# Transfer Feature Learning with Joint Distribution Adaptation

Mingsheng Long[†‡], Jianmin Wang[†], Guiguang Ding[†], Jiaguang Sun[†], and Philip S. Yu[§]
[†]School of Software, TNLIST, Tsinghua University, Beijing, China
[‡]Department of Computer Science, Tsinghua University, Beijing, China
[§]Department of Computer Science, University of Illinois at Chicago, IL, USA
longmingsheng@gmail.com, {jimwang,dinggg,sunjg}@tsinghua.edu.cn, psyu@uic.edu

## Abstract

*Transfer learning is established as an effective technology in computer vision for leveraging rich labeled data in the source domain to build an accurate classifier for the target domain. However, most prior methods have not simultaneously reduced the difference in both the marginal distribution and conditional distribution between domains. In this paper, we put forward a novel transfer learning approach, referred to as Joint Distribution Adaptation (JDA). Specifically, JDA aims to jointly adapt both the marginal distribution and conditional distribution in a principled dimensionality reduction procedure, and construct new feature representation that is effective and robust for substantial distribution difference. Extensive experiments verify that JDA can significantly outperform several state-of-the-art methods on four types of cross-domain image classification problems.*

## 1. Introduction

In computer vision, labeled information is crucial for a variety of recognition problems. For highly-evolving visual domains where labeled data are very sparse, one may expect to leverage abundant labeled data readily available in some related source domains for training accurate classifiers to be reused in the target domain. Recently, the literature has witnessed an increasing interest in developing *transfer learning* [16] algorithms for cross-domain knowledge adaptation problems. Transfer learning has proven to be promising in image classification [24, 12] and tagging [19, 25], object recognition [14, 2, 7, 10], and feature learning [13, 11, 17].

In cross-domain problems, the source and target data are usually sampled from *different* probability distributions. Therefore, a major computational issue of transfer learning is to reduce the distribution difference between domains. Recent works aim to discover a *shared* feature representation which can reduce the distribution difference and preserve the important properties of input data simultaneously

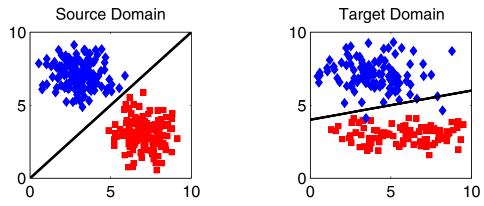


Figure 1. In our problem, *labeled* source and *unlabeled* target domains are different in *both* marginal and conditional distributions.

[21, 15, 18], or *re-weight* source data in order to minimize the distribution difference and then learn a classifier on the re-weighted source data [3, 4]. Most of existing methods measure the distribution difference based on either marginal distribution or conditional distribution. However, Figure 1 demonstrates the importance of matching *both* marginal and conditional distributions for robust transfer learning. Some very recent works started to match both the marginal and conditional distributions using sample selection [26], kernel density estimation [18], or two-stage re-weighting [23], but they may require either some labeled data in the target domain, or multiple source domains for consensus learning.

In this paper, we address a challenging scenario in which the source and target domains are different in *both* marginal and conditional distributions, and the target domain has *no* labeled data. We put forward a novel transfer learning solution, referred to as *Joint Distribution Adaptation* (JDA), to jointly adapt both the marginal and conditional distributions in a principled dimensionality reduction procedure. Specifically, we extend the nonparametric *Maximum Mean Discrepancy* (MMD) [8] to measure the difference in both marginal and conditional distributions, and integrate it with Principal Component Analysis (PCA) to construct feature representation that is effective and robust for substantial distribution difference. We present the underlying assumption and learning algorithm for the JDA optimization problem.

We perform comprehensive experiments on four types of real-world datasets: digit (USPS, MNIST), face (PIE), and object (COIL20, Office+Caltech [20]). From these datasets, we construct 36 *cross-domain* image datasets, each under a different difficulty in knowledge adaptation. Our empirical results demonstrate a significant improvement of **7.57%** in terms of classification accuracy, obtained by the proposed JDA approach over several state-of-the-art transfer learning methods. Our results reveal substantial effects of matching both marginal and conditional distributions across domains.

## 2. Related Work

In this section, we discuss prior works on transfer learning that are related to ours, and highlight their differences.

According to the literature survey [16], existing transfer learning methods can be roughly organized into two categories: *instance reweighting* [3, 4] and *feature extraction*. Our work belongs to the feature extraction category, which can be further reorganized into two rough subcategories.

1) *Property preservation*, which shares latent factors across domains by preserving important properties of data, *e.g*. statistical property [17, 11], geometric structure [19, 6].

