Y_t))\}. \tag{12}$$

As listed in Section 3.1, the Bregman divergence possesses various representations according to the convex function Φ. And which modality should be chosen in the transfer learning will be determined by the distributions of the training and testing samples. Fig. 3 provides an illustration of the transfer

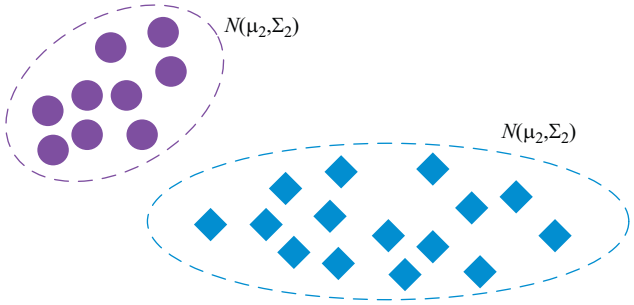

Fig. 2. Two Gaussian distributions with different means and covariance matrices.

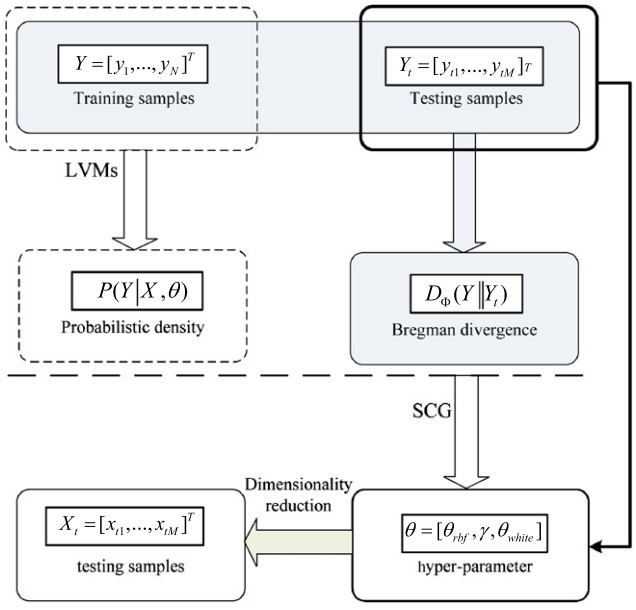

Fig. 3. The flowchart of TLVM.

learning framework with the regularization based on Bregman divergence.

Fig. 3 shows the framework of the transfer learning process. It can be divided into two steps. First, the hyper-parameters and the positions for training samples in the latent space are determined by the traditional GP-LVM through maximizing the likelihood which can be calculated by Eq. (1). Secondly, the divergence between the training samples and testing samples are calculated, and then the hyper-parameters are updated according to the divergence, the position of the testing samples in the latent space are obtained finally. The optimization method used in the framework is scaled conjugate gradient (SCG) algorithm [44]. SCG is a variation of standard conjugate gradient algorithm. It uses the second order information but requires only O(N) memory usage, so it is an efficient optimization method.

As well known, the GP-LVM establishes the Gaussian process mapping from the latent space to each dimension of the observed space, that is to say, the GP-LVM supposes that each dimension of the dataset is drawn from the same Gaussian distribution, i.e., $y_{:,d}\sim N(0,K)$ $(d=1,\ldots,D)$. Then for the testing dataset Y_t, let $X_t=[x_{t1},\ldots,x_{tM}]^T$ denote the low-dimensional variables corresponding to Y_t in the latent space. Under the GP-LVM framework, the likelihood function of the testing dataset Y_t can be represented as

$$P(Y_t|X_t)=\frac{1}{(2\pi)^{DM/2}|K_t|^{D/2}}\exp\left(-\frac{1}{2}tr(K_t^{-1}Y_tY_t^T)\right). \tag{13}$$

The distribution in each dimension of the testing dataset Y_t will also be a Gaussian, i.e., $Y_{t(:,d))}\sim N(0,K_t)$ $(d=1,\ldots,D)$.

To estimate the distance between the training and testing samples, we just need to measure the distance between two Gaussian distributions. As a special case of Bregman divergence, the KL divergence can effectively calculate the margin between two distributions. For two Gaussian distributions $Y_{:,d}\sim N(0,K)$ and $Y_{t(:,d)}\sim N(0,K_t)$, the KL divergence can be computed as

$$KL(Y_{:,d}\|Y_{t(:,d)})=\frac{1}{2}\ln|K_tK^{-1}|+\frac{1}{2}tr(K_t^{-1}(K-K_t)). \tag{14}$$

Thus the distance between the training and testing dataset is

$$\begin{aligned}D(p(Y)\|p(Y_t))&=\sum_{d=1}^{D}KL(Y_{(:,d)}\|Y_{t(:,d)})\\&=\frac{D}{2}\ln|K_tK^{-1}|+\frac{D}{2}tr(K_t^{-1}(K-K_t)).\end{aligned} \tag{15}$$

