# Nonlinear Embedding Transform for Unsupervised Domain Adaptation

Hemanth Venkateswara, Shayok Chakraborty, Sethuraman Panchanathan

Center for Cognitive Ubiquitous Computing
Arizona State University, AZ, USA
{hemanthv,schakr10,panch}@asu.edu

**Abstract.** The problem of domain adaptation (DA) deals with adapting classifier models trained on one data distribution to different data distributions. In this paper, we introduce the Nonlinear Embedding Transform (NET) for unsupervised DA by combining domain alignment along with similarity-based embedding. We also introduce a validation procedure to estimate the model parameters for the NET algorithm using the source data. Comprehensive evaluations on multiple vision datasets demonstrate that the NET algorithm outperforms existing competitive procedures for unsupervised DA.

**Keywords:** Unsupervised, MMD, Nonlinear Embedding, Validation

## 1   Introduction

Classification models trained on labeled datasets are ineffective over data from different distributions owing to data-shift [14]. The problem of domain adaptation (DA) deals with adapting models trained on one data distribution (source domain) to different data distributions (target domains). For the purpose of this paper, we organize unsupervised DA procedures under two categories: *linear* and *nonlinear*, based on feature representations used in the model. *Linear* techniques determine linear transformations of the source (target) data and align it with the target (source), or learn a linear classifier with the source data and adapt it to the target data [2], [13] and [15]. *Nonlinear* procedures on the other hand, apply nonlinear transformations to reduce cross-domain disparity [6], [11].

In this work we present the Nonlinear Embedding Transform (NET) procedure for unsupervised DA. The NET consists of two steps, (i) Nonlinear domain alignment using Maximum Mean Discrepancy (MMD) [9], (ii) similarity-based embedding to cluster the data for enhanced classification. In addition, we introduce a procedure to sample source data in order to generate a validation set for model selection. We study the performance of the NET algorithm with popular DA datasets for computer vision. Our results showcase significant improvement in the classification accuracies compared to competitive DA procedures.

## 2   Related Work

In this section we provide a concise review of some of the unsupervised DA procedures closely related to the NET. Under *linear* methods, Burzzone et al. [2], proposed the DASVM algorithm to iteratively adapt a SVM trained on the source data, to the unlabeled target data. The state-of-the-art linear DA procedures are Subspace Alignment (SA), by Fernando et al. [5], and the CORAL algorithm, by Sun et al. [13]. The SA aligns the subspaces of the source and the target with a linear transformation and the CORAL transforms the source data such that the covariance matrices of the source and target are aligned.

*Nonlinear* procedures generally project the data to a high-dimensional space and align the source and target distributions in that space. The popular GFK algorithm by Gong et al. [6], projects the two distributions onto a manifold and learns a transformation to align them. The Transfer Component Analysis (TCA) [11], Transfer Joint Matching (TJM) [8], and Joint Distribution Adaptation (JDA) [9], algorithms, apply MMD-based projection to nonlinearly align the domains. In addition, the TJM implements instance selection using $\ell_{2,1}$-norm regularization and the JDA performs a joint distribution alignment of the source and target domains. The NET implements nonlinear alignment of the domains along with a similarity preserving projection, which ensures that the projected data is clustered based on category. We compare the NET with only kernel-based nonlinear methods and do not include deep learning based DA procedures.

## 3   DA With Nonlinear Embedding

In this section we outline the problem of unsupervised DA and develop the NET algorithm. Let $\mathbf{X}_S = [\mathbf{x}_1^s, \ldots, \mathbf{x}_{n_s}^s] \in \mathbb{R}^{d \times n_s}$ and $\mathbf{X}_T = [\mathbf{x}_1^t, \ldots, \mathbf{x}_{n_t}^t] \in \mathbb{R}^{d \times n_t}$ be the source and target data points respectively. Let $Y_S = [y_1^s, \ldots, y_{n_s}^s]$ and $Y_T = [y_1^t, \ldots, y_{n_t}^t]$ be the source and target labels respectively. Here, $\mathbf{x}_i^s$ and $\mathbf{x}_i^t \in \mathbb{R}^d$ are data points and $y_i^s$ and $y_i^t \in \{1, \ldots, C\}$ are the associated labels. We define $\mathbf{X} := [\mathbf{x}_1, \ldots, \mathbf{x}_n] = [\mathbf{X}_S, \mathbf{X}_T]$, where $n = n_s + n_t$. In the case of unsupervised DA, the labels $Y_T$ are missing and the joint distributions for the two domains are different, i.e. $P_S(X, Y) \neq P_T(X, Y)$. The task lies in learning a classifier $f(\mathbf{x}) = p(y|\mathbf{x})$, that predicts the labels of the target data points.

