2015 IEEE 15th International Conference on Data Mining Workshops

Incremental Discriminant Learning for Heterogeneous Domain Adaptation

Peng Han* and Xinxiao Wu†
*The Administrative Center for China's Agenda21, Beijing 100038, P.R. China
Email: hanpeng0715@126.com
†Beijing Laboratory of Intelligent Information Technology
School of Computer Science, Beijing Institute of Technology, Beijing 100081, P.R. China
Email: wuxinxiao@bit.edu.cn

***Abstract*—This paper proposes a new incremental learning method for heterogeneous domain adaptation, in which the training data from both source domain and target domains are acquired sequentially, represented by heterogeneous features. Two different projection matrices are learned to map the data from two domains into a discriminative common subspace, where the intra-class samples are closely-related to each other, the inter-class samples are well-separated from each other, and the data distribution mismatch between the source and target domains is reduced. Different from previous work, our method is capable of incrementally optimizing the projection matrices when the training data becomes available as a data stream instead of being given completely in advance. With the gradually coming training data, the new projection matrices are computed by updating the existing ones using an eigenspace merging algorithm, rather than repeating the learning from the begin by keeping the whole training data set. Therefore, our incremental learning solution for the projection matrices can significantly reduce the computational complexity and memory space, which makes it applicable to a wider set of heterogeneous domain adaptation scenarios with a large training dataset. Furthermore, our method is neither restricted to the corresponding training instances in the source and target domains nor restricted to the same type of feature, which meaningfully relaxes the requirement of training data. Comprehensive experiments on three benchmark datasets clearly demonstrate the effectiveness and efficiency of our method.**

I. Introduction

Heterogeneous domain adaptation has attracted increasing interests because it can learn robust models with very few labeled data from the target domain by effectively leveraging a large amount of labeled data from other existing domains (i.e., source domains) when the data form source and target domains are represented by different features. In [9] , the authors proposed the so-called pivot features from the source and target domains for the cross-language text classification. Based on the co-occurrence information of both domains, the method [19] learns a feature translator between both domains for text-aid image clustering and classification. Harel and Mannor [4] learned rotation matrices to match source data distributions to that of the target domain in activity recognition task. Some other work [3], [5], [12], [11], [15], [18] resorts to discovering a common feature space shared by both source and target domains for more general HDA tasks.

In this paper, we propose a new heterogeneous domain adaptation method, which can incrementally learn a discriminative common feature subspace for linking the source domain to the target domain when the training samples are sequentially acquired. Two projection matrices are learned to respectively transform the source data and target data into a common subspace, by simultaneously maximizing the separability of between-class data and minimizing the variance of within-class data. With the increasing input training data, the new projection matrices can be efficiently computed via updating the existing ones using the eigenspace merging algorithm. Through merging eigenspace models, our method updates the principle components of the total scatter matrix and the between-class scatter matrix separately, and then computes the projection matrices directly from both updated principle component sets. Such incremental solution for projection matrices can greatly reduce the computation expense since it retains the knowledge learned in the past and need not repeat the learning from the beginning whenever the additional training data is presented. The incremental learning can also reduce the memory cost because it only stores the principle components of total and between-class scatter matrices without keeping a large number of training data.

Instead of requiring the corresponding observation of the same instances in source and target domains, our method exploits how to take advantage of label information to learn a common subspace with discrimination. Moreover, due to the limited number of labeled data in the target domain, we also use the unlabeled target-domain samples for domain adaptation by introducing a criterion in the objection function to reduce the data distribution mismatch between the source and target domains.

II. Related Work

A. Heterogeneous domain adaptation

In handling the heterogeneous domain adaptation problem, our work is closely related to the methods [3], [5], [12], [11], [15] which find a "good" common feature space for source and target domains. Taylor and Cristianini [11] learned a common feature space by maximizing the correlation between the source and target training data without any label information. Shi et al. [12] proposed a Heterogeneous Spectral Mapping to discover a common feature subspace by learning two feature mapping matrices as well as the optimal projection of the data from both domains. The label information of training data from both domains is not used. Different from [11] and [12], our method does not require the sample correspondence between

978-1-4673-8493-3/15 $31.00 © 2015 IEEE
DOI 10.1109/ICDMW.2015.186

source and target domains. Moreover, our method utilizes the label information to discover a common feature space with more discrimination. Wang and Mahadevan [15] proposed a manifold alignment based method to learn a common feature space for all heterogeneous domains by simultaneously maximizing the intra-domain similarity and minimizing the inter-domain similarity. Kulis et al. [5] proposed to learn an asymmetric kernel transformation to transfer feature knowledge between source and target domains. Duan et al. [3] first used two different projection matrices to obtain the common feature of source and target domains and then augmented the common feature by incorporating the original feature. Li et al.[6] extend the method in [3] into a convex optimization problem which shares a similar formulation with the well-known Multiple Kernel Learning problem.

Most of these methods have to require the whole training dataset to be given in advance. Whenever the additional training data becomes available, they have to discard the model acquired in the past and repeat the learning from the begin. In contrast, our method can incrementally learn the model by updating the existing ones using the new input training data.

B. Eigenanalysis based incremental learning

From the perspective of incremental learning based on eigenanalysis, several pioneer approaches [7] are related to our method. Hall et al. [7] proposed incremental principle component analysis based on the update of covariance matrix through a residue estimating procedure. Later on they improved their method by merging and splitting eigenspace models that allow a chunk of new samples to be learned in a single step [8]. Pang et al. [13] proposed an incremental linear discriminant analysis (ILDA) in two forms: sequential ILDA and chunk ILDA. The discriminant eigenspace is updated for classification when bursts of data are added to an initial discriminant eigenspace in the form of random chunks. As an improvement of ILDA, Kim applied the concept of the sufficient spanning set approximation in updating the between-class scatter matrix, the projected data matrix, and the total scatter matrix. They also applied the same concept of the spanning set to the image set-based recognition [14].

Unlike these popular incremental learning methods, the proposed method in this paper can handle the situation in which the training and the testing data come from different domains represented by heterogeneous features with different dimensions.

III. Incremental Heterogeneous Domain Adaptation

In this work, we assume there are only one source domain and one target domain. For some given class, we are provided with a large number of labeled training samples $\{(\mathrm{x}_i^s, y_i^s)|_{i=1}^{n_s}\}$ from the source domain as well as a limited number of labeled samples $\{(\mathrm{x}_i^l, y_i^l)|_{i=1}^{n_l}\}$ and some unlabeled samples $\{(\mathrm{x}_i^u, y_i^u)|_{i=1}^{n_u}\}$ from the target domain, where y_i^s, y_i^l and y_i^u are the labels of the samples x_i^s, x_i^l and x_i^u, respectively, and $y_i^s, y_i^l, y_i^u \in \{1, ..., c\}$ with c the number of classes. The dimension of source data x_i^s is r_s and that of target data x_i^l and x_i^u is r_t. In the heterogeneous domain adaptation, we have $r_s \neq r_t$.