Then the transfer learning for the LVMs can be shown in a uniform framework,

$$\{X,\theta\}=\underset{X,\theta}{\operatorname{argmin}}\{f(X,\theta)\}. \tag{16}$$

The objective function $F(X,\theta)$ is equal to

$$\begin{aligned}F(X,\theta)&=L(X,\theta)+D(p(Y)\|p(Y_t))\\&=\frac{DN}{2}\ln 2\pi+\frac{D}{2}\ln|K|+\frac{1}{2}tr(K^{-1}YY^T)+\frac{D}{2}\ln|K_tK^{-1}|\\&\quad+\frac{D}{2}tr(K_t^{-1}(K-K_t)).\end{aligned} \tag{17}$$

Eq. (17) can be rewritten as

$$\begin{aligned}F(X,\theta)&=\frac{DN}{2}\ln 2\pi+\frac{D}{2}\ln(|K|\cdot|K_tK^{-1}|)+\frac{1}{2}tr(K^{-1}YY^T+DK_t^{-1}(K-K_t))\\&=\frac{DN}{2}\ln 2\pi+\frac{D}{2}\ln(|K_t|)+\frac{1}{2}tr(K^{-1}YY^T+DK_t^{-1}K+DE),\end{aligned} \tag{18}$$

where E is an identity matrix.

There is an inevitable instance that the number of the training samples is different with the testing samples, i.e., $M\neq N$. In this case, kernel matrices $K\in R^{N\times N}$ and $K_t\in R^{M\times M}$ $(M\neq N)$, and it makes no sense to calculate $K_t\ K^{-1}$ or $K-K_t$. To handle this issue, we utilize the informative vector machine (IVM) algorithm to extract a subset from the dataset [45]. That is, the IVM can represent the dataset by a subset I, which is an active set contained r samples, where $r\leq\min\{N,M\}$. The objective function can be represented as

$$F_I(X,\theta)=\frac{DN}{2}\ln 2\pi+\frac{D}{2}\ln(|K_{tI}|)+\frac{1}{2}tr(K_I^{-1}Y_IY_I^T+DK_{tI}^{-1}K_I+DE_I). \tag{19}$$

The transfer learning framework

$$\{X,\theta\}=\underset{X,\theta}{\operatorname{argmin}}\{F_I(X,\theta)\}. \tag{20}$$

The solution of Eq. (19) can be obtained by the scalar conjugate gradient algorithm.

3.3. Parameters optimization

In the previous subsection, we gave a transfer learning framework for the LVMs. The parameters include hyper-parameter θ and the latent variables X. To obtain the optimal

parameters, Eq. (19) can be optimized with respect to X by using the scalar conjugate gradient algorithm.

Firstly, the optimal hyper-parameter θ is calculated through the chain rule

$$\frac{\partial F_I(X,\theta)}{\partial \theta} = \frac{\partial F_I(X,\theta)}{\partial K}\frac{\partial K}{\partial \theta} + \frac{\partial F_I(X,\theta)}{\partial K_t}\frac{\partial K_t}{\partial \theta}. \tag{21}$$

The gradients with respect to the kernel matrices will be given as follows:

$$\frac{\partial F_I(X,\theta)}{\partial K_I} = \frac{1}{2}K_I^{-1}Y_I Y_I^T K_I^{-1} + K_{tI}^{-1},$$
$$\frac{\partial F_I(X,\theta)}{\partial K_{tI}} = DK_{tI}^{-1} + K_{tI}^{-1}K_I K_{tI}^{-1}. \tag{22}$$

The gradients of the kernel matrices with respect to the hyper-parameter θ can be computed according to the kernel function chosen in the transfer learning framework. If the RBF in Eq.(2) is selected, the gradients will be obtained:

$$\frac{\partial K}{\partial \theta_{rbf}} = K/\theta_{rbf},$$
$$\frac{\partial K}{\partial \gamma} = -\frac{1}{2}(XX^T)K,$$
$$\frac{\partial K}{\partial \theta_{white}} = E. \tag{23}$$

Secondly, similar to optimizing hyper-parameter, the optimal latent variables X can also be obtained through the chain rule. The only difference is that the latent variables should be divided into two parts during optimization, one part comes from the active set X_I, and the other part includes the rest variables X_B, $X_B = X - X_I$. For the variables in X_I, the optimizing steps is the same as the hyper-parameter

$$\frac{\partial F_I(X,\theta)}{\partial X_I} = \frac{\partial F_I(X,\theta)}{\partial K}\frac{\partial K}{\partial X_I} + \frac{\partial F_I(X,\theta)}{\partial K_t}\frac{\partial K_t}{\partial X_I}. \tag{24}$$

For arbitrary variable $x \in X_B$, the relationship between latent variable x and its corresponding sample y can be represented through the Gaussian process mapping function f. According to the active set X_I, the likelihood of sample y will be given by adopting a Gaussian distribution, i.e.,

$$p(y|x) = N(y|\mu,\sigma^2 I). \tag{25}$$

The mean μ and the variance σ^2 are represented, respectively, as

$$\begin{cases} \mu = Y_I^T K_I^{-1} k_I, \\ \sigma^2 = k(x,x) - k^T K_I^{-1} k_I, \end{cases} \tag{26}$$

where K_I denotes the kernel matrix developed from the training set, and k_I is the vector of covariance matrix between the test point y and active set points in subset I. The point x can be obtained through optimizing Eq.(26) with gradients-descent.

