Unsupervised domain adaptation using eigenanalysis in kernel space for categorisation tasks

Article *in* IET Image Processing · August 2015

DOI: 10.1049/iet-ipr.2014.0754

2 authors:

Sukhendu Das
Indian Institute of Technology Madras
132 PUBLICATIONS **711** CITATIONS

SEE PROFILE

Suranjana Samanta
Indian Institute of Technology Madras
19 PUBLICATIONS **176** CITATIONS

SEE PROFILE

IET Image Processing

Research Article

IET Journals
The Institution of Engineering and Technology

Unsupervised domain adaptation using eigenanalysis in kernel space for categorisation tasks

ISSN 1751-9659
Received on 1st October 2014
Revised on 20th February 2015
Accepted on 18th June 2015
doi: 10.1049/iet-ipr.2014.0754
www.ietdl.org

Suranjana Samanta ✉, *Sukhendu Das*

VP Laboratory, Department of CS&E, Indian Institute of Technology, Madras, India
✉ *E-mail: ssamanta@cse.iitm.ac.in*

Abstract: This study describes a new technique of unsupervised domain adaptation based on eigenanalysis in kernel space, for the purpose of categorisation tasks. The authors propose a transformation of data in source domain, such that the eigenvectors and eigenvalues of the transformed source domain become similar to that of the target domain. They extend this idea to the reproducing kernel Hilbert space, which enables to deal with non-linear transformation of source domain. They also propose a measure to obtain the appropriate number of eigenvectors needed for transformation. Results on object, video and text categorisations tasks using real-world datasets show that the proposed method produces better results when compared with a few recent state-of-art methods of domain adaptation.

1 Introduction

Enormous amount of data generated everyday poses a challenge to analyse the data efficiently. Insufficient amount of labelled data in one dataset is overcome using the knowledge from the labelled examples of another dataset of a similar nature. In this paper, we deal with the problem of domain adaptation (DA), where knowledge from one dataset having a particular data distribution is used to improve classification performance on another dataset having a different data distribution.

DA has gained an increasing amount of attention in the recent past. In real-world scenarios, where the distributions of the training and the testing samples differ, DA can be also used to improve the performance of any statistical task (e.g. classification, regression etc.). The pair of domains from which the training and test samples are separately obtained, are termed as the source and target domains. By applying DA, one can use the training samples available from source domain to improve the performance of a statistical learning task to be done on testing samples obtained from the target domain.

To estimate the distribution of the target domain, few training samples must also be considered from the target domain. On the basis of the type of training samples available from the target domain, there are two categories of DA: (i) unsupervised – a large number of unlabelled samples and (ii) supervised – only a few number of labelled samples are available. Our work focuses on the former category of DA.

In this paper, we propose a technique of unsupervised DA using the eigenvectors and eigenvalues (EDA) of both the source and target domains. We first find an optimal linear transformation (termed EDA-L) of data in source domain such that the transformed source and target domains have similar eigenvectors and eigenvalues. Moreover, for non-linear projection of data, we use the concept of reproducing kernel Hilbert space (RKHS), and estimate a suitable transformation (termed EDA-K) in the (reduced) sub-space. An optimal dimension of the sub-space is also derived using an optimisation function. The method is fast, as eigendecomposition in lower dimensional sub-space (for EDA-L) and use of kernel Gram matrix (for EDA-K) help to reduce the time complexity in case of very large dimensional datasets. We evaluate our proposed method on real-world datasets for tasks, such as, object, video and text categorisation.

The rest of the paper is organised as follows. Section 2 briefly describes the related work published in literature. Section 3 describes the proposed methods of DA: using a linear transformation, which is later extended to a non-linear transformation with the help of RKHS. Section 4 describes the experimental results and Section 5 concludes the paper.

2 Related works

There has been a lot of work on DA in the recent past with various applications in the fields of computer vision, natural language processing (NLP) and text processing [1, 2]. A commonly used statistical approach is to estimate a domain invariant sub-space, such that the disparity (i.e. divergence) in the distributions of two domains is minimised in the projected space. Pan *et al.* [3] and Gretton *et al.* [4] have used this concept to build suitable sub-spaces. Most of these works perform the projection in RKHS to build a domain-invariant sub-space. Howard and Jebara [5] aligned the kernel Gram matrices of source and target domains such that the distance between two domains is reduced in RKHS.

Gopalan *et al.* [6] have considered the geodesic path between the principal components of source and target domain data in the Grassmann manifold, in his proposed method of Geodesic Flow Sampling (GFS). The intermediate sampled points on the path give an estimate of the continuous change of the properties of sub-spaces of source and target domains. This was later enhanced by Gong *et al.* [7] in his proposed method of Geodesic Flow Kernel (GFK), where an infinite number of intermediary sampled points are considered on the geodesic path, estimated by the geodesic flow kernel. In our earlier work of sequential domain invariant sub-space (SDIS) [8] method, we had considered a finite number of domain invariant sub-spaces lying along a suitable path on the Grassmann manifold, between the pair of sub-spaces spanning the source and the target domains in RKHS, to find an optimal feature set for DA. In another notable work, Fernando *et al.* [9] calculated a sub-space using eigenvectors of both the domains, such that the basis vectors of transformed source and target domains are aligned. Application of DA for improved results of object categorisation and video classification has also been discussed in [10, 11].

3 Proposed method

The proposed method uses eigenanalysis to minimise the disparity in distributions between the two domains. We aim to transform the

source domain in such a way that the eigenvectors and values of the covariance matrix of the transformed source domain are similar to that of the target domain. We extend this idea of transformation of source domain in RKHS to deal with non-linear transformation of data.

3.1 EDA-L: DA using linear transformation

Let $\boldsymbol{X}$ and $\boldsymbol{Y}$ be the data in source and target domains, with $n_{\boldsymbol{X}}$ and $n_{\boldsymbol{Y}}$ number of instances, respectively, and $\tilde{\boldsymbol{X}}$ be the transformed source domain. Let $\boldsymbol{X}_i$ and $\boldsymbol{Y}_j$ represent the ith and jth instances in $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively. Let the eigenvectors and eigenvalues of the covariance matrices of $\boldsymbol{X}$ and $\boldsymbol{Y}$ be denoted as $(\boldsymbol{U}, \boldsymbol{\Lambda})$ and $(\boldsymbol{V}, \boldsymbol{\Gamma})$ respectively. $\boldsymbol{\Lambda}$ and $\boldsymbol{\Gamma}$ are diagonal matrices whose ith diagonal element represents the ith largest eigenvalue of the covariance matrices of $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively. The ith column in $\boldsymbol{U}$ and $\boldsymbol{V}$ represents the eigenvector corresponding to the ith largest eigenvalue in $\boldsymbol{\Lambda}$ and $\boldsymbol{\Gamma}$, respectively.