A. Heterogeneous discriminant analysis

Our goal is to learn two different projection matrices w_s and w_t for respectively transforming the source data and target data into a common subspace, by simultaneously maximizing the variance of samples from different classes, minimizing the variance of samples from the same class, and reducing the data distribution mismatch between source and target domains. We define the common subspace as $\mathbb{R}^{r_c}$, and formulate the objective function to find the optimal w_s and w_t as follows:

$$\max_{\mathrm{w}} \frac{|\mathrm{w}^T \mathrm{S_B w}|}{|\mathrm{w}^T \mathrm{S_T w} + \alpha \mathrm{w}^T \mathrm{S_D w}|}, \tag{1}$$

where $\mathrm{w} = \begin{bmatrix} \mathrm{w}_s \\ \mathrm{w}_t \end{bmatrix}$ is the combination of w_s and w_t, $\mathrm{w}^T \mathrm{S_B w}$ is the scatter matrix of between-class data projected in the common subspace, $\mathrm{w}^T \mathrm{S_w w}$ the scatter matrix of within-class data projected in the common subspace, $\mathrm{w}^T \mathrm{S_T w} = \mathrm{w}^T(\mathrm{S_B} + \mathrm{S_w})\mathrm{w}$ the scatter matrix of all data in the common subspace, $\mathrm{w}^T \mathrm{S_D w}$ the distribution difference between source and target domains, $\alpha > 0$ the tradeoff parameter. The two criterions in (1) are equivalent and this paper adopts the second criterion.

Specifically, $\mathrm{w}^T \mathrm{S_T w}$ is defined by

$$\sum_{i=1}^{n_s} (\mathrm{w}_s^T \mathrm{x}_i^s - \mu)(\mathrm{w}_s^T \mathrm{x}_i^s - \mu)^T + \sum_{i=1}^{n_l} (\mathrm{w}_t^T \mathrm{x}_i^l - \mu)(\mathrm{w}_t^T \mathrm{x}_i^l - \mu)^T,$$

where μ is the global mean of all the projected labeled training data, n_s and n_l are the numbers of labeled samples from the source and target domains, respectively. By $\mu = \frac{n_s \mathrm{w}_s^T \mu_s + n_l \mathrm{w}_t^T \mu_l}{n_s + n_l}$ where μ_s and μ_l represent the mean of the projected labeled source and target data, respectively, $\mathrm{w}^T \mathrm{S_T w}$ can be rewritten as

$$\mathrm{w}^T \begin{bmatrix} \mathrm{S}_{\mathrm{T},s} & 0 \\ 0 & \mathrm{S}_{\mathrm{T},t} \end{bmatrix} \mathrm{w} + \frac{2 n_s n_l}{n_s + n_l} \mathrm{w}^T \begin{bmatrix} \mu_s \\ -\mu_l \end{bmatrix} \begin{bmatrix} \mu_s \\ -\mu_l \end{bmatrix}^T \mathrm{w},$$

where $\mathrm{S}_{\mathrm{T},s} = \sum_{i=1}^{n_s} (\mathrm{x}_i^s - \mu_s)(\mathrm{x}_i^s - \mu_s)^T$ and $\mathrm{S}_{\mathrm{T},t} = \sum_{i=1}^{n_l} (\mathrm{x}_i^l - \mu_l)(\mathrm{x}_i^l - \mu_l)^T$. Therefore, $\mathrm{S_T}$ becomes $\begin{bmatrix} \mathrm{S}_{\mathrm{T},s} & 0 \\ 0 & \mathrm{S}_{\mathrm{T},t} \end{bmatrix} + \frac{2 n_s n_l}{n_s + n_l} \begin{bmatrix} \mu_s \\ -\mu_l \end{bmatrix} \begin{bmatrix} \mu_s \\ -\mu_l \end{bmatrix}^T$.

The between-class scatter matrix $\mathrm{w}^T \mathrm{S_B w}$ is defined by

$$\sum_{j=1}^{c} n_j \left(\frac{\mathrm{w}_s^T \mathrm{m}_{sj} + \mathrm{w}_t^T \mathrm{m}_{tj}}{n_j} - \mu\right)\left(\frac{\mathrm{w}_s^T \mathrm{m}_{sj} + \mathrm{w}_t^T \mathrm{m}_{tj}}{n_j} - \mu\right)^T,$$

where m_{sj} indicates the sum of the j-th class samples from the source domain, m_{tj} the sum of the j-th class samples from the target domain, and n_j the total number of the j-th class samples from both source and target domains. With $\mu = \frac{n_s \mathrm{w}_s^T \mu_s + n_l \mathrm{w}_t^T \mu_l}{n_s + n_l}$, $\mathrm{w}^T \mathrm{S_B w}$ can be formulated as

$$\mathrm{w}^T \Big(\sum_{j=1}^{c} \frac{1}{n_j} \begin{bmatrix} \mathrm{m}_{sj} \\ \mathrm{m}_{lj} \end{bmatrix} \begin{bmatrix} \mathrm{m}_{sj} \\ \mathrm{m}_{lj} \end{bmatrix}^T\Big) \mathrm{w} - \frac{\mathrm{w}^T}{n_s + n_l} \begin{bmatrix} n_s \mu_s \\ n_l \mu_l \end{bmatrix} \begin{bmatrix} n_s \mu_s \\ n_l \mu_l \end{bmatrix}^T \mathrm{w},$$

and S_{B} is $\sum_{j=1}^{c} n_j \Big(\begin{bmatrix} \frac{\mathrm{m}_{sj}}{n_j} \\ \frac{\mathrm{m}_{lj}}{n_j} \end{bmatrix} - \begin{bmatrix} \frac{n_s \mu_s}{n_s + n_l} \\ \frac{n_l \mu_l}{n_s + n_l} \end{bmatrix} \Big) \Big(\begin{bmatrix} \frac{\mathrm{m}_{sj}}{n_j} \\ \frac{\mathrm{m}_{lj}}{n_j} \end{bmatrix} - \begin{bmatrix} \frac{n_s \mu_s}{n_s + n_l} \\ \frac{n_l \mu_l}{n_s + n_l} \end{bmatrix} \Big)^T$.

The between-domain distribution difference matrix $\mathrm{w}^T\mathrm{S_D}\mathrm{w}$ is defined by

$$\mathrm{w}^T\begin{bmatrix}\mu_s\\-\mu_t\end{bmatrix}\begin{bmatrix}\mu_s\\-\mu_t\end{bmatrix}^T\mathrm{w},$$

where $\mu_t = \frac{1}{n_l+n_u}(\sum_{i=1}^{n_l}\mathrm{x}_i^l+\sum_{i=1}^{n_u}\mathrm{x}_i^u)$ is the mean of all the projected target data, and $\mathrm{S_D} = \begin{bmatrix}\mu_s\\-\mu_t\end{bmatrix}\begin{bmatrix}\mu_s\\-\mu_t\end{bmatrix}^T$. With the eigen-decomposition

$$\mathrm{S_B}w = \lambda(\mathrm{S_T}+\alpha\mathrm{S_D})w = \lambda\tilde{\mathrm{S}}_\mathrm{T}w, \tag{2}$$

the optimal w_s and w_t are constructed by the first-r_s rows and the last-r_t rows of the top-r_c eigenvectors $[w_1, w_2, ..., w_{r_c}]$.