We apply the scaled conjugate gradient to obtain the optimal hyper-parameter θ for training. The detailed steps for training process will be given in Table 1.

As shown in Table 1, firstly the latent variables will be initialized through PCA and all the values in the hyper-parameter are set to 1. Then the latent variables and the hyper-parameters will be optimized alternatively. Finally, the TLVM can be established with the hyper-parameters which are obtained according to the likelihood and the Bregman divergence.

Table 1
The training process for TLVM.

Input: The high-dimensional training samples $Y \in R^{N\times D}$, testing samples $Y_t \in R^{M\times D}$, the number of training iterations *Iter*
Initialization: the latent variables $X \in R^{N\times L}$ through GP-LVM, the latent variables $X_t \in R^{M\times L}$ through GP-LVM, the hyper-parameter $\Theta = [1,1,1]$
Step1 For $i=1$ to *Iter*{
Step2 Calculate $L^{(i)}(X,\theta) = \frac{DN}{2}\ln 2\pi + \frac{D}{2}\ln|K| + \frac{1}{2}tr(K^{-1}YY^T)$
Step3 Calculate $D^{(i)}(p(Y)\|p(Y_t)) = \sum_{d=1}^{D} KL(Y_{(:,d)}\|Y_{t(:,d)})$
Step4 Optimize $\{\theta^{(i)}\} = \arg\min_{\theta}\{L^{(i)}(X,\theta) + D^{(i)}(p(Y)\|p(Y_t))\}$
Step5 Update $D^{(i)}(p(Y)\|p(Y_t)) = \sum_{d=1}^{D} KL(Y_{(:,d)}\|Y_{t(:,d)})$
Step 6 Optimize $\{X_t^{(i)}\} = \arg\min_{X}\{L^{(i)}(X,\theta) + D_t{}^{(i)}(p(Y)\|p(Y_t))\}$;
Check convergence: the stage of TLVM
converges if $Error(t) = \sum_{i=1}^{N}\|X_t^{(i)} - X_t^{(i-1)}\|^2 \le \varepsilon$ or $t > T$
}// For loop in step1
Output: the hyper-parameter θ and $X_t \in R^{M\times L}$

Fig. 4. Images sampled from Yale, ORL and YaleB.

The mapping function in TLVM is Gaussian process. Similar to Gaussian process regression, TLVM can be used for data reconstruction. As shown in Fig. 4, the latent variable x^* corresponding to observed sample y^* is obtained by the TLVM. The observed sample y^* can be reconstructed from x^* by using the Gaussian process regression. The reconstructed value is

$$\tilde{y} = Y_I^T K_I^{-1} k_I. \tag{27}$$

The right hand side of Eq. (27) is the mean value μ in Eq.(26). Thus, we can get the reconstruction error as $\|y^* - \tilde{y}\|$.

The recognition error is calculated by the nearest neighbor (NN) classifier in the latent space. That is, the observed sample y^* are projected into the latent space, i.e., x^*, and then recognized by the NN classifier in the latent space.

4. Experiments and analysis

In this section, we empirically investigate the performance of the proposed transfer learning algorithm on two kinds of real-world datasets. One collects three face databases, i.e., ORL, Yale and YaleB, and the other come from the USPS handwritten digits. The *ORL* face data contains 400 images, of which 40 individuals is selected as test-bed. For each of individual, there are ten different images are taken at different times, varying the lighting, facial expressions, as shown in Fig. 4. The Yale face data contain 165 images of 15 individuals and each has 11 images with different facial expressions or configurations. The YaleB face data contain a total of 38 (subjects) × 64 (illumination conditions) samples. The size of each cropped image for three datasets is 32×32 pixels.

The second database is the handwritten digits including 7921 samples of ten digits (from digit's '0' to digit's '9'), and each sample is described by 256 features [25] as shown in Fig. 7.

4.1. The face database

In order to testify the validity of the proposed TLVM-Based dimensionality reduction method, we compare the performance of the TLVM with the traditional LVM in two sides, e.g., the data reconstruction error (shown in Fig. 5) and recognition error (shown in Fig. 6). In traditional LVM learning, there is no consideration of the relationship between training and testing sets. While the transfer learning tasks could build on the cross-domain, and the distance between the training and testing data is considered for capturing the similar properties of the two datasets. In the following experiments, the number of active samples is 40 for each face database. For most dimensionality reduction algorithms, the dimensionality reduction results usually vary with the number of dimensions. Therefore, we study the performance of the proposed methods varying with different dimensions of the latent space in Fig. 5.