The proposed method needs an optimal dimension (p^*) of the transformed source domain which is estimated using a dissimilarity criterion. Next, the transformed source domain is obtained by a linear transformation of data, such that the eigenvectors and the eigenvalues of the transformed source domain are similar to that of the target domain. Since the principal components of a dataset along with their spectrum (dimension-wise) is an estimate of the distribution of the data, we can infer that the disparity between the distributions of the source and target domains is minimised after the transformation. Generally, p^* is less than the dimension of $\boldsymbol{X}$ or $\boldsymbol{Y}$, and hence we consider the first p^* largest eigenvalues and the corresponding eigenvectors of both the source and target domains during transformation. The two steps of EDA-L are described below.

3.1.1 Finding the optimal dimension: Let $\boldsymbol{U}_p$ and $\boldsymbol{V}_p$ represent the matrices whose columns consist of the first p eigenvectors corresponding to the p largest eigenvalues. Hence, $\boldsymbol{U}_p$ and $\boldsymbol{V}_p$ consists of the first p columns of $\boldsymbol{U}$ and $\boldsymbol{V}$, respectively. Similarly, let $\boldsymbol{\Lambda}_p$ and $\boldsymbol{\Gamma}_p$ denote the diagonal sub-matrices containing the first p largest eigenvalues as the diagonal elements. To find the optimal number of eigenvectors and eigenvalues to be considered for transforming the data in source domain, we estimate the square of the distance between sub-spaces, as given by

$$\delta^2_{\text{proj}}\left(\boldsymbol{U}_p, \boldsymbol{V}_p\right) = \sum_{i=1}^{p} \sin^2\theta_i = p - \sum_{i=1}^{p} \cos^2\theta_i \tag{1}$$

where θ_i, $i = 1, 2, \ldots p$ are the principal angles between these sub-spaces. The span of sub-spaces $\boldsymbol{U}_p$ and $\boldsymbol{V}_p$ are given by $\text{span}(\boldsymbol{U}_p) = \boldsymbol{U}_p\boldsymbol{U}_p^{\text{T}}$ and $\text{span}(\boldsymbol{V}_p) = \boldsymbol{V}_p\boldsymbol{V}_p^{\text{T}}$, respectively, and $\sum_{i=1}^{p}\cos^2\theta_i = \langle \text{span}(\boldsymbol{U}_p), \text{span}(\boldsymbol{V}_p)\rangle$. Let $\text{tr}(\boldsymbol{A})$ denote the trace of a matrix $\boldsymbol{A}$. Then, the square of the projection distance (as used in [7, 8, 12]) is expressed as

$$\begin{aligned}\delta^2_{\text{proj}}(\boldsymbol{U}_p, \boldsymbol{V}_p) &= p - \text{tr}\left(\boldsymbol{U}_p\boldsymbol{U}_p^{\text{T}}\boldsymbol{V}_p\boldsymbol{V}_p^{\text{T}}\right)\\ &= p - \text{tr}\left(\boldsymbol{V}_p^{\text{T}}\boldsymbol{U}_p\boldsymbol{U}_p^{\text{T}}\boldsymbol{V}_p\right)\end{aligned} \tag{2}$$

The appropriate dimension of the optimal sub-space is solved by maximising the following function

$$p^* = \underset{p}{\text{argmax}}\ \frac{1}{p}\delta^2_{\text{proj}}(\boldsymbol{U}_p, \boldsymbol{V}_p) \tag{3}$$

$$= \underset{p}{\text{argmax}}\ \frac{1}{p}\text{tr}\left(\boldsymbol{V}_p^{\text{T}}\boldsymbol{U}_p\boldsymbol{U}_p^{\text{T}}\boldsymbol{V}_p\right) \tag{4}$$

The term $1/p$ avoids the bias selection for a very low value of p^* (i.e. $p^* = 1$) as the optimal dimension of the sub-space to be considered. This concept (4) is similar to the sub-space disagreement measure [7], where the aim is to consider the optimal dimension of the sub-space such that the source and target domains are closest to each other in the Grassmann manifold.

3.1.2 Estimating the transformed source domain: Once we obtain the optimal value of p^*, we estimate the optimal sub-space of dimension p^*. If the transformed source domain data is given by

$$\tilde{\boldsymbol{X}} = \boldsymbol{X}\boldsymbol{U}_{p*}\boldsymbol{\Lambda}_{p*}^{-1/2}\boldsymbol{\Gamma}_{p*}^{1/2}\boldsymbol{V}_{p*}^{\text{T}} \tag{5}$$

then the eigenvectors and eigenvalues of $\tilde{\boldsymbol{X}}$ and $\boldsymbol{Y}$ are the same (see Lemma 1). Using (5), a linear transformation of the source domain data reduces the difference in distributions between the two domains by: (i) aligning their principal component vectors and (ii) making the eigenspectrums identical.

Lemma 1: Given $(\hat{\boldsymbol{U}}, \hat{\boldsymbol{\Lambda}})$ and $(\hat{\boldsymbol{V}}, \hat{\boldsymbol{\Gamma}})$ as the pairs of eigenvectors, eigenvalues of the covariance matrices of two datasets $\boldsymbol{A}$ and $\boldsymbol{B}$, then the eigenanalysis of the covariance matrix of $\boldsymbol{A}\hat{\boldsymbol{U}}\boldsymbol{\Lambda}^{-1/2}\hat{\boldsymbol{\Gamma}}^{1/2}\hat{\boldsymbol{V}}^{\boldsymbol{T}}$ is the same as that of $\boldsymbol{B}$.