2) *Distribution adaptation*, which explicitly minimizes predefined distance measures to reduce the difference in the marginal distribution [22, 15], conditional distribution [21], or both [26, 23, 18]. However, to match conditional distributions, these methods require either some labeled target data, or multiple source domains for consensus learning.

To our knowledge, our work is among the first attempts to jointly adapt *both* marginal and conditional distributions between domains, and *no* labeled data are required in the target domain. Our work is a principled dimensionality reduction procedure with MMD-based distribution matching, which is different from feature re-weighting methods [1, 5].

## 3. Joint Distribution Adaptation

In this section, we present in detail the Joint Distribution Adaptation (JDA) approach for effective transfer learning.

### 3.1. Problem Definition

We begin with the definitions of terminologies. For clarity, the frequently used notations are summarized in Table 1.

**Definition 1 (Domain)** *A domain $\mathcal{D}$ is composed of an $m$-dimensional feature space $\mathcal{X}$ and a marginal probability distribution $P(\mathbf{x})$,* i.e.*, $\mathcal{D} = \{\mathcal{X}, P(\mathbf{x})\}$, where $\mathbf{x} \in \mathcal{X}$.*

**Definition 2 (Task)** *Given domain $\mathcal{D}$, a task $\mathcal{T}$ is composed of a $C$-cardinality label set $\mathcal{Y}$ and a classifier $f(\mathbf{x})$,* i.e.*, $\mathcal{T} = \{\mathcal{Y}, f(\mathbf{x})\}$, where $y \in \mathcal{Y}$, and $f(\mathbf{x}) = Q(y|\mathbf{x})$ can be interpreted as the conditional probability distribution.*

**Problem 1 (Joint Distribution Adaptation)** *Given labeled source domain $\mathcal{D}_s = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{n_s}, y_{n_s})\}$ and unlabeled target domain $\mathcal{D}_t = \{\mathbf{x}_{n_s+1}, \ldots, \mathbf{x}_{n_s+n_t}\}$ under the assumptions that $\mathcal{X}_s = \mathcal{X}_t$, $\mathcal{Y}_s = \mathcal{Y}_t$, $P_s(\mathbf{x}_s) \neq P_t(\mathbf{x}_t)$, $Q_s(y_s|\mathbf{x}_s) \neq Q_t(y_t|\mathbf{x}_t)$, learn a feature representation in which the distribution differences between 1) $P_s(\mathbf{x}_s)$ and $P_t(\mathbf{x}_t)$, 2) $Q_s(y_s|\mathbf{x}_s)$ and $Q_t(y_t|\mathbf{x}_t)$ are explicitly reduced.*

Table 1. Notations and descriptions used in this paper.

| Notation | Description | Notation | Description |
|---|---|---|---|
| $\mathcal{D}_s, \mathcal{D}_t$ | source/target domain | $\mathbf{X}$ | input data matrix |
| $n_s, n_t$ | #source/target examples | $\mathbf{A}$ | adaptation matrix |
| $m, C$ | #shared features/classes | $\mathbf{Z}$ | embedding matrix |
| $k$ | #subspace bases | $\mathbf{H}$ | centering matrix |
| $\lambda$ | regularization parameter | $\mathbf{M}_c$ | MMD matrices, $c \in \{0, \ldots, C\}$ |

### 3.2. Proposed Approach

In this paper, we propose to adapt the joint distributions by a feature transformation $T$ so that the joint expectations of the features $\mathbf{x}$ and labels $y$ are matched between domains:

$$\begin{aligned}\min_T & \left\|\mathbb{E}_{P(\mathbf{x}_s,y_s)}\left[T\left(\mathbf{x}_s\right), y_s\right] - \mathbb{E}_{P(\mathbf{x}_t,y_t)}\left[T\left(\mathbf{x}_t\right), y_t\right]\right\|^2 \\ & \approx \left\|\mathbb{E}_{P_s(\mathbf{x}_s)}\left[T\left(\mathbf{x}_s\right)\right] - \mathbb{E}_{P_t(\mathbf{x}_t)}\left[T\left(\mathbf{x}_t\right)\right]\right\|^2 \\ & + \left\|\mathbb{E}_{Q_s(y_s|\mathbf{x}_s)}\left[y_s|T\left(\mathbf{x}_s\right)\right] - \mathbb{E}_{Q_t(y_t|\mathbf{x}_t)}\left[y_t|T\left(\mathbf{x}_t\right)\right]\right\|^2\end{aligned} \quad (1)$$

The problem is nontrivial, since there are no labeled data in the target domain, and $Q_t(y_t|\mathbf{x}_t)$ cannot be estimated exactly. The best approximation is to assume that $Q_t(y_t|\mathbf{x}_t) \approx Q_s(y_t|\mathbf{x}_t)$ [1]. This can be executed by applying a classifier $f$ trained on the labeled source data to the unlabeled target data. In order to achieve a more accurate approximation for $Q_t(y_t|\mathbf{x}_t)$, we propose an iterative pseudo label refinement strategy to iteratively refine the transformation $T$ and classifier $f$. The proposed approach is technically detailed later.