Fig. 5 shows the face reconstruction error versus the dimensions of the latent space. With considering the distances between training and testing dataset, the transfer learning framework works well under the condition that training and testing samples share common properties. TLVM outperforms the traditional LVM on two databases, especially for the ORL database. Although the YaleB face database is an extension of the Yale, the reconstruction error is larger than that on the ORL database. The reason is that the ORL database does not contain the variations of lighting

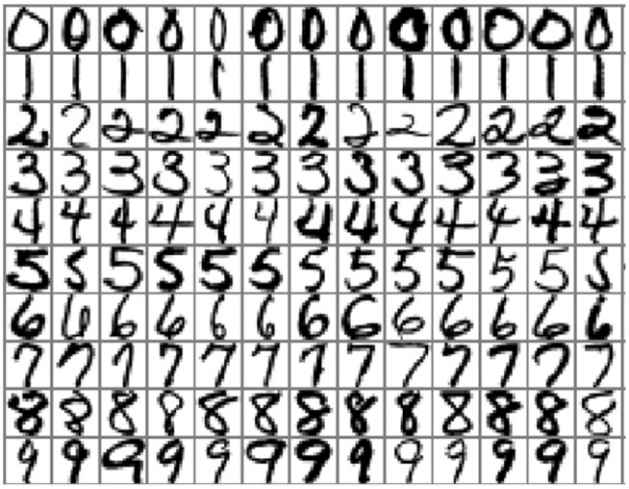

Fig. 7. Images sampled from handwritten digits.

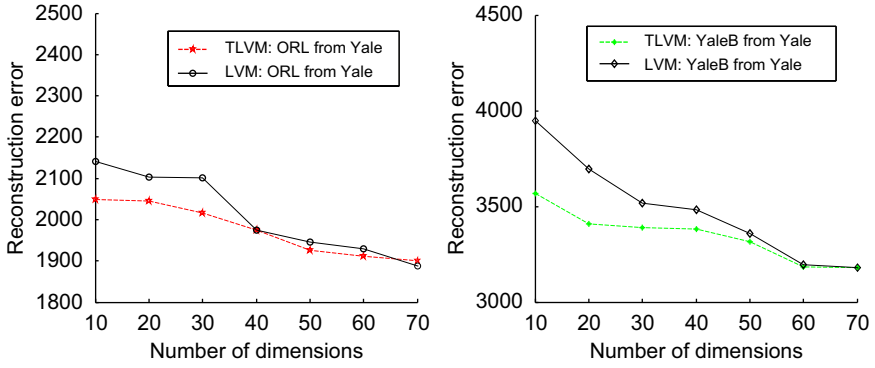

Fig. 5. The comparison of reconstruction error with different dimensions of the latent space on ORL and YaleB from Yale.

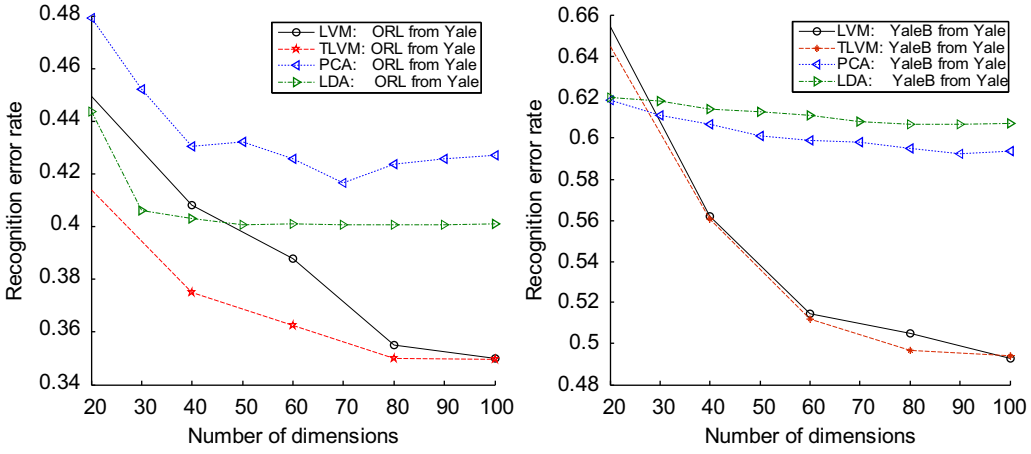

Fig. 6. The comparison of recognition error rate with different dimensions of the latent space on ORL and YaleB from Yale.

conditions, while the YaleB database contains strong variations of illumination and poses.

Fig. 6 presents the recognition error rate changing with the different dimension sizes of the latent space. The nearest neighborhood (NN) classifier is used for testing the recognition error rate. The experimental results confirm that the proposed TLVM framework actually outperforms the traditional LVM for cross-domain tasks. For the ORL face database, the performance of TLVM is improved obviously especially when the dimensions of the latent space are small. For the YaleB face database, TLVM could also acquire accurate recognition rate. However, as shown in the right subfigure of Fig. 6, it performs very similar to the traditional LVM. This is because the transfer information from training dataset, i.e., Yale face database, cannot represent so many variations in the YaleB face database which contains lighting conditions or postures.