### 3.1   Nonlinear Embedding for DA

One of the techniques to reduce domain disparity is to project the source and target data to a common subspace. KPCA is a popular nonlinear projection algorithm where data is first mapped to a high-dimensional (possibly infinite-dimensional) space given by $\Phi(\mathbf{X}) = [\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_n)]$. $\phi : \mathbb{R}^d \to \mathcal{H}$ defines the mapping and $\mathcal{H}$ is a RKHS with a psd kernel $k(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x})^\top \phi(\mathbf{y})$. The kernel matrix for $\mathbf{X}$ is given by $\mathbf{K} = \Phi(\mathbf{X})^\top \Phi(\mathbf{X}) \in \mathbb{R}^{n \times n}$. The mapped data is then projected onto a subspace of eigen-vectors (directions of maximum nonlinear variance in the RKHS). The top $k$ eigen-vectors in the RKHS are obtained

using the represeter theorem, $\mathbf{U} = \mathit{\Phi}(\mathbf{X})\mathbf{A}$, where $\mathbf{A} \in \mathbb{R}^{n \times k}$ is the matrix of coefficients that needs to be determined. The nonlinearly projected data is then given by, $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_n] = \mathbf{A}^\top \mathbf{K} \in \mathbb{R}^{k \times n}$, where $\mathbf{z}_i \in \mathbb{R}^k$, $i = 1, \ldots, n$, are the projected data points.

In order to reduce the domain discrepancy in the projected space, we implement the joint distribution adaptation (JDA), as outlined in [9]. The JDA seeks to align the mariginal and conditional probability distributions of the projected data $(\mathbf{Z})$, by estimating the coefficient matrix $\mathbf{A}$, which minimizes:

$$\min_{\mathbf{A}} \sum_{c=0}^{C} \mathrm{tr}(\mathbf{A}^\top \mathbf{K} \mathbf{M}_c \mathbf{K}^\top \mathbf{A}). \tag{1}$$

$\mathrm{tr}(.)$ refers to trace and $\mathbf{M}_c$, where $c = 0, \ldots, C$, are $n \times n$ matrices given by,

$$(M_c)_{ij} = \begin{cases} \frac{1}{n_s^{(c)} n_s^{(c)}}, & \mathbf{x}_i, \mathbf{x}_j \in \mathcal{D}_s^{(c)} \\ \frac{1}{n_t^{(c)} n_t^{(c)}}, & \mathbf{x}_i, \mathbf{x}_j \in \mathcal{D}_t^{(c)} \\ \frac{-1}{n_s^{(c)} n_t^{(c)}}, & \begin{cases} \mathbf{x}_i \in \mathcal{D}_s^{(c)}, \mathbf{x}_j \in \mathcal{D}_t^{(c)} \\ \mathbf{x}_j \in \mathcal{D}_s^{(c)}, \mathbf{x}_i \in \mathcal{D}_t^{(c)} \end{cases} \\ 0, & \text{otherwise,} \end{cases} \qquad (M_0)_{ij} = \begin{cases} \frac{1}{n_s n_s}, & \mathbf{x}_i, \mathbf{x}_j \in \mathcal{D}_s \\ \frac{1}{n_t n_t}, & \mathbf{x}_i, \mathbf{x}_j \in \mathcal{D}_t \\ \frac{-1}{n_s n_t}, & \text{otherwise,} \end{cases} \tag{3}$$

$$\tag{2}$$

where $\mathcal{D}_s$ and $\mathcal{D}_t$ are the sets of source and target data points respectively. $\mathcal{D}_s^{(c)}$ is the set of source data points belonging to class $c$ and $n_s^{(c)} = |\mathcal{D}_s^{(c)}|$. Likewise, $\mathcal{D}_t^{(c)}$ is the set of target data points belonging to class $c$ and $n_t^{(c)} = |\mathcal{D}_t^{(c)}|$. Since the target labels are unknown, we use predicted labels for the target data points. We begin with predicting the target labels using a classifier trained on the source data and refine these labels over iterations, to arrive at the final prediction. For more details please refer to [9].