#### 3.2.1 Feature Transformation

Dimensionality reduction methods can learn a transformed feature representation by minimizing the reconstruction error of the input data. For simplicity and generality, we will choose Principal Component Analysis (PCA) for data reconstruction. Denote $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{m \times n}$ the input data matrix, and $\mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbf{1}$ the centering matrix, where $n = n_s + n_t$ and $\mathbf{1}$ the $n \times n$ matrix of ones, then the covariance matrix can be computed as $\mathbf{XHX}^{\mathrm{T}}$. The learning goal of PCA is to find an orthogonal transformation matrix $\mathbf{A} \in \mathbb{R}^{m \times k}$ such that embedded data variance is maximized

$$\max_{\mathbf{A}^{\mathrm{T}}\mathbf{A}=\mathbf{I}} \operatorname{tr}\left(\mathbf{A}^{\mathrm{T}}\mathbf{XHX}^{\mathrm{T}}\mathbf{A}\right) \quad (2)$$

where $\operatorname{tr}(\cdot)$ denotes the trace of a matrix. This optimization problem can be efficiently solved by eigendecomposition $\mathbf{XHX}^{\mathrm{T}}\mathbf{A} = \mathbf{A\Phi}$, where $\mathbf{\Phi} = \operatorname{diag}(\phi_1, \ldots, \phi_k) \in \mathbb{R}^{k \times k}$ are the $k$ largest eigenvalues. Then we find the optimal $k$-dimensional representation by $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_n] = \mathbf{A}^{\mathrm{T}}\mathbf{X}$.

#### 3.2.2 Marginal Distribution Adaptation

However, even through the PCA-induced $k$-dimensional representation, the distribution difference between domains will still be significantly large. Thus a major computational issue is to reduce the distribution difference by explicitly minimizing proper distance measures. Since parametrically estimating the probability density for a distribution is often a nontrivial problem, we resort to explore the sufficient statistics instead. To reduce the difference between *marginal* distributions $P_s(\mathbf{x}_s)$ and $P_t(\mathbf{x}_t)$, we follow [8, 15, 23] and adopt the empirical *Maximum Mean Discrepancy* (MMD) as the distance measure to compare different distributions, which computes the distance between the sample means of the source and target data in the $k$-dimensional embeddings:

$$\left\| \frac{1}{n_s}\sum_{i=1}^{n_s} \mathbf{A}^{\mathrm{T}}\mathbf{x}_i - \frac{1}{n_t}\sum_{j=n_s+1}^{n_s+n_t} \mathbf{A}^{\mathrm{T}}\mathbf{x}_j \right\|^2 = \mathrm{tr}\left(\mathbf{A}^{\mathrm{T}}\mathbf{X}\mathbf{M}_0\mathbf{X}^{\mathrm{T}}\mathbf{A}\right) \tag{3}$$

where $\mathbf{M}_0$ is the MMD matrix and is computed as follows

$$(M_0)_{ij} = \begin{cases} \frac{1}{n_s n_s}, & \mathbf{x}_i, \mathbf{x}_j \in \mathcal{D}_s \\ \frac{1}{n_t n_t}, & \mathbf{x}_i, \mathbf{x}_j \in \mathcal{D}_t \\ \frac{-1}{n_s n_t}, & \text{otherwise} \end{cases} \tag{4}$$

By minimizing Equation (3) such that Equation (2) is maximized, the marginal distributions between domains are drawn close under the new representation $\mathbf{Z} = \mathbf{A}^{\mathrm{T}}\mathbf{X}$. Note that we have just developed JDA to be similar to TCA [15].

### 3.2.3 Conditional Distribution Adaptation

However, reducing the difference in the marginal distributions does not guarantee that the conditional distributions between domains can also be drawn close. Indeed, minimizing the difference between the *conditional* distributions $Q_s(y_s|\mathbf{x}_s)$ and $Q_t(y_t|\mathbf{x}_t)$ is crucial for robust distribution adaptation [23]. Unfortunately, it is nontrivial to match the conditional distributions, even by exploring sufficient statistics of the distributions, since there are no labeled data in the target domain, *i.e.*, $Q_t(y_t|\mathbf{x}_t)$ cannot be modeled directly. Some very recent works started to match the conditional distributions via sample selection in a kernel mapping space [26], circular validation [3], co-training [4], and kernel density estimation [18]. But they all require some labeled data in the target domain, and thus cannot address our problem.