In order to show the performance of the proposed method, we give comparison with other benchmark subspace methods in Fig. 6, such as PCA and LDA. It is obvious can be seen that advantage of the proposed method in face recognition. The recognition errors of PCA and LDA remain high and nearly unchanged with the increasing of number of dimension. The reason is that the projection directions trained from Yale are not suit for ORL or YaleB. There is no useful information transferred from Yale and help to learning ORL and YaleB. Although utilizing the label information, LDA cannot obtain lower recognition error

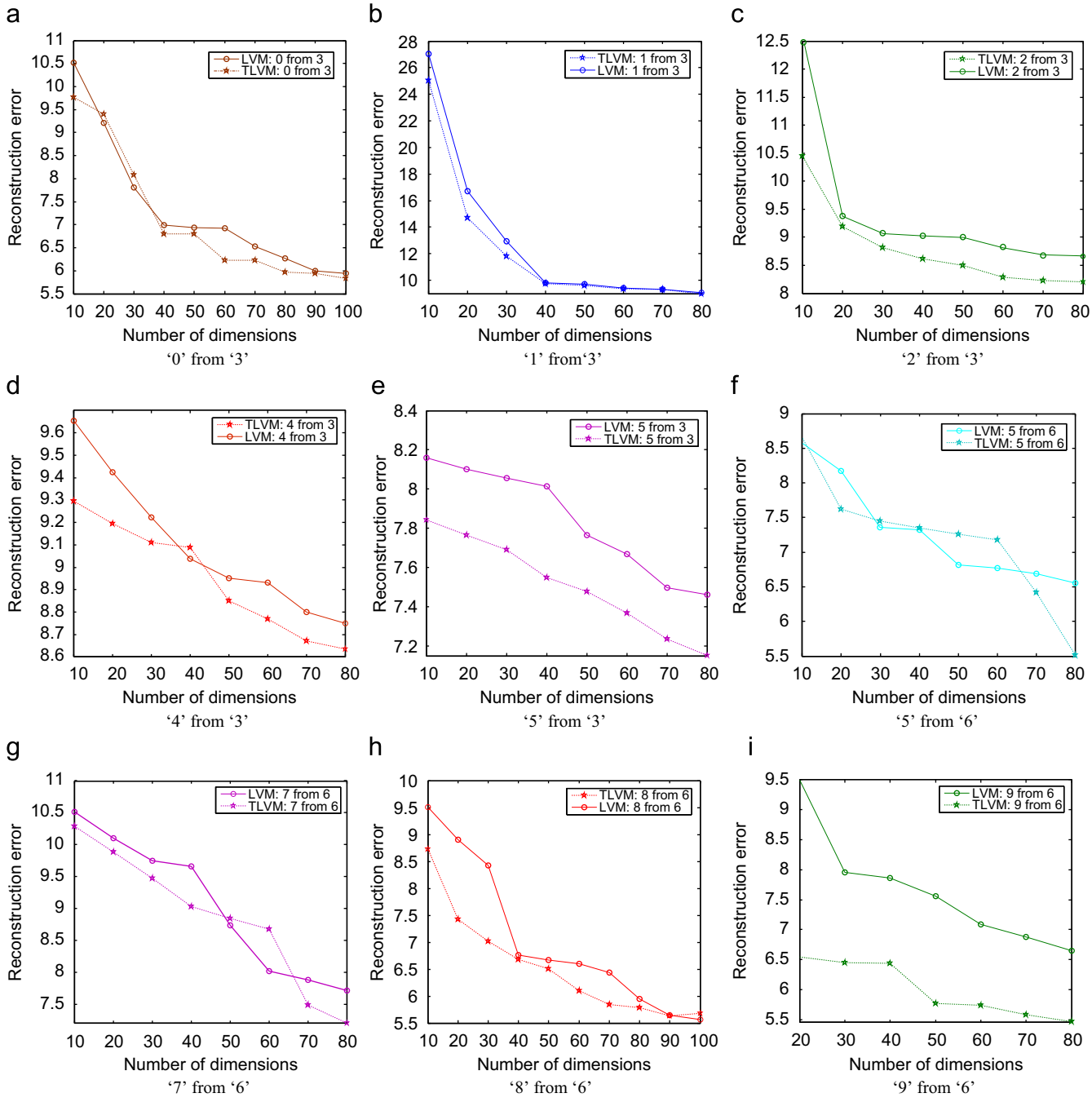

Fig. 8. The comparison of reconstruction error with different dimensions of the latent space on handwritten digits.