Proof: The covariance matrices of datasets $\boldsymbol{A}$ and $\boldsymbol{B}$ can be expressed as $\hat{\boldsymbol{U}}\hat{\boldsymbol{\Lambda}}\hat{\boldsymbol{U}}^{\boldsymbol{T}}$ and $\hat{\boldsymbol{V}}\hat{\boldsymbol{\Gamma}}\hat{\boldsymbol{V}}^{\boldsymbol{T}}$, respectively. Now, the covariance matrix of $\tilde{\boldsymbol{A}}\left(=\boldsymbol{A}\hat{\boldsymbol{U}}\hat{\boldsymbol{\Lambda}}^{-1/2}\hat{\boldsymbol{\Gamma}}^{1/2}\hat{\boldsymbol{V}}^{\boldsymbol{T}}\right)$ is given as

$$\tilde{\boldsymbol{A}}^{\boldsymbol{T}}\tilde{\boldsymbol{A}} = \hat{\boldsymbol{V}}\hat{\boldsymbol{\Gamma}}^{1/2}\hat{\boldsymbol{\Lambda}}^{-1/2}\hat{\boldsymbol{U}}^{\boldsymbol{T}}\boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{A}\hat{\boldsymbol{U}}\hat{\boldsymbol{\Lambda}}^{-1/2}\hat{\boldsymbol{\Gamma}}^{1/2}\hat{\boldsymbol{V}}^{\boldsymbol{T}} \tag{6}$$

$$= \hat{\boldsymbol{V}}\hat{\boldsymbol{\Gamma}}^{1/2}\hat{\boldsymbol{\Lambda}}^{-1/2}\hat{\boldsymbol{U}}^{\boldsymbol{T}}\hat{\boldsymbol{U}}\hat{\boldsymbol{\Lambda}}\hat{\boldsymbol{U}}^{\boldsymbol{T}}\hat{\boldsymbol{U}}\hat{\boldsymbol{\Lambda}}^{-1/2}\hat{\boldsymbol{\Gamma}}^{1/2}\hat{\boldsymbol{V}}^{\boldsymbol{T}} \tag{7}$$

$$= \hat{\boldsymbol{V}}\hat{\boldsymbol{\Gamma}}\hat{\boldsymbol{V}}^{\boldsymbol{T}} \tag{8}$$

This shows that the eigenvectors and eigenvalues of the covariance matrix of $\tilde{\boldsymbol{A}}$ are same as that of $\boldsymbol{B}$. □

This process (5) must be followed by shifting of $\tilde{\boldsymbol{X}}$ such that its mean is identical to that of the target domain data $\boldsymbol{Y}$. This is necessary to align the spatial arrangements of data in both the domains, which improves the classification accuracy.

A similar method had earlier been proposed for sub-space alignment (SA) [9], where instead of transforming the data the eigenvectors of source domain are rotated to align with that of the target domain. However, the eigenspectrum was not considered and thus the range of the corresponding feature values in transformed source and target domains were not coincident. While the proposed method transforms the source domain data such that the underlying sub-space aligns with that of the target domain, SA first aligns the two sub-spaces and then projects the data onto respective sub-spaces. It can be analytically verified that the sub-space spanning the projected source domain data will be different from that of the sub-space spanning the target domain.

The proposed method of EDA-L (5) thus works best for datasets having Gaussian-like distributions in the feature space. We extend EDA-L to RKHS to handle non-linear transformation of data, which is termed as EDA-K. In higher dimensional kernel space, the main advantages are: (i) non-linear transformations are implicitly incorporated and (ii) datasets having non-Gaussian distributions are less affected. The details of EDA-K are explained in the next sub-section.

3.2 EDA-K: extension of EDA-L to RKHS

EDA-L performs linear transformation of the source domain data. To handle non-linear transformation of data, we extend the formulation to RKHS. If $\Phi(.)$ is a kernel function, then in kernel space the source and target domains are denoted by $\Phi(\boldsymbol{X})$ and $\Phi(\boldsymbol{Y})$, respectively. Let $\boldsymbol{K}_{\boldsymbol{XX}}$ and $\boldsymbol{K}_{\boldsymbol{YY}}$ be the Gram matrices of $\Phi(\boldsymbol{X})$ and $\Phi(\boldsymbol{Y})$, respectively (i.e. $\boldsymbol{K}_{\boldsymbol{XX}} = \Phi(\boldsymbol{X})\Phi(\boldsymbol{X})^{\text{T}}$, $\boldsymbol{K}_{\boldsymbol{XY}} = \Phi(\boldsymbol{X})\Phi(\boldsymbol{Y})^{\text{T}}$, $\boldsymbol{K}_{\boldsymbol{YY}} = \Phi(\boldsymbol{Y})\Phi(\boldsymbol{Y})^{\text{T}}$ and $\boldsymbol{K}_{\boldsymbol{YX}} = \boldsymbol{K}_{\boldsymbol{XY}}^{\text{T}}$). Let $\boldsymbol{U}^{\boldsymbol{\Phi}}$ and $\boldsymbol{V}^{\boldsymbol{\Phi}}$ be the principal components of $\Phi(\boldsymbol{X})$ and $\Phi(\boldsymbol{Y})$. Also, let $\left(\boldsymbol{\Psi}^{\boldsymbol{\Phi}}, \boldsymbol{\Lambda}^{\boldsymbol{\Phi}}\right)$ and $(\boldsymbol{\Omega}^{\Phi}, \boldsymbol{\Gamma}^{\Phi})$ be the pairs of eigenvectors and eigenvalues of $\boldsymbol{K}_{\boldsymbol{XX}}$ and $\boldsymbol{K}_{\boldsymbol{YY}}$. Then, the

principal components of $\Phi(\boldsymbol{X})$ and $\Phi(\boldsymbol{Y})$ are (see [13])

$$\boldsymbol{U}^{\boldsymbol{\Phi}} = \boldsymbol{\Phi}(\boldsymbol{X})^{\mathrm{T}}\boldsymbol{\Psi}^{\boldsymbol{\Phi}} \tag{9}$$

$$\boldsymbol{V}^{\boldsymbol{\Phi}} = \boldsymbol{\Phi}(\boldsymbol{Y})^{\mathrm{T}}\boldsymbol{\Omega}^{\boldsymbol{\Phi}} \tag{10}$$