In this paper, we propose to explore the *pseudo* labels of the target data, which can be easily predicted by applying some base classifiers trained on the labeled source data to the unlabeled target data. The base classifier can be either standard learners, *e.g.*, Support Vector Machine (SVM), or transfer learners, *e.g.*, Transfer Component Analysis (TCA) [15]. Since the posterior probabilities $Q_s(y_s|\mathbf{x}_s)$ and $Q_t(y_t|\mathbf{x}_t)$ are quite involved, we resort to explore the sufficient statistics of class-conditional distributions $Q_s(\mathbf{x}_s|y_s)$ and $Q_t(\mathbf{x}_t|y_t)$ instead. Now with the true source labels and pseudo target labels, we can essentially match the class-conditional distributions $Q_s(\mathbf{x}_s|y_s = c)$ and $Q_t(\mathbf{x}_t|y_t = c)$ w.r.t. each class $c \in \{1, \ldots, C\}$ in the label set $\mathcal{Y}$. Here we modify MMD to measure the distance between the class-conditional distributions $Q_s(\mathbf{x}_s|y_s = c)$ and $Q_t(\mathbf{x}_t|y_t = c)$

$$\left\| \frac{1}{n_s^{(c)}} \sum_{\mathbf{x}_i \in \mathcal{D}_s^{(c)}} \mathbf{A}^{\mathrm{T}}\mathbf{x}_i - \frac{1}{n_t^{(c)}} \sum_{\mathbf{x}_j \in \mathcal{D}_t^{(c)}} \mathbf{A}^{\mathrm{T}}\mathbf{x}_j \right\|^2 = \mathrm{tr}\left(\mathbf{A}^{\mathrm{T}}\mathbf{X}\mathbf{M}_c\mathbf{X}^{\mathrm{T}}\mathbf{A}\right) \tag{5}$$

where $\mathcal{D}_s^{(c)} = \{\mathbf{x}_i : \mathbf{x}_i \in \mathcal{D}_s \wedge y(\mathbf{x}_i) = c\}$ is the set of examples belonging to class $c$ in the source data, $y(\mathbf{x}_i)$ is the true label of $\mathbf{x}_i$, and $n_s^{(c)} = |\mathcal{D}_s^{(c)}|$. Correspondingly, $\mathcal{D}_t^{(c)} = \{\mathbf{x}_j : \mathbf{x}_j \in \mathcal{D}_t \wedge \widehat{y}(\mathbf{x}_j) = c\}$ is the set of examples belonging to class $c$ in the target data, $\widehat{y}(\mathbf{x}_j)$ is the pseudo (predicted) label of $\mathbf{x}_j$, and $n_t^{(c)} = |\mathcal{D}_t^{(c)}|$. Thus the MMD matrices $\mathbf{M}_c$ involving class labels are computed as follows

$$(M_c)_{ij} = \begin{cases} \frac{1}{n_s^{(c)} n_s^{(c)}}, & \mathbf{x}_i, \mathbf{x}_j \in \mathcal{D}_s^{(c)} \\ \frac{1}{n_t^{(c)} n_t^{(c)}}, & \mathbf{x}_i, \mathbf{x}_j \in \mathcal{D}_t^{(c)} \\ \frac{-1}{n_s^{(c)} n_t^{(c)}}, & \begin{cases} \mathbf{x}_i \in \mathcal{D}_s^{(c)}, \mathbf{x}_j \in \mathcal{D}_t^{(c)} \\ \mathbf{x}_j \in \mathcal{D}_s^{(c)}, \mathbf{x}_i \in \mathcal{D}_t^{(c)} \end{cases} \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

By minimizing Equation (5) such that Equation (2) is maximized, the conditional distributions between domains are drawn close under the new representation $\mathbf{Z} = \mathbf{A}^{\mathrm{T}}\mathbf{X}$. With this important improvement, JDA can be robust for cross-domain problems with changes in conditional distributions.

It is important to note that, although many of the pseudo target labels are incorrect due to the differences in both the marginal and conditional distributions, we can still leverage them to match the conditional distributions with the revised MMD measure defined in Equation (5). The justification is that we match the distributions by exploring the sufficient statistics instead of the density estimates. In this way, we can leverage the source classifier to improve the target classifier. We will verify this argument thoroughly in the experiments.