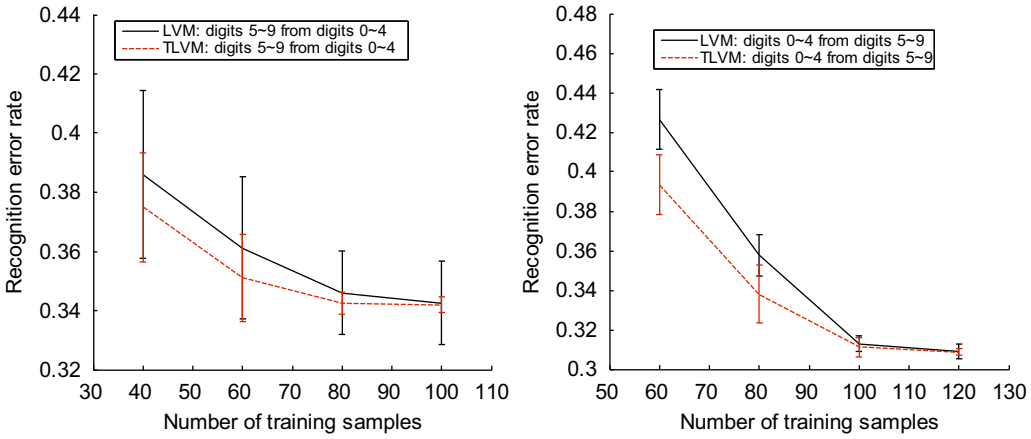

Fig. 9. The comparison of classification error rates with different number of training samples on handwritten digits.

than TLVM. The experimental results verify the validity of transfer learning framework.

4.2. The USPS handwritten digits

The USPS handwritten digits is a benchmark database in the most pattern recognition works. The database includes ten digits' samples, from '0' to '9'. Samples drawn from ten digits are given in Fig. 7.

This section will be divided into two parts. The first part shows the effectiveness of TLVM in reconstruction for USPS database in Fig. 8. The second part presents the performance of TLVM for recognition in Fig. 9. We build the transfer tasks by learning digits from other ones, e.g., '0' from '3' or '1' from '3'. In the following experiments, 300 samples will be drawn randomly from each digit.

Firstly, the digits '3' and '6' are drawn as the training digits respectively, the remained digits will be used for testing. The experimental results in reconstruction are presented in Fig. 8.

As the comparison results shown in Fig. 8, it can be concluded that the proposed method outperform consistently, especially for subfigures (b), (e), (f) and (i). Among the ten digits, several pairs of digits are similar and not easy to distinguish one from another, e.g., '3' and '5', '6' and '9', etc. If we want to obtain the significant reconstruction results, we need to take full advantage of the similarity between the pair of digits. The Bregman divergence just satisfies this requirement. In the TLVM, the hyper-parameter is obtained by considering the Bregman divergence between the training and testing samples. However, in the traditional LVM, only the training samples are considered. At the same time, some results in Fig. 8 could not obviously show the superiority of the TLVM, because there are little similarity can be learned from the pair of digits.

Fig. 9 presents the recognition rates versus number of training samples by using LVM and TLVM. The sample size of one training digit changes from 40 to 120, and each testing digit contains 300 randomly selected samples. In the first experiment, digits 0–4 are utilized for training, and digits 5–9 for testing. In the second experiment, the digits 5–9 are selected for training, the other five digits for testing. The nearest neighbor classifier is adopted here for recognition. Each experiment was repeated 50 times independently. As the results in Fig. 9 demonstrated, the proposed TLVM could almost always achieve the lower recognition error than the traditional LVM. The recognition error rates decreases with the increase of the training sample size. When the number of training samples is small, the TLVM performs significantly, which means that the proposed method works well even a little transferred information are available.

5. Conclusions

When confronting with the case that the training and testing samples are not i.i.d., the performance of traditional latent variable model would be degraded. To address this problem, we propose a transfer learning framework for latent variable model. By modifying the hyper-parameter according to the Bregman divergence between the training and testing samples, the proposed method can deal with the case that the training and testing samples are not i.i.d. Moreover, the transfer learning method can deal with the cross-domain tasks. Experimental results indicate the proposed method outperforms the traditional latent variable model in treating with the cross-domain tasks.

The proposed transfer learning framework can be generalized to other learning algorithms besides dimensionality reduction, such as clustering, regression and classification. As we know, these learning algorithms cannot deal with samples that are not i.i.d. In their transfer learning extensions, this problem may be solved by introducing the Bregman divergence between different distributions, which will be taken into consideration in our future work.

Acknowledgements

We want to thank the helpful comments and suggestions from the anonymous reviewers. This research was supported by the National Natural Science Foundation of China (Nos. 60771068, 60702061, 60832005), the Ph.D. Programs Foundation of Ministry of Education of China (No. 20090203110002), the Natural Science Basic Research Plane in Shaanxi Province of China (No. 2009JM8004), the Open Project Program of the National Laboratory of Pattern Recognition (NLPR) in China and the National Laboratory of Automatic Target Recognition, Shenzhen University, China.