Now, let the singular value decomposition of $\Phi(\boldsymbol{X})$ be $\boldsymbol{Z}_{\boldsymbol{L}}\boldsymbol{\Sigma}\boldsymbol{Z}_{\boldsymbol{R}}^{\mathrm{T}}$. Then, the eigendecomposition of the covariance matrix, $\Phi(\boldsymbol{X})^{\mathrm{T}}\Phi(\boldsymbol{X})/N$, is $\boldsymbol{Z}_{\boldsymbol{R}}\boldsymbol{\Upsilon}^{\boldsymbol{\Phi}}\boldsymbol{Z}_{\boldsymbol{R}}^{\mathrm{T}}$, where $\boldsymbol{\Upsilon}^{\boldsymbol{\Phi}}(=\boldsymbol{\Sigma}^2/N)$ is the diagonal matrix containing the eigenvalues where, $N = n_{\boldsymbol{X}}$. Similarly, the eigendecomposition of the kernel Gram matrix $\Phi(\boldsymbol{X})\Phi(\boldsymbol{X})^{\mathrm{T}}$ is $\boldsymbol{Z}_{\boldsymbol{L}}\boldsymbol{\Lambda}^{\boldsymbol{\Phi}}\boldsymbol{Z}_{\boldsymbol{L}}^{\mathrm{T}}$, where $\boldsymbol{\Lambda}^{\boldsymbol{\Phi}} = \boldsymbol{\Sigma}^2$. This may seem ambiguous, as even though the dimensions of $\boldsymbol{\Upsilon}^{\boldsymbol{\Phi}}$ and $\boldsymbol{\Lambda}^{\boldsymbol{\Phi}}$ are potentially different, the following can be written [14]: $\boldsymbol{\Upsilon}^{\boldsymbol{\Phi}} \times N = \boldsymbol{\Lambda}^{\boldsymbol{\Phi}} = \boldsymbol{\Sigma}^2$. Note that the larger matrix (say $\boldsymbol{\Upsilon}^{\boldsymbol{\Phi}}$) is an extension of the smaller one ($\boldsymbol{\Lambda}^{\boldsymbol{\Phi}}$) with zero padding. Hence, we can use the matrix $\boldsymbol{\Lambda}^{\boldsymbol{\Phi}}/N$ to represent the eigenvalues of the covariance matrix of $\Phi(\boldsymbol{X})$. Accordingly, $\boldsymbol{\Gamma}^{\Phi}/N$ can be used to represent the eigenvalues of the covariance matrix of $\Phi(\boldsymbol{Y})$.

Then, the distance between sub-spaces of dimension p, in the kernel space is

$$\delta^2_{\mathrm{proj}}\left(\boldsymbol{U}_p^{\boldsymbol{\Phi}}, \boldsymbol{V}_p^{\boldsymbol{\Phi}}\right) = p - \mathrm{tr}\left(\boldsymbol{V}_p^{\boldsymbol{\Phi}^{\mathrm{T}}}\boldsymbol{U}_P^{\boldsymbol{\Phi}}\boldsymbol{U}_P^{\boldsymbol{\Phi}^{\mathrm{T}}}\boldsymbol{V}_p^{\boldsymbol{\Phi}}\right) \tag{11}$$

$$= p - \mathrm{tr}\left(\boldsymbol{\Omega}_p^{\boldsymbol{\Phi}^{\mathrm{T}}}\boldsymbol{\Phi}(\boldsymbol{Y})\boldsymbol{\Phi}(\boldsymbol{X})^{\mathrm{T}}\boldsymbol{\Psi}^{\boldsymbol{\Phi}}\boldsymbol{\Psi}^{\boldsymbol{\Phi}^{\mathrm{T}}}\boldsymbol{\Phi}(\boldsymbol{X})\boldsymbol{\Phi}(\boldsymbol{Y})^{\mathrm{T}}\boldsymbol{\Omega}_p^{\boldsymbol{\Phi}}\right) \tag{12}$$

$$= p - \mathrm{tr}\left(\boldsymbol{\Omega}_p^{\boldsymbol{\Phi}^{\mathrm{T}}}\boldsymbol{K}_{\boldsymbol{XY}}^{\mathrm{T}}\boldsymbol{\Psi}_p^{\boldsymbol{\Phi}}\boldsymbol{\Psi}_p^{\boldsymbol{\Phi}^{\mathrm{T}}}\boldsymbol{K}_{\boldsymbol{XY}}\boldsymbol{\Omega}_p^{\boldsymbol{\Phi}}\right) \tag{13}$$

Hence, the appropriate dimension to be considered for transformation (as in (4)) is

$$p^* = \underset{p}{\mathrm{argmax}}\ \frac{1}{p}\mathrm{tr}\left(\boldsymbol{\Omega}_p^{\boldsymbol{\Phi}^{\mathrm{T}}}\boldsymbol{K}_{\boldsymbol{XY}}^{\mathrm{T}}\boldsymbol{\Psi}_p^{\boldsymbol{\Phi}}\boldsymbol{\Psi}_p^{\boldsymbol{\Phi}^{\mathrm{T}}}\boldsymbol{K}_{\boldsymbol{XY}}\boldsymbol{\Omega}_p^{\boldsymbol{\Phi}}\right) \tag{14}$$

Once we obtain the appropriate number of dimensions to be considered, we estimate the transformed source domain in RKHS as (similar to (5))

$$\boldsymbol{\Phi}(\tilde{\boldsymbol{X}}) = \frac{1}{N}\boldsymbol{\Phi}(\boldsymbol{X})\boldsymbol{U}_{p*}^{\boldsymbol{\Phi}}\boldsymbol{\Lambda}_{p*}^{\boldsymbol{\Phi}-1/2}\boldsymbol{\Gamma}_{p*}^{\boldsymbol{\Phi}1/2}\boldsymbol{V}_{p*}^{\boldsymbol{\Phi}\mathrm{T}} \tag{15}$$

The corresponding Gram matrices are given by (using (9), (10) and (15))

$$\begin{aligned}\boldsymbol{K}_{\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}} &= \boldsymbol{\Phi}(\tilde{\boldsymbol{X}})\boldsymbol{\Phi}(\tilde{\boldsymbol{X}})^{\mathrm{T}} \\ &= \frac{1}{N^2}\boldsymbol{K}_{\boldsymbol{XX}}\boldsymbol{\Psi}_{p*}^{\boldsymbol{\Phi}}\boldsymbol{\Lambda}_{p*}^{\boldsymbol{\Phi}-1/2}\boldsymbol{\Gamma}_{p*}^{\boldsymbol{\Phi}1/2}\boldsymbol{\Omega}_{p*}^{\boldsymbol{\Phi}\mathrm{T}} \\ &\quad \times \boldsymbol{K}_{\boldsymbol{YY}}\boldsymbol{\Omega}_{p*}^{\boldsymbol{\Phi}}\boldsymbol{\Gamma}_{p*}^{\boldsymbol{\Phi}1/2}\boldsymbol{\Lambda}_{p*}^{\boldsymbol{\Phi}-1/2}\boldsymbol{\Omega}_{p*}^{\boldsymbol{\Phi}\mathrm{T}}\boldsymbol{K}_{\boldsymbol{XX}}\end{aligned} \tag{16}$$

$$\boldsymbol{K}_{\tilde{\boldsymbol{X}}\boldsymbol{Y}} = \boldsymbol{\Phi}(\tilde{\boldsymbol{X}})\boldsymbol{\Phi}(\boldsymbol{Y})^{\mathrm{T}} = \frac{1}{N}\boldsymbol{K}_{\boldsymbol{XX}}\boldsymbol{\Psi}_{p*}^{\boldsymbol{\Phi}}\boldsymbol{\Lambda}_{p*}^{\boldsymbol{\Phi}-1/2}\boldsymbol{\Gamma}_{p*}^{\boldsymbol{\Phi}1/2}\boldsymbol{\Omega}_{p*}^{\boldsymbol{\Phi}\mathrm{T}}\boldsymbol{K}_{\boldsymbol{YY}} \tag{17}$$