B. Incremental learning

The proposed incremental learning algorithm mainly has three steps: (1) update the matrix $\tilde{\mathrm{S}}_\mathrm{T}$; (2) update the matrix $\mathrm{S_B}$; (3) compute the discriminant components w from the updated $\tilde{\mathrm{S}}_\mathrm{T}$ and $\mathrm{S_B}$.

Updating the matrix $\tilde{\mathrm{S}}_\mathrm{T}$. $\tilde{\mathrm{S}}_\mathrm{T}$ is formed by $\begin{bmatrix}\mathrm{S}_{\mathrm{T},s} & 0\\0 & \mathrm{S}_{\mathrm{T},t}\end{bmatrix}+\frac{2n_sn_l}{n_s+n_l}\begin{bmatrix}\mu_s\\-\mu_l\end{bmatrix}\begin{bmatrix}\mu_s\\-\mu_l\end{bmatrix}^T+\alpha\begin{bmatrix}\mu_s\\-\mu_t\end{bmatrix}\begin{bmatrix}\mu_s\\-\mu_t\end{bmatrix}^T$, where $\mathrm{S}_{\mathrm{T},s}$ can be represented by a set of orthogonal vectors that span the subspace occupied by the source data, and $\mathrm{S}_{\mathrm{T},t}$ be represented by a set of orthogonal vectors that span the subspace of the target data. So we use the modified eigenspace merging algorithm of Hall in order to incrementally compute the principal components of $\mathrm{S}_{\mathrm{T},s}$ and $\mathrm{S}_{\mathrm{T},t}$. Consequently, updating $\tilde{\mathrm{S}}_\mathrm{T}$ is actually updating the total scatter matrix of source data $\mathrm{S}_{\mathrm{T},s}$ and the total scatter matrix of labeled target data $\mathrm{S}_{\mathrm{T},t}$.

Given an existing data set represented by eigenspace models $\{\mathrm{P}_{s1}, \Lambda_{s1}, \mathrm{P}_{t1}, \Lambda_{t1}, \mu_{s1}, n_{s1}, \mu_{l1}, n_{l1}, \mu_{t1}, n_{t1}\}$ and a new set of data represented by eigenspace models $\{\mathrm{P}_{s2}, \Lambda_{s2}, \mathrm{P}_{t2}, \Lambda_{t2}, \mu_{s2}, n_{s2}, \mu_{l2}, n_{l2}, \mu_{t2}, n_{t2}, \}$, where μ_{sk} is the source mean, n_{sk} is the number of source samples, μ_{lk} the mean of the labeled target samples, n_{lk} the number of labeled target samples, μ_{tk} the mean of all the target samples, n_{tk} the total number of target samples, P_{sk} the source eigenvector matrix and Λ_{sk} the source eigenvalue matrix that satisfy $\mathrm{S}_{\mathrm{T},sk} \simeq \mathrm{P}_{sk}\Lambda_{sk}\mathrm{P}_{sk}^T$, P_{tk} the target eigenvector matrix and Λ_{tk} the target eigenvalue matrix that satisfy $\mathrm{S}_{\mathrm{T},tk} \simeq \mathrm{P}_{tk}\Lambda_{tk}\mathrm{P}_{tk}^T$, $k \in \{1, 2\}$, the update is formulated by

$$\begin{aligned}&\xi_1(\{\mathrm{P}_{sk}, \Lambda_{sk}, \mathrm{P}_{tk}, \Lambda_{tk}, \mu_{sk}, n_{sk}, \mu_{lk}, n_{lk}, \mu_{tk}, n_{tk}\})\\&= \{\mathrm{P}_{s3}, \Lambda_{s3}, \mathrm{P}_{t3}, \Lambda_{t3}, \mu_{s3}, n_{s3}, \mu_{l3}, n_{l3}, \mu_{t3}, n_{t3}\},\end{aligned}$$

where P_{s3} and Λ_{s3} are the updated eigenvectors and eigenvalues that satisfy with $\mathrm{S}_{\mathrm{T},s3} \simeq \mathrm{P}_{s3}\Lambda_{s3}\mathrm{P}_{s3}^T$, P_{t3} and Λ_{t3} are the updated eigenvectors and eigenvalues that satisfy with $\mathrm{S}_{\mathrm{T},t3} \simeq \mathrm{P}_{t3}\Lambda_{t3}\mathrm{P}_{t3}^T$, $n_{s3} = n_{s1} + n_{s2}$, $n_{l3} = n_{l1} + n_{l2}$, $n_{t3} = n_{t1} + n_{t2}$, $\mu_{s3} = (\mu_{s1}n_{s1} + \mu_{s2}n_{s2})/n_{s3}$ the updated mean of source data, $\mu_{l3} = (\mu_{l1}n_{l1} + \mu_{l2}n_{l2})/n_{l3}$ the updated mean of labeled target data, and $\mu_{t3} = (\mu_{t1}n_{t1} + \mu_{t2}n_{t2})/n_{t3}$ the updated mean of all the target data.

To reduce the dimension of eigenvalue problem, the concept of the sufficient spanning set is used. Since both the updated total scatter matrix $\mathrm{S}_{\mathrm{T},s3}$ of source data and the updated total scatter matrix $\mathrm{S}_{\mathrm{T},t3}$ of labeled target data can be represented as the sum of the scatter matrices of the two sets explicitly as

$$\begin{aligned}\mathrm{S}_{\mathrm{T},s3} &= \mathrm{S}_{\mathrm{T},s1}+\mathrm{S}_{\mathrm{T},s2}+\frac{n_{s1}n_{s2}}{n_{s3}}(\mu_{s1}-\mu_{s2})(\mu_{s1}-\mu_{s2})^T\\\mathrm{S}_{\mathrm{T},t3} &= \mathrm{S}_{\mathrm{T},t1}+\mathrm{S}_{\mathrm{T},t2}+\frac{n_{l1}n_{l2}}{n_{l3}}(\mu_{l1}-\mu_{l2})(\mu_{l1}-\mu_{l2})^T.\end{aligned} \tag{3}$$

Then the sufficient spanning sets of P_{s3} and P_{t3} can be represented as

$$\Phi_s = h([\mathrm{P}_{s1}, \mathrm{P}_{s2}, \mu_{s1} - \mu_{s2}]), \Phi_t = h([\mathrm{P}_{t1}, \mathrm{P}_{t2}, \mu_{l1} - \mu_{l2}]), \tag{4}$$

where h is an orthonormalization function (e.g. QR decomposition). With the rotation matrices R_s and R_t, P_{s3} and P_{t3} can be computed by $\mathrm{P}_{s3} = \Phi_s\mathrm{R}_s$ and $\mathrm{P}_{t3} = \Phi_t\mathrm{R}_t$. Thus, we solve two smaller eigenproblems to obtain R_s and Λ_{s3} as well as R_t and Λ_{t3}:

$$\begin{aligned}\mathrm{S}_{\mathrm{T},s3} = \mathrm{P}_{s3}\Lambda_{s3}\mathrm{P}_{s3}^T \implies \Phi_s^T\mathrm{S}_{\mathrm{T},s3}\Phi_s = \mathrm{R}_s\Lambda_{s3}\mathrm{R}_s^T,\\\mathrm{S}_{\mathrm{T},t3} = \mathrm{P}_{t3}\Lambda_{t3}\mathrm{P}_{t3}^T \implies \Phi_t^T\mathrm{S}_{\mathrm{T},t3}\Phi_s = \mathrm{R}_t\Lambda_{t3}\mathrm{R}_t^T.\end{aligned}$$