References

[1] H. Hotelling, Analysis of a complex of statistical variables into principal components, Journal of Educational Psychology 24 (1933) 417–441.
[2] Y. Zhang Z.-H. Zhou, Multi-label dimensionality reduction via dependency maximization, in: AAAI Conference on Artificial Intelligence, Chicago, IL, 2008, pp. 1503–1505.
[3] X. He, P. Niyogi, Locality preserving projections, Advances in Neural Information Processing Systems 16 (2003) 153–160.

[4] X. He, D. Cai, S. Yan, H. Zhang, Neighborhood preserving embedding, in: IEEE International Conference on Computer Vision, 2005, pp. 1208–1213.
[5] D. Cai, X. He, J. Han, Semi-supervised discriminant analysis, in: IEEE International Conference on Computer Vision, 2007, pp. 1–7.
[6] X. He, D. Cai, H. Liu, W. Ma, Locality preserving indexing for document representation, SIGIR'04, 2004, pp. 96–103.
[7] Y. Yuan, X. Li, Y. Pang, X. Lu, D. Tao, Binary sparse nonnegative matrix factorization, IEEE Transactions on Circuits and Systems for Video Technology 19 (5) (2009) 772–777.
[8] T. Zhang, D. Tao, X. Li, J. Yang, Patch alignment for dimensionality reduction, IEEE Transactions on Knowledge and Data Engineering 21 (9) (2009) 1299–1313.
[9] X. He, Laplacian regularized d-optimal design for active learning and its application to image retrieval, IEEE Transactions on Image Processing 19 (1) (2010) 254–263.
[10] X. He, S. Yan, Y. Hu, P. Niyogi, H.-J. Zhang, Face recognition using Laplacian faces, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (3) (2005) 328–340.
[11] X. He, M. Ji, H. Bao, Graph embedding with constraints, in: Proceedings of the 2009 International Joint Conference on Artificial Intelligence (IJCAI), Pasadena, CA, July 2009.
[12] I.T. Jolliffe, Principal Component Analysis, Springer-Verlag, New York, 1986.
[13] P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman, Eigenfaces vs. Fisherfaces: recognition using class specific linear projection, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7) (1997) 711–720.
[14] B. Scholkopf, A. Smola, K.-R. Muller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Computation 10 (5) (1998) 1299–1319.
[15] J. Li, X. Li, D. Tao, KPCA for semantic object extraction in images, Pattern Recognition 41 (10) (2008) 3244–3250.
[16] X. Gao, B. Xiao, D. Tao, X. Li, Image categorization: graph edit distance+edge direction histogram, Pattern Recognition 41 (10) (2008) 3179–3191.
[17] S. Roweis, L. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.
[18] X. Li, Y. Pang, Y. Yuan, L1-norm-based 2DPCA, IEEE Transactions on Systems, Man, and Cybernetics, Part B, 2010. Digital Object Identifier, doi:10.1109/TSMCB.2009.2035629.
[19] Y. Pang, D. Tao, Y. Yuan, X. Li, Binary two-dimensional PCA, IEEE Transactions on Systems, Man, and Cybernetics, Part B 38 (4) (2008) 1176–1180.
[20] Y. Yuan, Y. Pang, J. Pan, X. Li, Scene segmentation based on IPCA for visual surveillance, Neurocomputing 72 (10–12) (2009) 2450–2454.
[21] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Computation 15 (6) (2003) 1373–1396.
[22] J.B. Tenenbaum, V. Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323.
[23] V. Silva, J.B. Tenenbaum, Global versus local methods in nonlinear dimensionality reduction, Advances in Neural Information Processing Systems 15 (2003) 705–712.
[24] M.E. Tipping, C.M. Bishop, Probabilistic principal component analysis, Journal of Royal Statistical Society B 61 (3) (1999) 611–622.
[25] N.D. Lawrence, Gaussian process models for visualization of high dimensional data, Advances in Neural Information Processing Systems 16 (2004) 329–336.
[26] N.D. Lawrence, Probabilistic non-linear principal component analysis with Gaussian process latent variable models, Journal of Machine Learning Research 6 (2005) 1783–1816.
[27] J.M. Wang, D.J. Fleet, A. Hertzmann, Gaussian process dynamical models for human motion, IEEE Transactions on Pattern Recognition and Machine Intelligence 30 (2) (2008) 283–298.
[28] R. Urtasun, D.J. Fleet, P. Fua, 3D people tracking with Gaussian process dynamical models, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, New York, USA, June 2006, pp. 238–245.
[29] W.Y. Dai, Q. Yang, G. Xue, Boosting for transfer learning, in: Proceedings of the International Conference on Machine Learning, Oregon, USA, June 2007, pp. 193–200.
[30] J. Sinno, Y. Qiang, A survey on transfer learning, IEEE Transactions on Knowledge and Data Engineering, preprint.
[31] S. Si, D. Tao, B. Geng, Bregman divergence based regularization for transfer subspace learning, IEEE Transactions on Knowledge and Data Engineering, 22 (7) (2010) 929–942.
[32] A. Farhadi, D. Forsyth, R. White, Transfer learning in sign language, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, Minnesota, USA, June 2007, pp. 1–8.
[33] S. Yan, D. Xu, B. Zhang, Graph embedding and extensions: a general framework for dimensionality reduction, neighborhood preserving embedding, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (1) (2007) 40–51.
[34] X. Li, S. Lin, S. Yan, D. Xu, Discriminant locally linear embedding with high-order tensor data, IEEE Transactions on Systems, Man, and Cybernetics, Part B 38 (2) (2008) 342–352.
[35] D.J. Bartholomew, Latent Variable Models and Factor Analysis, Charles Griffin & Co. Ltd., London, 1987.
[36] M.E. Tipping, M. Svensén, C.K.I. Williams, A fast EM algorithm for latent variable density models, Advances in Neural Information Processing Systems 8 (1996) 465–471.
[37] C.E. Rasmussen, C.K.I. Williams, Gaussian Processes for Machine Learning, The MIT press, Cambridge, 2006.
[38] L.M. Bregman, The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming, USSR Computational Mathematics and Physics 7 (1967) 200–217.
[39] X. Li, Y. Pang, Deterministic column-based matrix decomposition, IEEE Transactions on Knowledge and Data Engineering 22 (1) (2010) 145–149.
[40] R. Nock, F. Nielsen, On weight clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (8) (2006) 1223–1235.
[41] L. Cayton, Fast nearest neighbor retrieval for Bregman divergences, in: Proceedings of the International Conference on Machine Learning, Helsinki, Finland, July 2008, pp. 112–119.
[42] A. Banerjee, S. Merugu, I.S. Dhillon, Clustering with Bregman divergences, Journal of Machine Learning Research 6 (2005) 1705–1749.
[43] V.N. Vapnik, in: Statistical Learning Theory, Wiley, New York, 1998.
[44] M.F. Møller, A scaled conjugate gradient algorithm for fast supervised learning, Neural Networks 6 (4) (1993) 525–533.
[45] N.D. Lawrence, M. Seeger, R. Herbrich, Fast sparse Gaussian process methods: the informative vector machine, Advances in Neural Information Processing Systems 15 (2003) 625–632.
[46] 〈http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html〉.
[47] 〈http://cvc.yale.edu/projects/yalefaces/yalefaces.html〉.
[48] 〈http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html〉.