If $\boldsymbol{o}_{\boldsymbol{X}} \in \mathbb{R}^{n_X}$ and $\boldsymbol{o}_{\boldsymbol{Y}} \in \mathbb{R}^{n_Y}$ denote two vectors with all elements as $1/n_{\boldsymbol{X}}$ and ,$1/n_{\boldsymbol{Y}}$ respectively, then in RKHS the means of source domain after transformation and that of target domain, are given by $\boldsymbol{o}_{\boldsymbol{X}}^{\mathrm{T}}\boldsymbol{\Phi}(\tilde{\boldsymbol{X}})$ and $\boldsymbol{o}_{\boldsymbol{Y}}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{Y})$. Next, it is necessary to modify the Gram matrices appropriately, such that the mean of the transformed source and target domains are identical in kernel space. Let for any arbitrary matrix $\boldsymbol{A}$, the ith row of $\boldsymbol{A}$ be denoted by $\boldsymbol{A}(i, \bullet)$ and the jth column by $\boldsymbol{A}(\bullet, j)$. If $\hat{\boldsymbol{K}}_{\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}}$ represents the mean-shifted Gram matrix, then each element of this matrix is calculated as

$$\begin{aligned}\hat{\boldsymbol{K}}_{\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}}(i,j) &= \begin{pmatrix}\boldsymbol{\Phi}(\tilde{\boldsymbol{X}}_i) - \boldsymbol{o}_{\boldsymbol{X}}^{\mathrm{T}}\boldsymbol{\Phi}(\tilde{\boldsymbol{X}}) + \boldsymbol{o}_{\boldsymbol{Y}}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{Y}))\\ (\boldsymbol{\Phi}(\tilde{\boldsymbol{X}}_i) - \boldsymbol{o}_{\boldsymbol{X}}^{\mathrm{T}}\boldsymbol{\Phi}(\tilde{\boldsymbol{X}}) + \boldsymbol{o}_{\boldsymbol{Y}}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{Y})\end{pmatrix}^{\mathrm{T}} \\ &= \boldsymbol{K}_{\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}}(i,j) - \boldsymbol{K}_{\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}}(i,\bullet)\boldsymbol{o}_{\boldsymbol{X}} + \boldsymbol{K}_{\tilde{\boldsymbol{X}}\boldsymbol{Y}}(i,\bullet)\boldsymbol{o}_{\boldsymbol{Y}} \\ &\quad - \boldsymbol{o}_{\boldsymbol{X}}^{\mathrm{T}}\boldsymbol{K}_{\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}}(\bullet,j) + \boldsymbol{o}_{\boldsymbol{X}}^{\mathrm{T}}\boldsymbol{K}_{\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}}\boldsymbol{o}_{\boldsymbol{X}} \\ &\quad - \boldsymbol{o}_{\boldsymbol{X}}^{\mathrm{T}}\boldsymbol{K}_{\tilde{\boldsymbol{X}}\boldsymbol{Y}}\boldsymbol{o}_{\boldsymbol{Y}} + \boldsymbol{o}_{\boldsymbol{Y}}^{\mathrm{T}}\boldsymbol{K}_{\boldsymbol{Y}\tilde{\boldsymbol{X}}}(\bullet,j) - \boldsymbol{o}_{\boldsymbol{Y}}^{\mathrm{T}}\boldsymbol{K}_{\boldsymbol{Y}\tilde{\boldsymbol{X}}}\boldsymbol{o}_{\boldsymbol{X}} + \boldsymbol{o}_{\boldsymbol{Y}}^{\mathrm{T}}\boldsymbol{K}_{\boldsymbol{YY}}\boldsymbol{o}_{\boldsymbol{Y}}\end{aligned} \tag{18}$$

Similarly, each element of the mean shifted Gram matrix $\hat{\boldsymbol{K}}_{\tilde{\boldsymbol{X}}\boldsymbol{Y}}$ is calculated as

$$\begin{aligned}\hat{\boldsymbol{K}}_{\tilde{\boldsymbol{X}}\boldsymbol{Y}}(i,j) &= \left(\boldsymbol{\Phi}(\tilde{\boldsymbol{X}}_i) - \boldsymbol{o}_{\boldsymbol{X}}^{\mathrm{T}}\boldsymbol{\Phi}(\tilde{\boldsymbol{X}}) + \boldsymbol{o}_{\boldsymbol{Y}}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{Y})\right).\boldsymbol{\Phi}\left(\boldsymbol{Y}_j\right)^{\mathrm{T}} \\ &= \boldsymbol{K}_{\tilde{\boldsymbol{X}}\boldsymbol{Y}}(i,j) - \boldsymbol{o}_{\boldsymbol{X}}^{\mathrm{T}}\boldsymbol{K}_{\tilde{\boldsymbol{X}}\boldsymbol{Y}}(\bullet,j) + \boldsymbol{o}_{\boldsymbol{Y}}^{\mathrm{T}}\boldsymbol{K}_{\boldsymbol{YY}}(\bullet,j)\end{aligned} \tag{19}$$

In RKHS, we thus calculate the modified kernel Gram matrices using the implicit expressions ((18), (19)) of the transformed source domain.

3.3 Classification of test dataset

Let the test samples to be classified be given by the rows in the matrix, $\boldsymbol{W}$. We calculate $\hat{\boldsymbol{K}}_{\tilde{\boldsymbol{X}}\boldsymbol{W}}$ by replacing $\boldsymbol{Y}$ with $\boldsymbol{W}$ in (19). Once we obtain the Gram matrices $\hat{\boldsymbol{K}}_{\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}}$ and $\hat{\boldsymbol{K}}_{\tilde{\boldsymbol{X}}\boldsymbol{W}}$, we first calculate the overall Gram matrix

$$\hat{\boldsymbol{K}} = \begin{bmatrix}\hat{\boldsymbol{K}}_{\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}} & \hat{\boldsymbol{K}}_{\tilde{\boldsymbol{X}}\boldsymbol{W}} \\ \hat{\boldsymbol{K}}_{\tilde{\boldsymbol{X}}\boldsymbol{W}}^{\mathrm{T}} & \boldsymbol{K}_{\boldsymbol{WW}}\end{bmatrix}.$$

The Euclidean distance between any two data instances (i and j) in RKHS is then given as: $\mathrm{dist}(i,j) = \hat{\boldsymbol{K}}(i,i) + \hat{\boldsymbol{K}}(j,j) - 2\hat{\boldsymbol{K}}(i,j)$. This distance matrix helps to classify test samples appropriately, using k-nearest neighbour (kNN) classifier, trained using transformed source domain data in RKHS.