### 3.2.4 Optimization Problem

In JDA, to achieve effective and robust transfer learning, we aim to simultaneously minimize the differences in both the marginal distributions and conditional distributions across domains. Thus, we incorporate Equations (3) and (5) into Equation (2), which leads to the JDA optimization problem:

$$\min_{\mathbf{A}^{\mathrm{T}}\mathbf{X}\mathbf{H}\mathbf{X}^{\mathrm{T}}\mathbf{A}=\mathbf{I}} \sum_{c=0}^{C} \mathrm{tr}\left(\mathbf{A}^{\mathrm{T}}\mathbf{X}\mathbf{M}_c\mathbf{X}^{\mathrm{T}}\mathbf{A}\right) + \lambda \|\mathbf{A}\|_F^2 \tag{7}$$

where $\lambda$ is the regularization parameter to guarantee the optimization problem to be well-defined. Based on the generalized Rayleigh quotient, minimizing Equations (3) and (5) such that Equation (2) is maximized is equivalent to minimizing Equations (3) and (5) such that Equation (2) is fixed.

It is obvious that TCA can be viewed as a special case of JDA with $C = 0$. With JDA, we can simultaneously adapt both the marginal distributions and conditional distributions between domains to facilitate joint distribution adaptation. A fascinating property of JDA is its capability to effectively explore the conditional distributions only using principled unsupervised dimensionality reduction and base classifier. Thus JDA can be easy to implement and deploy practically.

**Kernelization:** For nonlinear problems, consider kernel mapping $\psi : \mathbf{x} \mapsto \psi(\mathbf{x})$, or $\psi(\mathbf{X}) = [\psi(\mathbf{x}_1), \ldots, \psi(\mathbf{x}_n)]$, and kernel matrix $\mathbf{K} = \psi(\mathbf{X})^{\mathrm{T}}\psi(\mathbf{X}) \in \mathbb{R}^{n \times n}$. We utilize the Representer theorem to formulate Kernel-JDA as

$$\min_{\mathbf{A}^{\mathrm{T}}\mathbf{KHK}^{\mathrm{T}}\mathbf{A}=\mathbf{I}} \sum\nolimits_{c=0}^{C} \operatorname{tr}\left(\mathbf{A}^{\mathrm{T}}\mathbf{K}\mathbf{M}_c\mathbf{K}^{\mathrm{T}}\mathbf{A}\right) + \lambda \left\|\mathbf{A}\right\|_F^2 \quad (8)$$

where $\mathbf{A} \in \mathbb{R}^{n \times k}$ is the adaptation matrix for Kernel-JDA.

### 3.2.5 Iterative Refinement

It is worth noting that, with JDA, we can usually obtain a more accurate labeling for the target data. Thus, if we use this labeling as the pseudo target labels and run JDA iteratively, then we can alternatingly improve the labeling quality until convergence. This EM-like pseudo label refinement procedure is empirically effective as shown in experiments.

## 3.3. Learning Algorithm

According to the constrained optimization theory, we denote $\mathbf{\Phi} = \operatorname{diag}(\phi_1, \ldots, \phi_k) \in \mathbb{R}^{k \times k}$ as the Lagrange multiplier, and derive the Lagrange function for problem (7) as

$$\begin{aligned} L = &\operatorname{tr}\left(\mathbf{A}^{\mathrm{T}}\left(\mathbf{X}\sum\nolimits_{c=0}^{C}\mathbf{M}_c\mathbf{X}^{\mathrm{T}} + \lambda\mathbf{I}\right)\mathbf{A}\right) \\ &+ \operatorname{tr}\left(\left(\mathbf{I} - \mathbf{A}^{\mathrm{T}}\mathbf{XHX}^{\mathrm{T}}\mathbf{A}\right)\mathbf{\Phi}\right) \end{aligned} \quad (9)$$

Setting $\frac{\partial L}{\partial \mathbf{A}} = \mathbf{0}$, we obtain generalized eigendecomposition

$$\left(\mathbf{X}\sum\nolimits_{c=0}^{C}\mathbf{M}_c\mathbf{X}^{\mathrm{T}} + \lambda\mathbf{I}\right)\mathbf{A} = \mathbf{XHX}^{\mathrm{T}}\mathbf{A}\mathbf{\Phi} \quad (10)$$

Finally, finding the optimal adaptation matrix $\mathbf{A}$ is reduced to solving Equation (10) for the $k$ smallest eigenvectors. A complete procedure of JDA is summarized in Algorithm 1.

## 3.4. Computational Complexity

We analyze the computational complexity of Algorithm 1 using the big $O$ notation. We denote $T$ the number of iterations, then typical values of $k$ are not greater than 500, $T$ not greater than 50, so $k \ll \min(m, n)$, $T \ll \min(m, n)$. The computational cost is detailed as follows: $O\left(Tkm^2\right)$ for solving the generalized eigendecomposition problem with dense matrices, *i.e.*, Line 4; $O\left(TCn^2\right)$ for constructing the MMD matrices, i.e., Lines 2 and 6; $O\left(Tmn\right)$ for all other steps. In summary, the overall computational complexity of Algorithm 1 is $O\left(Tkm^2 + TCn^2 + Tmn\right)$.