Xinbo Gao received B.Sc., M.Sc. Ph.D. degree in Signal and Information Processing from Xidian University, Xi'an, China, in 1994, 1996 and 1999, respectively. From 1997 to 1998, he was a Research Fellow in the Department of Computer Science at Shizuoka University, Hamamatsu, Japan. From 2000 to 2001, he was a Postdoctoral research Fellow in the Department of Information Engineering at the Chinese University of Hong Kong, Shatin, NT, Hong Kong SAR, China. Since 2001, he join the School of Electronic Engineering at Xidian. Currently, he is a professor of pattern recognition and intelligent system, and Director of the VIPS Lab, Xidian University. His research interests include machine learning and computational Intelligence, pattern recognition, and video content analysis. In these areas, he has published 4 books and around 100 technical articles in refereed journals and proceedings including IEEE TIP, TCSVT, TNN, TSMC etc. He is on the editorial boards of international journals including EURASIP Signal Processing (Elsevier) and Neurocomputing (Elsevier). He served as general chair/co-chair or program committee chair/co-chair or PC member for around 30 major international conferences.

Xiumei Wang received the B.Math Degree from Shandong Normal University in 2002 and the M.Sc. degree from Xidian University in 2005. She is currently a Ph.D. candidate at the School of Electronic Engineering at the Xidian University. Her research interests mainly involve nonparametric statistical models and machine learning.

Xuelong Li is with the Center for OPTical IMagery Analysis and Learning (OPTIMAL), State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, Shaanxi, PR China.

Dacheng Tao received Ph.D. degree from the University of London. Currently, he is a NANYANG Assistant Professor in the Nanyang Technological University, a Visiting Professor in Xi Dian University, a Guest Professor in Wu Han University, and a Research Associate Fellow in the University of London. He works on computational neuroscience, biologically inspired model, statistics and their applications in computational vision and video surveillance. He has authored more than 150 scientific articles at top venues including IEEE, T-PAMI, T-IP, T-KDE, NIPS, KDD, ICDM, and AISTATS with one best paper award. He is an associate editor of *IEEE Transactions on Knowledge and Data Engineering (T-KDE) and Elsevier Neurocomputing*. He has (co-)chaired more than 30 times for special sessions, invited sessions, workshops, panels and conferences. He has served with more than 110 major international conferences including CVPR, ICCV, ECCV, ICDM, KDD and Multimedia, and more than 50 prestigious journals. He is a member of IEEE and IEEE TC on Cognitive Computing.