4 Experimental results

We evaluate the performances of EDA-L and EDA-K on three real-world datasets, for the tasks of object, video and text categorisation. The details of the experimental results are given below.

4.1 Object categorisation

We use the Office+Caltech dataset [7], which contains four domains: Amazon (A), Caltech (C), Dslr (D) and Webcam (W), with ten classes of objects in each domain. Each image is resized to 300 × 300 dimension and speeded-up robust features [15] are extracted from the images to form a codebook of size 800. A few sample images from the four domains are shown in Fig. 1. The results reported in this paper are obtained by using the features shared by Gong *et al.* [7], from which we follow the same experimental protocols. For EDA-K, we have used linear kernel to build a generic model for all datasets, avoiding any parameter tuning for obtaining the appropriate kernel Gram matrix. However, any kernel function can be used with appropriate values. Eight random samples per class for the Amazon and Caltech domains, three random samples per class for the Dslr and Webcam domains, have been chosen when each of these domains are being considered as the target domain for the experimentation. We compare the performances of our methods with transfer component analysis (TCA) [3], geodesic flow sub-space [6], geodesic flow kernel [7], SA [9] and SDIS [8]; while NA denotes 'no adaptation', where the given source domain samples are used for training the classifier.

Fig. 1 *Samples of two object classes, from four domains of Office+Caltech dataset [7]*

Table 1 Classification accuracies (in %-age) on Office+Caltech dataset [7], using different techniques of unsupervised DA (best results highlighted in bold)

Method	C → A	D → A	W → A	A → C	D → C	W → C	
NA	21.5	26.9	20.8	22.8	24.8	16.4	
TCA [3]	21.96	16.81	13.43	16.18	17.67	11.14	
GFS [6]	36.9	32	27.5	35.3	29.4	21.7	
GFK [7]	36.9	32.5	31.1	35.6	29.8	27.2	
SA [9]	39.0	38.0	37.4	35.3	32.4	32.3	
SDIS [8]	**42.63**	44.16	44.65	34.40	**41.56**	**43.26**	
EDA-L(LDA)	24.69	19.83	21.38	13.78	26.20	21.04	
EDA-L(PCA)	27.79	21.42	26.42	17.78	26.00	25.50	
EDA-K	40.66	**44.41**	**46.81**	**36.62**	37.02	36.21	
Method	A → D	C → D	W → D	A → W	C → W	D → W	Average
NA	22.4	21.7	40.5	23.3	20.0	53.0	26.18
TCA [3]	16.69	22.8	32.31	23.60	22.03	44.69	21.61
GFS [6]	30.7	32.6	54.3	31.0	30.6	66.0	35.67
GFK [7]	35.2	35.2	70.6	34.4	33.7	74.9	39.76
SA [9]	37.6	39.6	80.3	38.60	36.80	83.6	44.24
SDIS [8]	38.82	43.64	80.57	39.31	**42.27**	78.03	47.76
EDA-L(LDA)	18.91	30.44	51.60	31.97	24.84	45.44	27.52
EDA-L(PCA)	26.77	29.92	58.50	32.83	30.19	62.64	32.15
EDA-K	**43.31**	**44.09**	**85.82**	**40.00**	38.49	**85.28**	**48.22**

As a special case, we also obtain the performance of DA when the sub-space of the source domain is estimated using the traditional supervised method of linear discriminant analysis (LDA). LDA obtains an optimal sub-space where the class separation is maximum. However, due to the absence of class labels in the target domain, we use principal component analysis (PCA) to estimate the corresponding sub-space. We varied the number of dimensions to be considered from 1 to $C-1$ (C = number of classes) and only the best accuracies for each case have been reported in Table 1.

Table 1 shows the classification accuracies for 12 different pairs of source and target domains, using a 25-fold cross-validation. Results given in seventh row as 'EDA-L(LDA)' gives the classification accuracy when sub-spaces are calculated using LDA for source domain and PCA for target domain. Results obtained using EDA-L(PCA) shows the result of our approach, when the sub-spaces of both source and target domains are estimated using PCA. Results given in ninth row give the classification accuracies when the transformation of source domain is done in RKHS ((18), (19)). The right-most column of Table 1 (see lower sub-part) shows the average of classification accuracies over all 12 cases.

EDA-K outperforms all the other cases of DA. Even though we use class information to estimate the source domain sub-space in case of EDA-L(LDA), the average classification accuracy is much lower. This is due to the fact that sub-space estimated using LDA gives the maximum class separability but do not characterise the distribution of the overall data. Since the correspondences between the sub-spaces estimated by PCA and LDA is missing, EDA-L (LDA) produces a poor performance of object categorisation (seventh row of Table 1).

SA performs close (but inferior) to our method, as it directly uses eigenvectors of source and target domains in feature space to formulate the transformation matrix. Our recently proposed SDIS method [8] performs better in a few cases, but the average performance is best for EDA-K. The average time necessary for the task of object categorisation using EDA-K is 61 s, while that of SDIS is 437 s. SDIS [8] uses manifold-based optimisation, whereas the proposed method uses eigenvectors and eigenvalues, which is computationally efficient. Another added advantage is that the optimal number of dimensions is automatically chosen for this proposed method, which makes it adaptable to various datasets. Table 2 shows the optimal number of dimensions/ eigenvectors considered for transformation of the source domain in both the feature and kernel spaces.