**Algorithm 1:** JDA: Joint Distribution Adaptation

```
Input: Data $\mathbf{X}$, $\mathbf{y}_s$; #subspace bases $k$, regularization parameter $\lambda$.
Output: Adaptation matrix $\mathbf{A}$, embedding $\mathbf{Z}$, adaptive classifier $f$.
1 begin
2     Construct MMD matrix $\mathbf{M}_0$ by Eq. (4), set $\{\mathbf{M}_c := \mathbf{0}\}_{c=1}^{C}$.
3     repeat
4         Solve the generalized eigendecomposition problem in
          Equation (10) and select the $k$ smallest eigenvectors to
          construct the adaptation matrix $\mathbf{A}$, and $\mathbf{Z} := \mathbf{A}^{\mathrm{T}}\mathbf{X}$.
5         Train a standard classifier $f$ on $\left\{(\mathbf{A}^{\mathrm{T}}\mathbf{x}_i, y_i)\right\}_{i=1}^{n_s}$ to
          update pseudo target labels $\{\widehat{y}_j := f(\mathbf{A}^{\mathrm{T}}\mathbf{x}_j)\}_{j=n_s+1}^{n_s+n_t}$.
6         Construct MMD matrices $\{\mathbf{M}_c\}_{c=1}^{C}$ by Equation (6).
7     until Convergence
8     Return an adaptive classifier $f$ trained on $\{\mathbf{A}\mathbf{x}_i, y_i\}_{i=1}^{n_s}$.
```

## 4. Experiments

In this section, we conduct extensive experiments for image classification problems to evaluate the JDA approach. Datasets and codes will be available online on publication.

## 4.1. Data Preparation

USPS+MNIST, COIL20, PIE, and Office+Caltech (refer to Figure 2 and Table 2) are six benchmark datasets widely adopted to evaluate visual domain adaptation algorithms.

**USPS** dataset consists of 7,291 training images and 2,007 test images of size $16 \times 16$.

**MNIST** dataset has a training set of 60,000 examples and a test set of 10,000 examples of size $28 \times 28$.

From Figure 2, we see that USPS and MNIST follow very different distributions. They share 10 classes of digits. To speed up experiments, we construct one dataset *USPS vs MNIST* by randomly sampling 1,800 images in USPS to form the source data, and randomly sampling 2,000 images in MNIST to form the target data. We switch source/target pair to get another dataset *MNIST vs USPS*. We uniformly rescale all images to size $16 \times 16$, and represent each one by a feature vector encoding the gray-scale pixel values. Thus the source and target data can share the same feature space.

**COIL20** contains 20 objects with 1,440 images. The images of each object were taken 5 degrees apart as the object is rotated on a turntable and each object has 72 images. Each image is $32 \times 32$ pixels with 256 gray levels per pixel.

In experiments, we partition the dataset into two subsets **COIL1** and **COIL2**: COIL1 contains all images taken in the directions of $[0°, 85°] \cup [180°, 265°]$ (quadrants 1 and 3); COIL2 contains all images taken in the directions of $[90°, 175°] \cup [270°, 355°]$ (quadrants 2 and 4). In this way, subsets COIL1 and COIL2 will follow relatively different distributions. We construct one dataset *COIL1 vs COIL2* by selecting all 720 images in COIL1 to form the source data, and all 720 images in COIL2 to form the target data. We switch source/target to get another dataset *COIL2 vs COIL1*.

**PIE**, which stands for "Pose, Illumination, Expression",

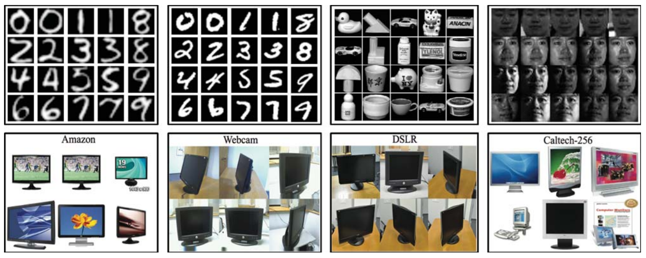


Figure 2. USPS, MNIST, COIL20, PIE, Office, and Caltech-256.

is a benchmark face database. The database has 68 individuals with 41,368 face images of size 32×32. The face images were captured by 13 synchronized cameras (different poses) and 21 flashes (different illuminations and/or expressions).