In addition to domain alignment, we would like the projected data $\mathbf{Z}$, to be classification friendly (easily classifiable). To this end, we introduce Laplacian eigenmaps to ensure a similarity-preserving projection such that data points with the same class label are clustered together. The similarity relations are captured by the $(n \times n)$ adjacency matrix $\mathbf{W}$, and the optimization problem estimates the projected data $\mathbf{Z}$;

$$\mathbf{W}_{ij} := \begin{cases} 1 & y_i^s = y_j^s \text{ or } i = j \\ 0 & y_i^s \neq y_j^s \text{ or labels unknown,} \end{cases} \qquad \min_{\mathbf{Z}} \frac{1}{2} \sum_{ij} \left\lVert \frac{\mathbf{z}_i}{\sqrt{d_i}} - \frac{\mathbf{z}_j}{\sqrt{d_j}} \right\rVert^2 \mathbf{W}_{ij} = \mathrm{tr}(\mathbf{Z}\mathbf{L}\mathbf{Z}^\top).$$
$$\qquad\qquad (4) \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (5)$$

$\mathbf{D} = diag(d_1, \ldots, d_n)$ is the $(n \times n)$ diagonal matrix where, $d_i = \sum_j \mathbf{W}_{ij}$ and $\mathbf{L}$ is the normalized graph laplacian matrix that is symmetric positive semidefinite and is given by $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{W} \mathbf{D}^{-1/2}$, where $\mathbf{I}$ is an identity matrix. When $\mathbf{W}_{ij} = 1$, the projected data points $\mathbf{z}_i$ and $\mathbf{z}_j$ are close together (as they belong to the same category). The normalized distance between the vectors $\lVert \mathbf{z}_i/\sqrt{d_i} - \mathbf{z}_j/\sqrt{d_j} \rVert^2$, captures a more robust measure of data point clustering compared to the un-normalized distance $\lVert \mathbf{z}_i - \mathbf{z}_j \rVert^2$, [4].

### 3.2  Optimization Problem

The optimization problem for NET is obtained from (1) and (5) by substituting, $\mathbf{Z} = \mathbf{A}^\top \mathbf{K}$. Along with regularization and a constraint, we get,

$$\min_{\mathbf{A}^\top \mathbf{KDK}^\top \mathbf{A} = \mathbf{I}} \alpha.\mathrm{tr}(\mathbf{A}^\top \mathbf{K} \sum_{c=0}^{C} \mathbf{M}_c \mathbf{K}^\top \mathbf{A}) + \beta.\mathrm{tr}(\mathbf{A}^\top \mathbf{KLK}^\top \mathbf{A}) + \gamma ||\mathbf{A}||_F^2. \qquad (6)$$

$\mathbf{A} \in \mathbb{R}^{n \times k}$ is the projection matrix. The regularization term $||\mathbf{A}||_F^2$ (Frobenius norm), controls the smoothness of projection and the magnitudes of $(\alpha, \beta, \gamma)$, denote the importance of the individual terms in (6). The constraint prevents the data points from collapsing onto a subspace of dimensionality less than $k$, [1]. Equation (6) can be solved by constructing the Lagrangian $L(\mathbf{A}, \mathbf{\Lambda})$, where, $\mathbf{\Lambda} = diag(\lambda_1, \ldots, \lambda_k)$, is the diagonal matrix of Lagrangian constants (see [8]). Setting the derivative $\frac{\partial L}{\partial \mathbf{A}} = 0$, yields the generalized eigen-value problem,

$$\left(\alpha \mathbf{K} \sum_{c=0}^{C} \mathbf{M}_c \mathbf{K}^\top + \beta \mathbf{KLK}^\top + \gamma \mathbf{I}\right) \mathbf{A} = \mathbf{KDK}^\top \mathbf{A} \mathbf{\Lambda}. \qquad (7)$$

$\mathbf{A}$ is the matrix of the $k$-smallest eigen-vectors of (7) and $\mathbf{\Lambda}$ is the diagonal matrix of eigen-values. The projected data points are given by, $\mathbf{Z} = \mathbf{A}^\top \mathbf{K}$.