4.2 Event categorisation in videos

We use two video datasets: Kodak and YouTube [16]. YouTube data is considered as the source domain and classification accuracies are observed on Consumer (Kodak) domain, as done in [17]. We consider six common classes (events), 'birthday', 'picnic', 'parade', 'show', 'sports' and 'wedding', between YouTube (906

Table 2 Optimal number of dimensions obtained ((4), (14)) for transforming the source domain data, in case of Office+Caltech dataset [7]

Method	C → A	D → A	W → A	A → C	D → C	W → C
EDA-L(PCA)	209	231	128	157	144	162
EDA-K	18	23	27	10	9	35
Method	A → D	C → D	W → D	A → W	C → W	D → W
EDA-L(PCA)	183	156	164	239	243	251
EDA-K	15	11	12	19	17	9

Table 3 Values of MAPs (in %-age) for event categorisation in videos [16], using different techniques of DA (best results highlighted in bold)

Method	SIFT feature				ST feature				Average
	Gaussian	Laplacian	ISD	ID	Gaussian	Laplacian	ISD	ID	
A-MKL [16]	50.4	**53.8**	52.9	51.0	20.6	35.8	22.3	35.9	40.3
TCA [3]	42.6	40.8	43.4	39.7	17.5	19.1	18.2	18.4	30.0
NA	22.3	28.4	24.8	22.5	15.6	22.3	27.4	25.5	23.6
EDA-K(*k*NN)	49.3	52.9	52.6	**52.5**	23.6	**37.2**	26.7	**36.4**	**41.4**
EDA-K(SVM)	**51.3**	41.6	**55.2**	44.3	**40.6**	25.5	**40.9**	24.5	40.5

videos) and Consumer (195 videos) [16]. We have used the distance matrices of Kodak and YouTube domains, with scale invariant feature transform (SIFT) descriptors and spatio-temporal (ST) features – histogram of oriented gradients (HOG) and histogram of optical flow (HOF) features [http://vc.sce.ntu.edu.sg/index_files/VisualEventRecognition/VisualEventRecognition.html]. Three samples per class have been randomly selected from the target domain for training. We have used four different kernel functions for evaluation as in [16], which are: Gaussian, Laplacian, inverse square distance (ISD) and inverse distance (ID) kernel.

Performances over ten-fold experimentation have been compared with adaptive multiple kernel learning (A-MKL) [16], TCA [3], as well as the case of NA (no adaptation). Since, our experiments use the distance matrix as input, we are unable to obtain the performances of GFS [6], GFK [7] and SA [9] methods for this task. Table 3 shows the mean average precision (MAP), using both SIFT and ST features separately, using a 25-fold cross-validation. We have trained both *k*NN ($k=1$) and support vector machine (SVM) classifiers, using the transformed source domain. Results in Table 3 show that the proposed EDA-K method outperforms TCA. In case of ST features, EDA-K outperforms all other methods when *k*NN classifier is used. However, A-MKL [16] is better than the proposed method in one case, only when SIFT features are used to build the three different kernel Gram matrices. Note that, although A-MKL uses labelled training samples from target domain unlike the proposed method, it exhibits inferior performance for most cases, as shown in Table 3. Also, observe that results of categorisation using SVM, as given in last row of Table 3, is not consistent over different kernels. For both SIFT and ST features, SVM gives the best results with Gaussian and ISD kernels. Fig. 2 shows the change in the criterion function, as given in the right hand side (RHS) of Eqn. (14), with increasing number of eigenvectors. A global maxima is considered as the optimal (reduced) number of eigenvectors/dimensions for the transformation of source domain data.

4.3 Text categorisation

Performance of the proposed method of DA was observed on the pre-processed Reuters-21578 dataset [18]. This popular dataset contains five top categories and many sub-categories. Data from different sub-categories under the same parent category are considered to be from different but related domains. By this way, three datasets are obtained, namely 'orgs vs. people', 'orgs vs. places' and 'people vs. places', as used by Dai *et al.* [18]. The number of instances in each of the source and target domains for the three datasets are 1109 (average) and the number of features is 4563. We have used linear kernel to evaluate our method. We compare our work with topic correlation analysis (ToCA) [19] and spectral feature alignment (SFA) [20]. Table 4 shows the classification accuracies using various techniques of DA using SVM classifier. A Gaussian kernel is used for our proposed EDA-K, where the mean of the dataset (over all dimensions) is taken as the sigma of the kernel function. We considered 20% of the data from target domain as the training samples to estimate the transformed source domain. Since it is easier to consider large

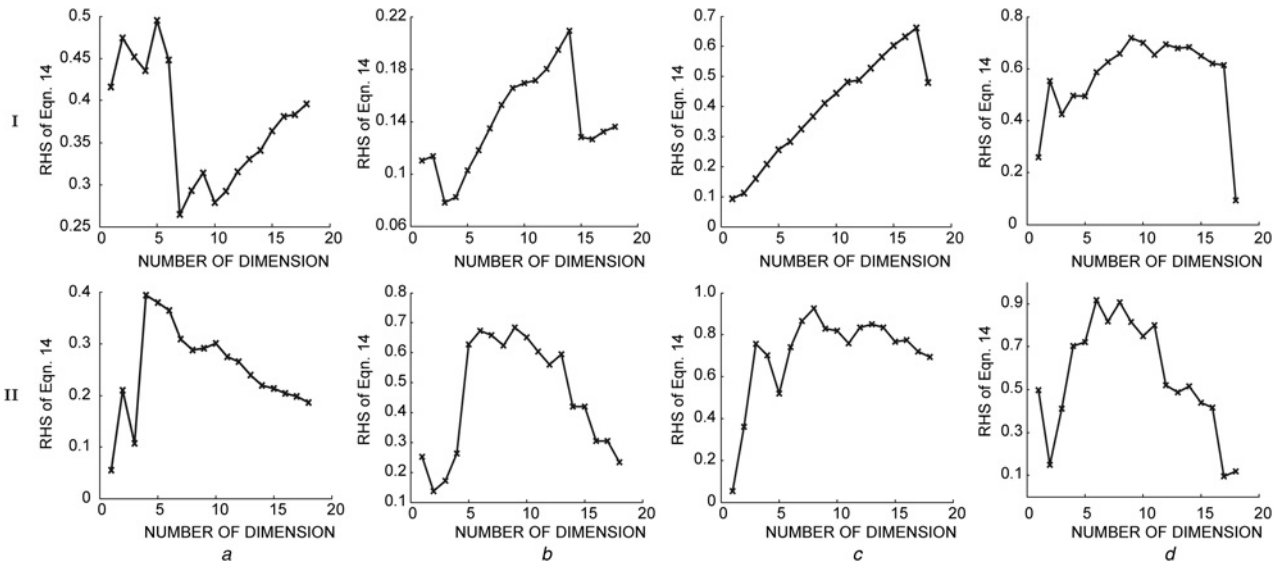

Fig. 2 *Two sets of plots of criterion function (RHS of (14)) with increasing dimension, using features – (i) SIFT, in top row and (ii) ST, in bottom row, for four different kernels*