With the updated P_{s3} and P_{t3}, the computation of the eigenvector matrix P_3 and eigenvalue matrix Λ_3 that satisfy $\tilde{\mathrm{S}}_{\mathrm{T}3} = \mathrm{P}_3\Lambda_3\mathrm{P}_3^T$ is formulated as follows:

$$\mathrm{P}_3 = \Phi\mathrm{R} = h(\left[\begin{bmatrix}\mathrm{P}_{s3} & 0\\0 & \mathrm{P}_{t3}\end{bmatrix}, \begin{bmatrix}\mu_{s3}\\-\mu_{l3}\end{bmatrix}, \begin{bmatrix}\mu_{s3}\\-\mu_{t3}\end{bmatrix}\right])\mathrm{R}, \tag{5}$$

where h is an orthonormalization function and R is a rotation matrix. Therefore, the final P_3 and Λ_3 can be computed by a smaller eigen-analysis, given by

$$\tilde{\mathrm{S}}_{\mathrm{T}3} = \mathrm{P}_3\Lambda_3\mathrm{P}_3^T \implies \Phi^T\tilde{\mathrm{S}}_{\mathrm{T}3}\Phi = \mathrm{R}\Lambda_3\mathrm{R}^T.$$

Suppose that $d_{\mathrm{T},s1}$ and $d_{\mathrm{T},s2}$ are the numbers of eigenvectors of P_{s1} and P_{s2}, respectively, the matrix $\Phi_s^T\mathrm{S}_{\mathrm{T},s3}\Phi_s$ has the reduced size $d_{\mathrm{T},s1} + d_{\mathrm{T},s2} + 1$, and the eigen-analysis of $\mathrm{S}_{\mathrm{T},s3}$ requires $O((d_{\mathrm{T},s1} + d_{\mathrm{T},s2} + 1)^3)$. Similarly, the complexity of eigen-analyzing $\mathrm{S}_{\mathrm{T},t3}$ is $O((d_{\mathrm{T},t1}+d_{\mathrm{T},t2}+1)^3)$ where $d_{\mathrm{T},t1}$ and $d_{\mathrm{T},t2}$ represent the number of eigenvectors of P_{t1} and P_{t2}, respectively. Let $d_{\mathrm{T},s3}$ and $d_{\mathrm{T},t3}$ be the number of eigenvectors of P_{s3} and P_{t3}, respectively, then the time cost of eigen-analyzing $\tilde{\mathrm{S}}_{\mathrm{T}3}$ is $O((d_{\mathrm{T},s3} + d_{\mathrm{T},t3} + 2)^3)$. So the total time complex of eigen-analysis in our method is $O((d_{\mathrm{T},s1}+d_{\mathrm{T},s2}+1)^3+(d_{\mathrm{T},t1}+d_{\mathrm{T},t2}+1)^3+(d_{\mathrm{T},s3}+d_{\mathrm{T},t3}+2)^3)$. While in batch mode, the costs of eigen-decomposing $\mathrm{S}_{\mathrm{T},s3}$ and $\mathrm{S}_{\mathrm{T},t3}$ are $O((\min(r_s, n_s))^3)$ and $O((\min(r_t, n_l))^3)$. By adding the cost of eigen-decomposing $\tilde{\mathrm{S}}_{\mathrm{T}3}$, the total time complex of batch mode is $O((\min(r_s, n_s))^3+(\min(r_t, n_l))^3+(d_{\mathrm{T},s3} + d_{\mathrm{T},t3} + 2)^3)$. As $d_{\mathrm{T},s1} + d_{\mathrm{T},s2} \ll n_s < r_s$ and $d_{\mathrm{T},t1} + d_{\mathrm{T},t2} \ll n_l < r_t$ in practice, our incremental learning method is more efficient than the batch mode.

Updating the matrix $\mathrm{S_B}$**.** The between-class scatter matrix of the existing data set is denoted by $\{\mathrm{Q}_1, \Delta_1, \mathrm{m}_{sj,1}, n_{sj,1}, \mathrm{m}_{lj,1}, n_{lj,1}, \mu_{s1}, n_{s1}, \mu_{l1}, n_{l1}\}$ with $j = \{1, ..., c\}$, where $\mathrm{m}_{sj,1}$ is the sum of source-domain samples from class j, $n_{sj,1}$ is the number of source-domain samples from class j, $\mathrm{m}_{lj,1}$ the sum of labeled target-domain samples from class j, $n_{lj,1}$ the number of labeled target-domain samples from class j, Q_1 and Δ_1 are the eigenvectors and eigenvalues of $\mathrm{S}_{\mathrm{B}1}$, i.e., $\mathrm{S}_{\mathrm{B}1} \simeq \mathrm{Q}_1\Delta_1\mathrm{Q}_1^T$. Given a new data set represented by $\{\mathrm{Q}_2, \Delta_2, \mathrm{m}_{sj,2}, n_{sj,2}, \mathrm{m}_{lj,2}, n_{lj,2}, \mu_{s2}, n_{s2}, \mu_{l2}, n_{l2}\}$

with $j = \{1, ..., c\}$, $k \in \{1, 2\}$, the update is described as

$$\xi_2(\{\mathrm{Q}_k, \Delta_k, \mathrm{m}_{sj,k}, n_{sj,k}, \mathrm{m}_{lj,k}, n_{lj,k}, \mu_{sk}, n_{sk}, \mu_{lk}, n_{lk}\}) = \{\mathrm{Q}_3, \Delta_3, \mathrm{m}_{sj,3}, n_{sj,3}, \mathrm{m}_{lj,3}, n_{lj,3}, \mu_{s3}, n_{s3}, \mu_{l3}, n_{l3}\},$$

where $\mathrm{S}_{\mathrm{B3}} \simeq \mathrm{Q}_3\Delta_3\mathrm{Q}_3^T$, $\mathrm{m}_{sj,3} = (\mathrm{m}_{sj,1} + \mathrm{m}_{sj,2})/(n_{sj,1} + n_{sj,2})$, $\mathrm{m}_{lj,3} = (\mathrm{m}_{lj,1} + \mathrm{m}_{lj,2})/(n_{lj,1} + n_{lj,2})$, $n_{sj,3} = n_{sj,1} + n_{sj,2}$ and $n_{lj,3} = n_{lj,1} + n_{lj,2}$. To obtain the sufficient spanning set for efficient eigen-computation, the updated between-class scatter matrix S_{B3} is represented by the sum of the between-class scatter matrices of the first two data sets S_{B1} and S_{B2}:

$$\mathrm{S}_{\mathrm{B1}} + \mathrm{S}_{\mathrm{B2}} + \mathrm{A} + \frac{n_1 n_2}{n_1 + n_2} \begin{bmatrix} \frac{n_{s1}\mu_{s1}}{n_1} - \frac{n_{s2}\mu_{s2}}{n_2} \\ \frac{n_{l1}\mu_{l1}}{n_1} - \frac{n_{l2}\mu_{l2}}{n_2} \end{bmatrix} \begin{bmatrix} \frac{n_{s1}\mu_{s1}}{n_1} - \frac{n_{s2}\mu_{s2}}{n_2} \\ \frac{n_{l1}\mu_{l1}}{n_1} - \frac{n_{l2}\mu_{l2}}{n_2} \end{bmatrix}^T, \tag{6}$$