### 3.3  Model Selection

Current DA methods use the target data to validate the optimum parameters for their models [8], [9]. We introduce a new technique to evaluate $(\alpha, \beta, \gamma, k)$, using a subset of the source data as a validation set. The subset is selected by weighting the source data points using Kernel Mean Matching (KMM). The KMM computes source instance weights $w_i$, by minimizing, $\left|\left| \frac{1}{n_s} \sum_{i=1}^{n_s} w_i \phi(\mathbf{x}_i^s) - \frac{1}{n_t} \sum_{j=1}^{n_t} \phi(\mathbf{x}_i^t) \right|\right|_{\mathcal{H}}^2$. Defining $\kappa_i := \frac{n_s}{n_t} \sum_{j=1}^{n_t} k(\mathbf{x}_i^s, \mathbf{x}_j^t)$, $i = 1, \ldots, n_s$ and $\mathbf{K}_{S_{ij}} = k(\mathbf{x}_i^s, \mathbf{x}_j^s)$, the minimization can be written in terms of quadratic programming:

$$\min_{\mathbf{w}} = \frac{1}{2} \mathbf{w}^\top \mathbf{K}_S \mathbf{w} - \kappa^\top \mathbf{w}, \quad \text{s.t. } w_i \in [0, B], \left| \sum_{i=1}^{n_s} w_i - n_s \right| \leq n_s \epsilon. \qquad (8)$$

The first constraint limits the scope of discrepancy between source and target distributions with $B \to 1$, leading to an unweighted solution. The second constraint ensures the measure $w(x)P_S(x)$, is a probability distribution [7]. In our experiments, the validation set is 30% of the source data with the largest weights. This validation set is used to estimate the best values for $(\alpha, \beta, \gamma, k)$.

## 4  Experiments

We compare the NET algorithm with the following baseline and state-of-the-art methods. NA (No Adaptation - classifier trained on the source and tested on the target), SA (Subspace Alignment [5]), CA (Correlation Alignment (CORAL) [13]), GFK (Geodesic Flow Kernel [6]), TCA (Transfer Component Analysis
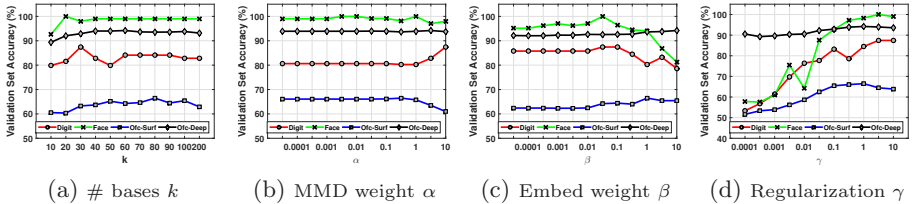
Fig. 1: Each figure depicts the accuracies over the validation set for a range of values. When studying a parameter (say $k$), the remaining parameters $(\alpha, \beta, \gamma)$ are fixed at the optimum value.

Table 1: Recognition accuracies (%) for DA experiments on the digit and face datasets. {MNIST(M), USPS(U), CKPlus(CK), MMI(MM). M→U implies M is source domain and U is target domain. The best and second best results are in **bold** and *italic*.

| Expt. | NA | SA | CA | GFK | TCA | JDA | NET$_v$ | NET$^*$ | Expt. | NA | SA | CA | GFK | TCA | JDA | NET$_v$ | NET$^*$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| M→ U | 65.94 | 67.39 | 59.33 | 66.06 | 60.17 | 67.28 | *72.72* | **75.39** | CK→ MM | 29.90 | 31.12 | 31.89 | 28.75 | 32.72 | 29.78 | **30.54** | *29.97* |
| U→ M | 44.70 | 51.85 | 50.80 | 47.40 | 39.85 | *59.65* | 61.35 | **62.60** | MM→ CK | 41.48 | 39.75 | 37.74 | 37.94 | 31.33 | 28.39 | *40.08* | **45.83** |
| **Avg.** | 55.32 | 59.62 | 55.07 | 56.73 | 50.01 | 63.46 | *67.04* | **68.99** | **Avg.** | 35.69 | 35.43 | 34.81 | 33.35 | 32.02 | 29.08 | *35.31* | **37.90** |

[11]), JDA (Joint Distribution Adaptation [9]). NET$_v$ is a special case of the NET algorithm where parameters $(\alpha, \beta, \gamma, k)$, have been estimated using (8) (see Sec. 3.3). For NET$^*$, the optimum values for $(\alpha, \beta, \gamma, k)$ are estimated using the target data for cross validation.

### 4.1 Datasets

***Office-Caltech* datasets**: This object recognition dataset [6], consists of images of everyday objects categorized into 4 domains; `Amazon`, `Caltech`, `Dslr` and `Webcam`. It has 10 categories of objects and a total of 2533 images. We experiment with two kinds of features (i) SURF features obtained from [6], (ii) Deep features. To extract deep features, we use an 'off-the-shelf' deep convolutional neural network (VGG-F model [3]). We use the 4096-dimensional features from the $fc8$ layer and apply PCA to reduce the feature dimension to 500.