a Gaussian
b Laplacian
c ISD
d ID
Global peak gives the optimal value of p^*

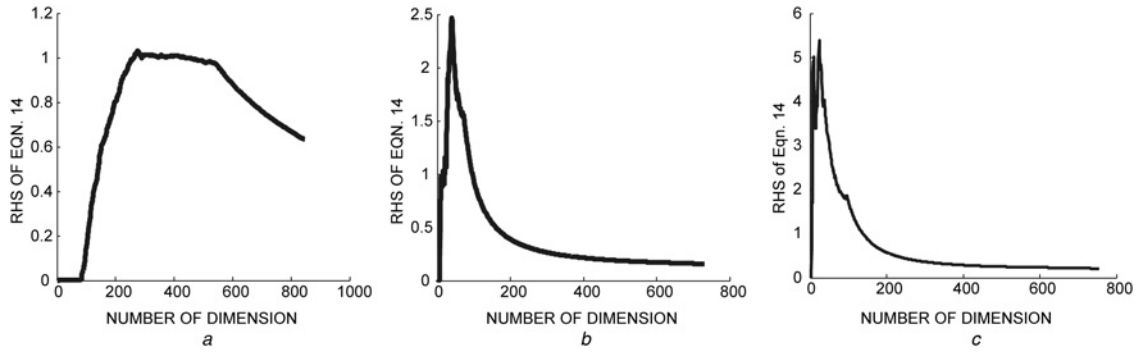

Fig. 3 *Plots of the criterion function (RHS in (14)) with the increasing dimension for*

a Orgs people,
b Orgs places
c People places, in Reuters-21578 [18] dataset
Global peak gives the optimal value of p^*

Table 4 Classification accuracies (in %-age) for text categorisation using Reuters-21578 dataset [18], using different techniques of DA (best results highlighted in bold)

Method	Orgs vs. people	Orgs vs. places	People vs. places	Average
ToCA [19]	79.2	73.0	62.6	70.5
SFA [20]	67.1	68.3	50.6	62
NA	67.0	66.9	52.0	62
EDA-K	**85.3**	**82.7**	**81.4**	**83.1**

dimensional dataset using kernel Gram matrices, we perform transformation of source domain data only in RKHS in this case. Fig. 3 shows the change in the criterion function (see (14)) with increasing number of eigenvectors, to obtain the optimal value of p^*.

Results from all experimentations reveal that the proposed method of DA is robust and can be applied to a wide range of datasets. Selecting the optimal number of dimensions and estimating the transformed source domain in RKHS significantly improves the performance. Using reduced number of eigenvectors (dimension) for estimating the transformation, also improves the computational cost.

5 Conclusions

In this paper, we propose an efficient technique of unsupervised DA using the eigenvectors and eigenvalues of source and target domains. We extend the concept from feature space to the kernel space, to deal with non-linear transformation of data. Experimental results on real-world image, video and text datasets show that the proposed method gives better classification accuracy in most of the cases, when compared with recent state-of-art works on DA.

6 References

1 Pan, S.J., Yang, Q.: 'A survey on transfer learning', *IEEE Trans. Knowl. Data Eng.*, 2010, **22**, pp. 1345–1359
2 Beijbom, O.: 'Domain adaptation for computer vision applications'. Technical report, University of California, San Diego, June 2012
3 Pan, S.J., Tsang, I.W., Kwok, J.T., Yang, Q.: 'Domain adaptation via transfer component analysis', *IEEE Trans. Neural Netw.*, 2011, **22**, (2), pp. 199–210
4 Gretton, A., Smola, A., Huang, J., Schmittfull, M., Borgwardt, K., Schölkopf, B.: 'Covariate shift by kernel mean matching', in Quiñonero-Candela, J., Sugiyama, M., Schwaighofer, A., Lawrence, N., (Eds.): 'Dataset shift in machine learning', (The MIT Press, 2009), Ch. 8, pp. 131–160
5 Howard, A., Jebara, T.: 'Transformation learning via kernel alignment'. Int. Conf. on Machine Learning and Applications, 2009, pp. 301–308
6 Gopalan, R., Li, R., Chellappa, R.: 'Domain adaptation for object recognition: an unsupervised approach'. IEEE Int. Conf. on Computer Vision, 2011, pp. 999–1006
7 Gong, B., Shi, Y., Sha, F., Grauman, K.: 'Geodesic flow Kernel for unsupervised domain adaptation'. IEEE Conf. on Computer Vision and Pattern Recognition, 2012, pp. 2066–2073
8 Samanta, S., Das, S.: 'Modeling sequential domain shift through estimation of subspaces for categorization'. British Machine Vision Conf., 2014
9 Fernando, B., Habrard, A., Sebban, M., Tuytelaars, T.: 'Unsupervised visual domain adaptation using subspace alignment'. IEEE Int. Conf. in Computer Vision, 2013
10 Saenko, K., Kulis, B., Fritz, M., Darrell, T.: 'Adapting visual category models to new domains'. European Conf. on Computer Vision, 2010, pp. 213–226
11 Samanta, S., Das, S.: 'Domain adaptation based on eigen-analysis and clustering, for object categorization'. Int. Conf. on Computer Analysis of Images and Patterns, 2013, pp. 245–253
12 Hamm, J., Lee, D.D.: 'Grassmann discriminant analysis: a unifying view on subspace-based learning'. Int. Conf. on Machine Learning, 2008, pp. 376–383
13 Shawe-Taylor, J., Cristianini, N.: 'Kernel methods for pattern analysis' (Cambridge University Press, New York, NY, USA, 2004)
14 Shawe-Taylor, J., Williams, C.K.I.: 'The stability of kernel principal components analysis and its relation to the process eigenspectrum'. In: Becker, S., Thrun, S., Obermayer, K., (Eds.): 'Advances in neural information processing systems 15 Neural Information Processing Systems', (NIPS, MIT Press, Vancouver, British Columbia, Canada, 9-14 December 2002), pp. 367–374
15 Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: 'Speeded-up robust features (SURF)', *Comput. Vis. Image Underst.*, 2008, **110**, (3), pp. 346–359
16 Duan, L., Xu, D., Tsang, I.W., Luo, J.: 'Visual event recognition in videos by learning from web data', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2012, **34**, (9), pp. 1667–1680
17 Duan, L., Xu, D., Chang, S.-F.: 'Exploiting web images for event recognition in consumer videos: a multiple source domain adaptation approach'. IEEE Conf. on Computer Vision and Pattern Recognition, 2012, pp. 1338–1345
18 Dai, W., Yang, Q., Xue, G.-R., Yu, Y.: 'Boosting for transfer learning'. Int. Conf. on Machine Learning, 2007, pp. 193–200
19 Li, L., Jin, X., Long, M.: 'Topic correlation analysis for cross-domain text classification'. AAAI Conf. on Artificial Intelligence, 2012, pp. 1–7
20 Pan, S.J., Ni, X., Sun, J.-T., Yang, Q., Chen, Z.: 'Cross-domain sentiment classification via spectral feature alignment'. Int. Conf. on World Wide Web, 2010, pp. 751–760