where $\mathrm{A} = \sum_{j=1}^{c} \frac{-n_{j1}n_{j2}}{n_{j1}+n_{j2}} \begin{bmatrix} \frac{\mathrm{m}_{sj,1}}{n_{j1}} - \frac{\mathrm{m}_{sj,2}}{n_{j2}} \\ \frac{\mathrm{m}_{lj,1}}{n_{j1}} - \frac{\mathrm{m}_{lj,2}}{n_{j2}} \end{bmatrix} \begin{bmatrix} \frac{\mathrm{m}_{sj,1}}{n_{j1}} - \frac{\mathrm{m}_{sj,2}}{n_{j2}} \\ \frac{\mathrm{m}_{lj,1}}{n_{j1}} - \frac{\mathrm{m}_{lj,2}}{n_{j2}} \end{bmatrix}^T$, $n_k = n_{sk} + n_{lk}$ and $n_{jk} = n_{sj,k} + n_{lj,k}$ with $k \in \{1, 2\}$. Since both S_{B1} and S_{B2} are represented by the first few eigenvectors such that $\mathrm{S}_{\mathrm{B1}} \simeq \mathrm{Q}_1\Delta_1\mathrm{Q}_1^T$ and $\mathrm{S}_{\mathrm{B2}} \simeq \mathrm{Q}_2\Delta_2\mathrm{Q}_1^T$, the sufficient spanning set for S_{B3} can be similarly set as

$$\Psi = h(\begin{bmatrix} \mathrm{Q}_1, \mathrm{Q}_2, \begin{bmatrix} n_{s1}\mu_{s1}/n_1 - n_{s2}\mu_{s2}/n_2 \\ n_{t1}\mu_{l1}/n_1 - n_{t2}\mu_{l2}/n_2 \end{bmatrix} \end{bmatrix}). \tag{7}$$

The negative semi-definite matrix A does not add any more dimensions to Ψ and it can be seen as the scatter matrix of the components to be removed from the updated data. With a rotation matrix R_{B}, Q_3 can be computed by $\mathrm{Q}_3 = \Psi\mathrm{R}_{\mathrm{B}}$. Accordingly, the new small dimensional eigen-problem is given by

$$\mathrm{S}_{\mathrm{B3}} = \mathrm{Q}_3\Delta_3\mathrm{Q}_3^T \implies \Psi^T\mathrm{S}_{\mathrm{B3}}\Psi = \mathrm{R}_{\mathrm{B}}\Delta_3\mathrm{R}_{\mathrm{B}}^T.$$

Let d_{B1} and d_{B2} be the numbers of eigenvectors of Q_1 and Q_2, respectively. Whereas the eigenanalysis of the between-class scatter in batch mode requires $O(\min(r_s + r_t, c)^3)$, the proposed incremental scheme requires only $O((d_{\mathrm{B1}} + d_{\mathrm{B2}} + 1)^3)$.

Updating the projection matrices w. With the updated total data $\{\mathrm{P}_3, \Lambda_3\}$ and the updated between-class data $\{\mathrm{Q}_3, \Delta_3\}$, the projection matrices can be found by

$$\xi_3(\{\mathrm{P}_3, \Lambda_3, \mathrm{Q}_3, \Delta_3\}) = \mathrm{w}.$$

In order to further reduce the computation complexity, we introduce new sufficient spanning set to change the eigen-analysis into a smaller dimensional eigen-problem. Let $\mathrm{G} = \mathrm{P}_3\Lambda_3^{-1/2}$, then $\mathrm{G}\tilde{\mathrm{S}}_{\mathrm{T3}}\mathrm{G}^T = \mathrm{I}$. As the denominator of the second criterion in (1) is the identity matrix, the problem becomes to find the components that maximize $\mathrm{G}^T\mathrm{S}_{\mathrm{B3}}\mathrm{G}$, s.t. $\mathrm{G}^T\mathrm{S}_{\mathrm{B3}}\mathrm{G} = \mathrm{H}\Sigma\mathrm{H}^T$, and then the final projection matrices are obtained by $\mathrm{w} = \mathrm{GH}$. This eigen-problem can be solved using the sufficient spanning set defined by $\Omega = h(\mathrm{G}^T\mathrm{Q}_3)$. By rotating the sufficient spanning set, the eigen-analysis problem becomes

$$\mathrm{G}^T\mathrm{S}_{\mathrm{B3}}\mathrm{G} = \Omega\mathrm{R}_p\Sigma\mathrm{R}_p^T\Omega^T \Longrightarrow \Omega^T\mathrm{G}^T\mathrm{S}_{\mathrm{B3}}\mathrm{G}\Omega = \mathrm{R}_p\Sigma\mathrm{R}_p^T. \tag{8}$$

The updated projection matrices is given by $\mathrm{w} = \begin{bmatrix} \mathrm{w}_s \\ \mathrm{w}_t \end{bmatrix} = \mathrm{GH} = \mathrm{G}\Omega\mathrm{R}_p$.

Let d_{B3} be number of eigenvectors Q_3, the computational time of eigen-problem in (8) is $O(d_{\mathrm{B3}}^3)$. In practice, the dimension d_{T3} of P_3 is usually larger than d_{B3}, therefore the computation efficiency of w improves from $O(d_{\mathrm{B3}}^3)$ to $O(d_{\mathrm{T3}}^3)$. In terms of the space complexity, the proposed incremental learning costs $O((r_s + r_t)d_{\mathrm{B3}} + r_s d_{\mathrm{T},s3} + r_t d_{\mathrm{T},t3} + (r_s + r_t)c_3)$ while the batch mode costs $O(r_s n_{s3} + r_t n_{t3} + (r_s + r_t)c)$ with $d_{\mathrm{T},s3} \ll n_{s3}$, $d_{\mathrm{T},t3} \ll n_{t3}$ and $d_{\mathrm{B3}} \leq d_{\mathrm{T3}} \leq d_{\mathrm{T},s3} + d_{\mathrm{T},t3} + 2$, which clearly demonstrates that our method can effectively reduce the space complexity. We summarize the proposed incremental heterogeneous domain adaptation in Alogrithm1.

IV. Experiments

In this section, the proposed incremental heterogeneous domain adaptation method is evaluated for cross-view action classification, objection recognition and multilingual text categorization. All the experiments are performed on a PC with Intel Core 2.83 GHz CPU and 8 GB of RAM using the non-optimized Matlab codes.

A. Datasets

The IXMAS multi-view dataset [17] is employed for cross-view action classification. It consists of 12 complete action classes and each is executed three times by 12 subjects. Each action is recorded by 5 cameras observing the subjects from very different perspectives. We take one view as the source domain and another different view as the target domain, and look into the classification performances of all possible pairwise combinations. The 4000-dimensional HOG/HOF and 2000-dimensional MBH features of dense trajectory [16] are adopted in source and target domains, respectively.