***MNIST-USPS* datasets**: We use a subset of the popular handwritten digit (0-9) recognition datasets (2000 images from `MNIST` and 1800 images from `USPS` based on [8]). The images are resized to $16 \times 16$ pixels and represented as 256-dimensional vectors.

***CKPlus-MMI* datasets**: The CKPlus [10], and MMI [12], datasets consist of facial expression videos. From these videos, we select the frames with the most-intense expression to create the domains `CKPlus` and `MMI`, with around 1500 images each and 6 categories viz., *anger, disgust, fear, happy, sad, surprise.* We use a pre-trained deep neural network to extract features (see *Office-Caltech*).

### 4.2 Results and Discussion

For $k$, we explore optimum values in the set $\{10, 20, \ldots, 100, 200\}$. For $(\alpha, \beta, \gamma)$, we select from $\{0, 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5, 10\}$. For the

**Table 2:** Recognition accuracies (%) for DA experiments on the *Office-Caltech* dataset with SURF and Deep features. {Amazon(A), Webcam(W), Dslr(D), Caltech(C)}. A→W implies A is source and W is target. The best and second best results are in **bold** and *italic*.

| Expt. | SURF Features | | | | | | | | Deep Features | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | NA | SA | CA | GFK | TCA | JDA | NET$_v$ | NET* | NA | SA | CA | GFK | TCA | JDA | NET$_v$ | NET* |
| A→ C | 34.19 | 38.56 | 33.84 | 39.27 | 39.89 | 39.36 | *43.10* | **43.54** | **83.01** | 80.55 | *82.47* | 81.00 | 75.53 | **83.01** | 82.28 | **83.01** |
| A→ D | 35.67 | 37.58 | 36.94 | 34.40 | 33.76 | *39.49* | 36.31 | **40.76** | 84.08 | 82.17 | 87.90 | 82.80 | 82.17 | *89.81* | 80.89 | **91.08** |
| A→ W | 31.19 | 37.29 | 31.19 | *41.70* | 33.90 | 37.97 | 35.25 | **44.41** | 79.32 | 82.37 | 80.34 | 84.41 | 76.61 | 87.12 | *87.46* | **90.85** |
| C→ A | 36.01 | 43.11 | 36.33 | *45.72* | 44.47 | 44.78 | 46.24 | **46.45** | 90.70 | 88.82 | *91.12* | 90.60 | 89.13 | 90.07 | 90.70 | **92.48** |
| C→ D | 38.22 | 43.95 | 38.22 | 43.31 | 36.94 | *45.22* | 36.31 | **45.86** | 83.44 | 80.89 | 82.80 | 77.07 | 75.80 | 89.17 | *90.45* | **92.36** |
| C→ W | 29.15 | 36.27 | 29.49 | 35.59 | 32.88 | *41.69* | 33.56 | **44.41** | 76.61 | 77.29 | 79.32 | 78.64 | 78.31 | *85.76* | 84.07 | **90.85** |
| D→ A | 28.29 | 29.65 | 28.39 | 26.10 | 31.63 | 33.09 | *35.60* | **39.67** | 88.51 | 84.33 | 86.63 | 88.40 | 88.19 | 91.22 | *91.43* | **91.54** |
| D→ C | 29.56 | 31.88 | 29.56 | 30.45 | 30.99 | 31.52 | *34.11* | **35.71** | 77.53 | 76.26 | 75.98 | 78.63 | 74.43 | 80.09 | **83.38** | *82.10* |
| D→ W | 83.73 | 87.80 | 83.39 | 79.66 | 85.42 | *89.49* | **90.51** | 87.80 | *99.32* | 98.98 | 99.32 | 98.31 | 97.97 | 98.98 | **99.66** | **99.66** |
| W→ A | 31.63 | 32.36 | 31.42 | 27.77 | 29.44 | 32.78 | *39.46* | **41.65** | 82.34 | 84.01 | 82.76 | 88.61 | 86.21 | 91.43 | *91.95* | **92.58** |
| W→ C | 28.76 | 29.92 | 28.76 | 28.41 | 32.15 | 31.17 | *32.77* | **35.89** | 76.53 | 78.90 | 74.98 | 76.80 | 76.71 | **82.74** | 82.28 | *82.56* |
| W→ D | 84.71 | 90.45 | 85.35 | 82.17 | 85.35 | *89.17* | **91.72** | 89.81 | *99.36* | **100.00** | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | *99.36* |
| **Avg.** | 40.93 | 44.90 | 41.07 | 42.88 | 43.07 | *46.31* | 46.24 | **49.66** | 85.06 | 84.55 | 85.30 | 85.44 | 83.42 | *89.12* | 88.71 | **90.70** |