In these experiments, to thoroughly evaluate that our approach can perform robustly across different distributions, we adopt five subsets of PIE, each corresponding to a different pose. Specifically, we choose **PIE1** (C05, left pose), **PIE2** (C07, upward pose), **PIE3** (C09, downward pose), **PIE4** (C27, frontal pose), **PIE5** (C29, right pose). In each subset (pose), all the face images are taken under different lighting, illumination, and expression conditions. By randomly selecting two different subsets (poses) as the source domain and target domain respectively, we can construct $5 \times 4 = 20$ cross-domain face datasets, *e.g.*, *PIE1 vs PIE2*, *PIE1 vs PIE3*, *PIE1 vs PIE4*, *PIE1 vs PIE5*, $\ldots$, *PIE5 vs PIE4*. In this way, the source and target data are constructed using face images from different poses, and thus will follow significantly different distributions. Moreover, the distribution differences in these datasets may vary a lot, since for example, the difference between the left and right poses is larger than the difference between the left and frontal poses.

**Office** [20, 6] is an increasingly popular benchmark for visual domain adaptation. The database contains three real-world object domains, **Amazon** (images downloaded from online merchants), **Webcam** (low-resolution images by a web camera), and **DSLR** (high-resolution images by a digital SLR camera). It has 4,652 images and 31 categories.

**Caltech-256** [9] is a standard database for object recognition. The database has 30,607 images and 256 categories.

In these expriments, we adopt the public Office+Caltech datasets released by Gong *et al.* [6]. SURF features are extracted and quantized into an 800-bin histogram with codebooks computed with Kmeans on a subset of images from *Amazon*. Then the histograms are standardized by z-score. Specifically, we have four domains, **C** (Caltech-256), **A** (Amazon), **W** (Webcam), and **D** (DSLR). By randomly selecting two different domains as the source domain and target domain respectively, we construct $4 \times 3 = 12$ cross-domain object datasets, *e.g.*, $C \rightarrow A$, $C \rightarrow W$, $C \rightarrow D$, $\ldots$, $D \rightarrow W$.

## 4.2. Baseline Methods

We compare our JDA approach with five state-of-the-art (related) baseline methods for image classification problem.

Table 2. Statistics of the six benchmark digit/face/object datasets.

| Dataset | Type | #Examples | #Features | #Classes | Subsets |
|---|---|---|---|---|---|
| USPS | Digit | 1,800 | 256 | 10 | USPS |
| MNIST | Digit | 2,000 | 256 | 10 | MNIST |
| COIL20 | Object | 1,440 | 1,024 | 20 | COIL1, COIL2 |
| PIE | Face | 11,554 | 1,024 | 68 | PIE1, . . . , PIE5 |
| Office | Object | 1,410 | 800 | 10 | A, W, D |
| Caltech | Object | 1,123 | 800 | 10 | C |

- 1-Nearest Neighbor Classifier (NN)
- Principal Component Analysis (PCA) + NN
- Geodesic Flow Kernel (GFK) [6] + NN
- Transfer Component Analysis (TCA) [15] + NN
- Transfer Subspace Learning (TSL) [22] + NN

Specifically, TCA and TSL can both be viewed as a special case of JDA with $C = 0$. TSL adopts Bregman divergence instead of MMD as the distance for comparing distributions. As suggested by [6], NN is chosen as the base classifier since it does not require tuning cross-validation parameters.

## 4.3. Implementation Details

Following [6, 15], NN is trained on the labeled source data, and tested on the unlabeled target data; PCA, TSL, TCA, and JDA are performed on all data as a dimensionality reduction procedure, then an NN classifier is trained on the labeled source data for classifying the unlabeled target data.

Under our experimental setup, it is impossible to tune the optimal parameters using cross validation, since labeled and unlabeled data are sampled from different distributions. Thus we evaluate all methods by empirically searching the parameter space for the optimal parameter settings, and report the best results of each method. For subspace learning methods, we set #bases by searching $k \in [10, 20, \ldots, 200]$. For transfer learning methods, we set adaptation regularization parameter $\lambda$ by searching $\lambda \in \{0.01, 0.1, 1, 10, 100\}$.

The JDA approach involves only two model parameters: #subspace bases $k$ and regularization parameter $\lambda$. In the coming sections, we provide empirical analysis on parameter sensitivity, which verifies that JDA can achieve stable performance under a wide range of parameter values. In the comparative study, we set $k = 100$ and 1) $\lambda = 0.1$ for the digit/face datasets, 2) $\lambda = 1.0$ for the object datasets. The number of iterations for JDA to converge is $T = 10$. We do not run JDA repeatedly since it has no random initialization.