The dataset for object recognition [10], [5] contains a total of 4106 images with 31 categories from three sources: amazon (web images downloaded from an online merchant), dslr (high-resolution images taken from a digital DLR camera) and webcam (low-resolution images taken from a web camera). SURF features [2] are extracted for all the images. The images from amazon and webcam are clustered into 800 visual words by using k-means. After vector quantization, each image is represented as a 800 dimensional histogram feature. Similarly, we represented each image from dslr as a 600-dimensional histogram feature. In the experiments, dslr is used as the target domain, while amazon and webcam are considered as two individual source domains.

We use the Reuters multilingual dataset [1] for text categorization, which is collected by sampling parts of the Teuters RCV1 and RCV2 collections. It contains about 11K articles from 6 classes in 5 languages (i.e., English, French, German, Italian and Spanish). While each document was also translated into the other four languages in this dataset, we do not use the translated documents in this work. All documents are represented as a bag of words and the TF-IDF are extracted. We perform PCA with 60% energy preserved on the TF-IDF features and the feature dimensions of 5 languages (i.e., English, French, German, Italian and Spanish) are 1131, 1230, 1417, 1041 and 807, respectively. We take one language as the source domain and another different language as the target domain, and look into the categorization performances of all possible pairwise combinations.

Algorithm 1 Incremental Discriminant Heterogeneous Domain Adaptation

Input:An existing data set represented by its total eigenmodel $\{\mathrm{P}_{s1}, \Lambda_{s1}, \mathrm{P}_{t1}, \Lambda_{t1}, ...\}$ and its between-class eigenmodel $\{\mathrm{Q}_1, \Delta_1\}$, and a new data set

Output: Two updated projection matrices w_s and w_t.

1. Compute $\{\mathrm{P}_{s2}, \Lambda_{s2}, \mathrm{P}_{t2}, \Lambda_{t2}, ...\}$ and $\{\mathrm{Q}_2, \Delta_2, ...\}$ of the new data set in batch mode.
2. Update the total scatter matrix for $\{\mathrm{P}_{s3}, \Lambda_{s3}, \mathrm{P}_{t3}, \Lambda_{t3}, ...\}$:
   Compute $\mathrm{S}_{\mathrm{T},s3}$ and $\mathrm{S}_{\mathrm{T},t3}$ by (3), $\mathrm{S}_{\mathrm{T},sk} \simeq \mathrm{P}_{sk}\Lambda_{sk}\mathrm{P}_{sk}^T$ and $\mathrm{S}_{\mathrm{T},tk} \simeq \mathrm{P}_{tk}\Lambda_{tk}\mathrm{P}_{tk}^T$, $k \in \{1, 2\}$.
   Set the spanning sets Φ_s and Φ_t by (4).
   Compute eigenvectors R_s of $\Phi_s^T\mathrm{S}_{T,s3}\Phi_s$. $\mathrm{P}_{s3} = \Phi_s\mathrm{R}_s$.
   Compute eigenvectors R_t of $\Phi_t^T\mathrm{S}_{T,t3}\Phi_t$. $\mathrm{P}_{t3} = \Phi_t\mathrm{R}_t$.
   Set the spanning set Φ by (5).
   Compute eigenvectors R of $\Phi^T\tilde{\mathrm{S}}_{T3}\Phi$. $\mathrm{P}_3 = \Phi\mathrm{R}$.
3. Update the between-class scatter matrix for $\{\mathrm{Q}_3, \Theta_3, ...\}$:
   Compute $\mathrm{S}_{\mathrm{B}3}$ from (6) and $\mathrm{S}_{\mathrm{B}k} \simeq \mathrm{Q}_k\Delta_k\mathrm{Q}_k^T$, $k \in \{1, 2\}$.
   Set the spanning set Ψ by (7).
   Compute eigenvectors R_{B} of $\Psi^T S_{\mathrm{B}3}\Psi$. $\mathrm{Q}_3 = \Psi R_{\mathrm{B}}$.
4. Update the projection matrices:
   Compute $\mathrm{G} = \mathrm{P}_3\Lambda_3^{-1/2}$ and $\Omega = h([\mathrm{G}^T\mathrm{Q}_3])$.
   Eigendecompose $\Omega^T\mathrm{G}^T\mathrm{S}_{\mathrm{B}3}\mathrm{G}\Omega$ for the eigenvectors R_p. $\mathrm{w} = \begin{bmatrix}\mathrm{w}_s \\ \mathrm{w}_t\end{bmatrix} = \mathrm{G}\Omega\mathrm{R}_p$.

TABLE I. SUMMARIZATION OF ALL THE DATASETS. THE COLUMNS OF "# LABELED", "# UNLABELED" AND "# TEST" RESPECTIVELY INDICATE THE NUMBER OF LABELED TRAINING SAMPLES, UNLABELED TRAINING SAMPLES AND TEST SAMPLES FOR EACH CLASS. THE TWO NUMBERS IN A TUPLE IN THE COLUMNS OF "# LABELED" AND "# TRAINING" REPRESENT THE NUMBER OF INITIAL TRAINING DATA AND NEW TRAINING DATA, RESPECTIVELY, IN OUR INCREMENTAL LEARNING. FOR THE DSLR TARGET DOMAIN IN THE OBJECT DATASET, 6 IMAGES ARE SELECTED FROM EACH CATEGORY, AND THE REST IMAGES ARE USED FOR TESTING (I.E., "REST" IN THE COLUMN OF "TEST").

Dataset	domain	# labeled	# unlabeled	# test
IXMAS	Source	(6 , 30)	-	-
IXMAS	Target	(1 , 5)	(3 , 15)	12
Object	amazon	(8 , 12)	-	-
Object	webcam	(4 , 4)	-	-
Object	dslr	(1 , 2)	(1 , 2)	rest
Retuers	Source	(20 , 80)	-	-
Retuers	Target	(4 , 16)	(8 , 32)	60

For each class, we randomly sample labeled data from both source and target domains for training and also collect unlabeled target-domain samples as training data. During the incremental learning, the whole training dataset is partitioned into an initial subset which is used for learning the initial projection matrices and the remaining subsets which are added successively for re-training. Each subset consists of the labeled source training samples, the labeled target training samples, and the unlabeled target training samples. The test data from the target domain is unseen at the training phase. The detailed settings of training and test data for all the datasets are summarized in Table I.

B. Setup

To evaluate the effectiveness and efficiency of our method, we compare it with its batch mode in which the whole training dataset from both the source and target domains are completely given in advance. Our method is also compared with other state-of-the art methods [11], [12], [15], [5], [3] of domain adaptation on heterogeneous feature spaces. For our method, the batch mode, KCCA[11], HeMap[12] and DAMA [15], after learning the projection matrices, we apply SVM to train their final classifiers by using the projected training data from both domains. For ARC-t [5], we construct the kernel matrix based on the learned asymmetric transformation metric, and then SVM is also applied to t rain its final classifier. For HFA [3], the two projection matrices for the source and target data are found by using the standard SVM with the hingeloss. For all these methods, we set the regularization parameter $C = 1$ in SVM and use the RBF kernel for fair comparison. As we only have a very limited number of labeled training samples in the target domain, the cross-validation technique cannot be effectively employed to determine the optimal parameters. Instead, For our method and the batch mode, the parameter α is empirically set to $\alpha = 1$ for all the datasets. For other methods we validate all their parameters chosen from $\{0.01, 0.1, 1, 10, 100\}$ according to their best results on the test data.