sake of brevity, we evaluate and present one set of parameters $(\alpha, \beta, \gamma, k)$, for all the DA experiments in a dataset. For all the experiments, we choose 10 iterations to converge to the predicted test/validation labels when estimating $\mathbf{M}_c$. Figure (1), depicts the variation in validation set accuracies for each of the parameters. We select the parameter value with the highest validation set accuracy as the optimal value in the NET$_v$ experiments.

For fair comparison with existing methods, we follow the same experimental protocol as in [6], [8]. We train a nearest neighbor (NN) classifier on the projected source data and test on the projected target data. Table (1), captures the results for the digit and face datasets. Table (2), outlines the results for the *Office-Caltech* dataset. The accuracies reflect the percentage of correctly classified target data points. The accuracies obtained with NET$_v$, demonstrate that the validation set generated from the source data is a good option for validating model parameters in unsupervised DA. The parameters for the NET* experiment are estimated using the target datset; $(\alpha = 1, \beta = 1, \gamma = 1, k = 20)$ for the object recognition datasets, $(\alpha = 1, \beta = 0.01, \gamma = 1, k = 20)$ for the digit dataset and $(\alpha = 0.01, \beta = 0.01, \gamma = 1, k = 20)$ for the face dataset. The accuracies obtained with the NET algorithm are consistently better than existing methods, demonstrating the role of nonlinear embedding along with domain alignment.

## 5   Conclusions and Acknowledgments

We have proposed the NET algorithm for unsupervised DA along with a procedure for generating a validation set for model selection using the source data. Both the validation procedure and NET have better recognition accuracies than competitive visual DA methods across multiple vision based datasets. This material is based upon work supported by the National Science Foundation (NSF) under Grant No:1116360. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

# References

1. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural computation 15(6), 1373–1396 (2003)
2. Bruzzone, L., Marconcini, M.: Domain adaptation problems: A dasvm classification technique and a circular validation strategy. IEEE, PAMI 32(5), 770–787 (2010)
3. Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the devil in the details: Delving deep into convolutional nets. In: BMVC (2014)
4. Chung, F.R.: Spectral graph theory, vol. 92. American Mathematical Soc. (1997)
5. Fernando, B., Habrard, A., Sebban, M., Tuytelaars, T.: Unsupervised visual domain adaptation using subspace alignment. In: CVPR. pp. 2960–2967 (2013)
6. Gong, B., Shi, Y., Sha, F., Grauman, K.: Geodesic flow kernel for unsupervised domain adaptation. In: IEEE CVPR (2012)
7. Gretton, A., Smola, A., Huang, J., Schmittfull, M., Borgwardt, K., Schölkopf, B.: Covariate shift by kernel mean matching. Dataset shift in machine learning 3(4), 5 (2009)
8. Long, M., Wang, J., Ding, G., Sun, J., Yu, P.: Transfer joint matching for unsupervised domain adaptation. In: CVPR. pp. 1410–1417 (2014)
9. Long, M., Wang, J., Ding, G., Sun, J., Yu, P.S.: Transfer feature learning with joint distribution adaptation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2200–2207 (2013)
10. Lucey, P., Cohn, J.F., Kanade, T., Saragih, J., Ambadar, Z., Matthews, I.: The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In: CVPR. pp. 94–101. IEEE (2010)
11. Pan, S.J., Tsang, I.W., Kwok, J.T., Yang, Q.: Domain adaptation via transfer component analysis. Neural Networks, IEEE Trans. on 22(2), 199–210 (2011)
12. Pantic, M., Valstar, M., Rademaker, R., Maat, L.: Web-based database for facial expression analysis. In: ICME. IEEE (2005)
13. Sun, B., Feng, J., Saenko, K.: Return of frustratingly easy domain adaptation. In: ICCV, TASK-CV (2015)
14. Torralba, A., Efros, A.A.: Unbiased look at dataset bias. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. pp. 1521–1528. IEEE (2011)
15. Venkateswara, H., Lade, P., Ye, J., Panchanathan, S.: Coupled support vector machines for supervised domain adaptation. In: ACM MM. pp. 1295–1298 (2015)