C. Results

Cross-view action classification: Fig.1 summarizes the results of our method and the batch mode with the increasing training samples. For the limited paper space, we only report the mean of classification accuracies and the mean of computation time from all the source-target pairwise combinations. It is interesting to observe that our incremental method achieves approximate accuracy as the batch mode, provided that enough components of the total and between class scatter matrices are stored. We can also notice that as the training samples arrive successively, the execution time of the batch method increases more significantly than that of our method. Table II compares the mean of classification accuracies from all the individual

source domains for each target domain between different heterogeneous domain adaptation methods. Compared with KCCA [11] and HeMap [12], our method is able to learn a common feature space with discriminative ability by using the label information of the target training data. Our method outperforms DAMA [15], possibly due to the lack of the strong manifold structure on this dataset. The explanation for the better performance of our method than ARC-t [5] and HFA [3] may be that it utilizes unlabeled target-domain training data and incorporates the minimization of the distribution mismatch between source and target views in the objective function. The comparisons of mean computation time for each target domain between different methods are illustrated in Table III, from which we observe that our method is faster than all the other methods.

Object recognition: Fig.2 demonstrates the results of our method and the batch mode on the recognition accuracy and computation time, where the target domain is dslr and the source domain is amazon. Table IV and Table V report the recognition accuracies and computation time of our method and several related methods, respectively. In terms of recognition accuracy, our method achieves better results than most of the methods (i.e., KCCA, HeMap, DAMA and ARC-t) and is comparable to HFA. From the respective of computation efficiency, the training time of our method is much less than that of HFA, DAMA and ARC-t. KCCA and HeMap are faster than our method, possibly owing to the that KCCA and HeMap only utilize very limited labeled training samples from source and target domains to align the samples for training on this object dataset.

Text categorization: We report the recognition accuracies and computational time of our method and batch mode in Fig.3, where for each target domain we show the mean recognition accuracy and mean time of all the individual source domains. Table VI and VII compare the means of classification accuracies and computation time from all the individual source domains for each target domain, respectively. From the results, we have a similar observation as the object dataset.

From Table II, III, IV,V, VI and VII, it is important to note that our method achieves promising results on both recognition accuracy and computation time.

V. Conclusion

We have proposed a new incremental learning method to learn a discriminant common feature subspace for heterogeneous domain adaptation. Our method dose not require the whole training dataset to be given completely in advance and can incrementally update the model with the increasing input training samples, which effectively reduces the computational complexity and memory space during the learning. Extensive experiments on three benchmark datasets have demonstrated the promising results of our method.

References

[1] M. Amini, N. Usunier, and C. Goutte. Learning from multiple partially observed views - an application to multilingual text categorization. *NIPS*, 2009. 4

[2] H. Bay, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. *ECCV*, 2006. 4

[3] L. Duan, D. Xu, and I. Tsang. Learning with augmented features for heterogeneous domain adaptation. *ICML*, 2012. 1, 2, 5, 6, 7, 8

[4] M. Harel and S. Mannor. Learning from multiple outlooks. *ICML*, 2011. 1

[5] B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. *CVPR*, 2011. 1, 2, 4, 5, 6, 7, 8

[6] W. Li, L. Duan, D. Xu, and I. Tsang. Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation. *IEEE T-PAMI*, 36(6):1134–1148, 2014. 2

[7] P.M.Hall, D.Marshall, and R.Martin. Incremental eigenanalysis for classification. *BMVC*, 1998. 2

[8] P.M.Hall, D.Marshall, and R.Martin. Mergingand splittingeigenspace-models. *IEEE T-PAMI*, 22(9):1042–1049, 2000. 2

[9] P. Prettenhofer and B.Stein. Cross-language text classification using structural correspondence learning. *ACL*, 2010. 1

[10] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. *ECCV*, 2010. 4

[11] J. Shawe-Taylor and N. Cristianini. Kernel methods for pattern analysis. *Cambridge University Press*, 2004. 1, 5, 6, 7, 8

[12] X. Shi, Q. Liu, W. Fan, P. Yu, and R. Zhu. Transfer learning on heterogeneous feature spaces via spectral transformation. *ICDM*, 2010. 1, 5, 6, 7, 8

[13] S.N.Pang, S.Ozawa, and N.Kasabov. Incrementallineardiscriminantanal-ysisfor classification ofdatastreams. *IEEE T-SMC-Part B*, 35(5):905–914, 2005. 2

[14] T.K.Kim, J.Kittler, and R.Cipolla. Incrementallearningoflocallyorthog-onal subspaces forset-basedobjectrecognition. *BMVC*, 2006. 2

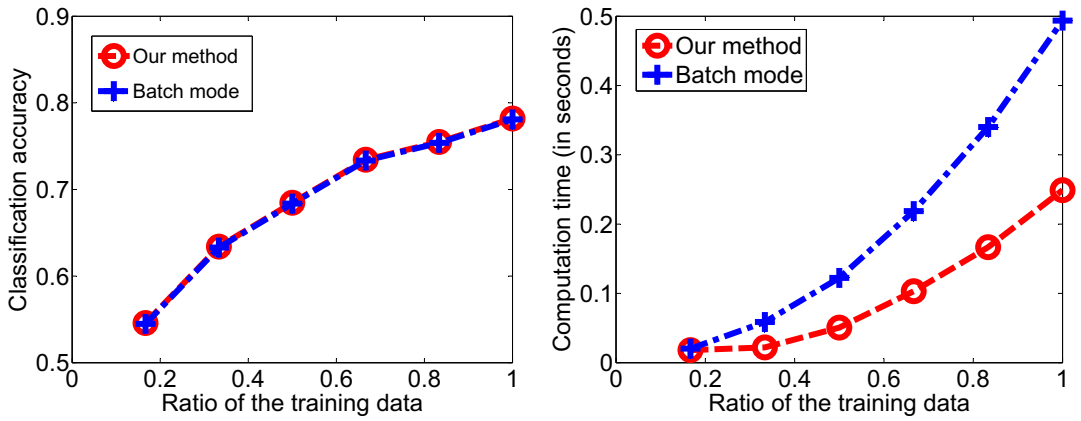
[15] C. Wang and S. Mahadevan. Heterogeneous domain adaptation using manifold alignment. *IJCAI*, 2011. 1, 2, 5, 6, 7, 8

[16] H. Wang, A. Klaer, C. Schmid, and C. Liu. Action recognition by dense trajectories. *CVPR*, 2011. 4

[17] D. Weinland, E. Boyer, and R. Ronfard. Action recognition from arbitrary views using 3d exemplars. *ICCV*, 2007. 4

[18] Y. Yeh, C. Huang, and Y. Wang. Heterogeneous domain adaptation and classification by exploiting the correlation subspace. *IEEE T-IP*, 23(5):2009–2018, 2014. 1

[19] Y. Zhu, Y. Chen, Z. Lu, S. Pan, G. Xue, Y. Yu, and Q. Yang. Heterogeneous transfer learning for image classification. *AAAI*, 2011. 1



(a) accuracy (b) time

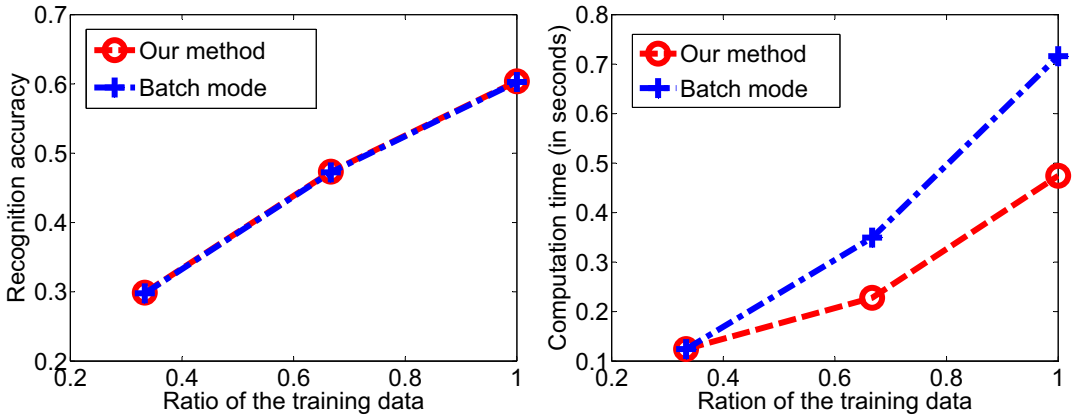
Fig. 1. Classification accuracies Fig.1(a) and computation time Fig.1(b) of our incremental method and the batch mode on the IXMAS action dataset.

TABLE II. MEAN CLASSIFICATION ACCURACIES (%) FOR EACH TARGET DOMAIN OF DIFFERENT HETEROGENEOUS DOMAIN ADAPTATION METHODS ON THE IXMAS ACTION DATASET.

	KCCA [11]	HeMap [12]	DAMA [15]	ARC-t [5]	HFA [3]	Our method
Target view1	79.86	71.70	82.81	80.21	81.94	85.42
Target view2	70.14	69.44	75.87	75.35	76.04	79.86
Target view3	69.79	67.01	75.87	74.48	75.87	77.08
Target view4	68.75	64.06	71.35	74.48	75.52	77.43
Target view5	60.24	57.64	64.58	64.93	68.06	70.83

TABLE III. MEAN COMPUTATION TIME (IN SECONDS) FOR EACH TARGET DOMAIN OF DIFFERENT HETEROGENEOUS DOMAIN ADAPTATION METHODS ON THE IXMAS ACTION DATASET.

	KCCA [11]	HeMap [12]	DAMA [15]	ARC-t [5]	HFA [3]	Our method
Target view1	0.3439	0.3270	1129	8.5464	5.7239	0.0628
Target view2	0.3509	0.3264	75.87	8.9131	5.8000	0.0655
Target view3	0.3431	0.3276	75.87	7.5702	5.7791	0.0651
Target view4	0.3465	0.3275	71.35	8.3719	25.3025	0.0617
Target view5	0.3436	0.3299	64.58	10.9421	24.8638	0.0632



(a) accuracy (b) time

Fig. 2. Classification accuracies Fig.2(a) and computation time Fig.2(b) of our incremental method and the batch mode on the object dataset.

TABLE IV. RECOGNITION ACCURACIES (%) OF DIFFERENT HETEROGENEOUS DOMAIN ADAPTATION METHODS ON THE OBJECT DATASET.

Source Domains	KCCA [11]	HeMap [12]	DAMA [15]	ARC-t [5]	HFA [3]	Our method
amazon	47.08	45.25	55.31	55.51	59.78	60.44
webcam	45.31	46,17	55.35	55.35	59.69	60.28

TABLE V. COMPUTATION TIME (IN SECONDS) OF DIFFERENT HETEROGENEOUS DOMAIN ADAPTATION METHODS ON THE OBJECT DATASET.

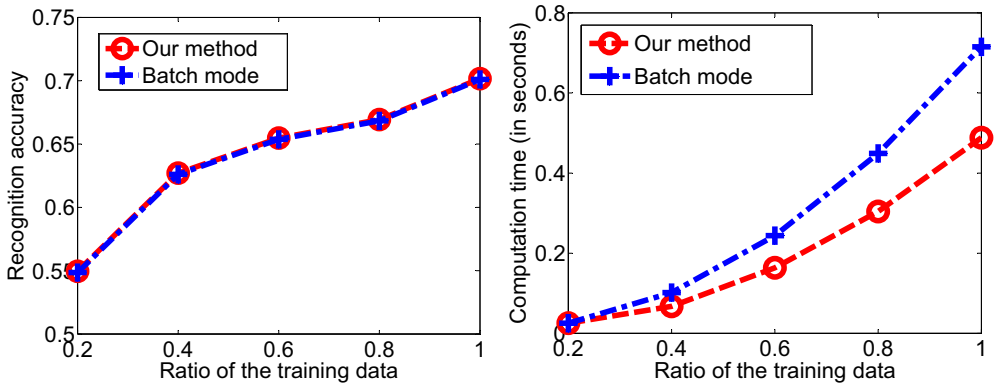
Source Domains	KCCA [11]	HeMap [12]	DAMA [15]	ARC-t [5]	HFA [3]	Our method
amazon	0.2024	0.1855	6.4908	4.7961	188.0412	0.4745
webcam	0.1098	0.0988	6.2916	2.7474	32.3577	0.0892

TABLE VI. MEAN CLASSIFICATION ACCURACIES (%) FOR EACH TARGET DOMAIN OF DIFFERENT HETEROGENEOUS DOMAIN ADAPTATION METHODS ON THE REUTERS MULTILINGUAL DATASET.

	KCCA [11]	HeMap [12]	DAMA [15]	ARC-t [5]	HFA [3]	Our method
English	58.96	61.67	58.96	67.40	69.90	70.52
French	66.04	68.12	66.04	70.21	70.83	69.79
German	66.98	67.50	66.98	71.35	72.92	74.17
Italian	60.70	63.44	60.73	68.33	68.33	68.65
Spanish	64.58	68.13	64.58	66.56	67.50	68.23

TABLE VII. MEAN COMPUTATION TIME (IN SECONDS) FOR EACH TARGET DOMAIN OF DIFFERENT HETEROGENEOUS DOMAIN ADAPTATION METHODS ON THE REUTERS MULTILINGUAL DATASET.

	KCCA [11]	HeMap [12]	DAMA [15]	ARC-t [5]	HFA [3]	Our method
English	0.3045	0.2155	29.4685	6.8384	27.7159	0.4872
French	0.2970	0.2131	33.0749	7.3404	27.9169	0.4859
German	0.2949	0.2177	41.3936	7.4873	27.7515	0.4915
Italian	0.2973	0.2118	26.2643	7.4517	39.0040	0.4833
Spanish	0.2906	0.2124	18.7847	7.3747	29.0110	0.4939



(a) accuracy (b) time

Fig. 3. Classification accuracies Fig.3(a) and computation time Fig.3(b) of our method and the batch mode on the Reuters multilingual dataset.