We use classification *Accuracy* on test data as the evaluation metric, which is widely used in literature [22, 15, 6]

$$Accuracy = \frac{|\mathbf{x} : \mathbf{x} \in \mathcal{D}_t \wedge \widehat{y}(\mathbf{x}) = y(\mathbf{x})|}{|\mathbf{x} : \mathbf{x} \in \mathcal{D}_t|} \quad (11)$$

where $\mathcal{D}_t$ is the set of test data, $y(\mathbf{x})$ is the truth label of $\mathbf{x}$, $\widehat{y}(\mathbf{x})$ is the label predicted by the classification algorithm.

## 4.4. Experimental Results

The classification accuracies of JDA and the five baseline methods on the 36 cross-domain image (digit, face, and

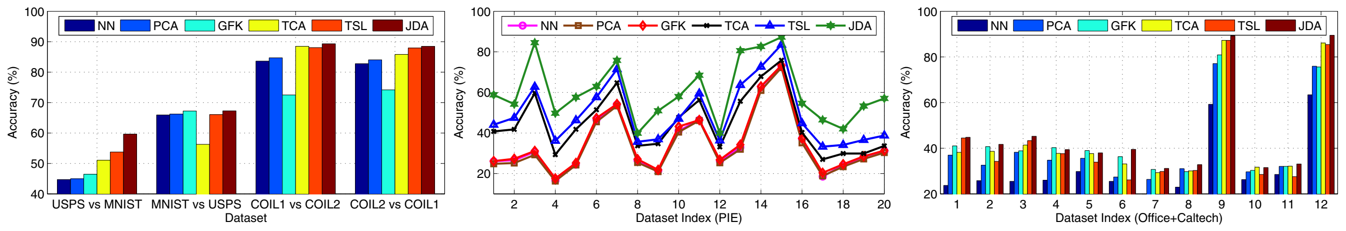


Figure 3. Accuracy (%) on the 4 types of 36 cross-domain image datasets, each under different difficulty in knowledge adaptation.

Table 3. Accuracy (%) on 4 types of cross-domain image datasets.

| Dataset | Standard Learning | | Transfer Learning | | | |
|---|---|---|---|---|---|---|
| | NN | PCA | GFK | TCA | TSL | JDA |
| USPS vs MNIST | 44.7 | 44.95 | 46.45 | 51.05 | 53.75 | **59.65** |
| MNIST vs USPS | 65.94 | 66.22 | 67.22 | 56.28 | 66.06 | **67.28** |
| COIL1 vs COIL2 | 83.61 | 84.72 | 72.50 | 88.47 | 88.06 | **89.31** |
| COIL2 vs COIL1 | 82.78 | 84.03 | 74.17 | 85.83 | 87.92 | **88.47** |
| PIE1 vs PIE2 (1) | 26.09 | 24.80 | 26.15 | 40.76 | 44.08 | **58.81** |
| PIE1 vs PIE3 (2) | 26.59 | 25.18 | 27.27 | 41.79 | 47.49 | **54.23** |
| PIE1 vs PIE4 (3) | 30.67 | 29.26 | 31.15 | 59.63 | 62.78 | **84.50** |
| PIE1 vs PIE5 (4) | 16.67 | 16.30 | 17.59 | 29.35 | 36.15 | **49.75** |
| PIE2 vs PIE1 (5) | 24.49 | 24.22 | 25.24 | 41.81 | 46.28 | **57.62** |
| PIE2 vs PIE3 (6) | 46.63 | 45.53 | 47.37 | 51.47 | 57.60 | **62.93** |
| PIE2 vs PIE4 (7) | 54.07 | 53.35 | 54.25 | 64.73 | 71.43 | **75.82** |
| PIE2 vs PIE5 (8) | 26.53 | 25.43 | 27.08 | 33.70 | 35.66 | **39.89** |
| PIE3 vs PIE1 (9) | 21.37 | 20.95 | 21.82 | 34.69 | 36.94 | **50.96** |
| PIE3 vs PIE2 (10) | 41.01 | 40.45 | 43.16 | 47.70 | 47.02 | **57.95** |
| PIE3 vs PIE4 (11) | 46.53 | 46.14 | 46.41 | 56.23 | 59.45 | **68.45** |
| PIE3 vs PIE5 (12) | 26.23 | 25.31 | 26.78 | 33.15 | 36.34 | **39.95** |
| PIE4 vs PIE1 (13) | 32.95 | 31.96 | 34.24 | 55.64 | 63.66 | **80.58** |
| PIE4 vs PIE2 (14) | 62.68 | 60.96 | 62.92 | 67.83 | 72.68 | **82.63** |
| PIE4 vs PIE3 (15) | 73.22 | 72.18 | 73.35 | 75.86 | 83.52 | **87.25** |
| PIE4 vs PIE5 (16) | 37.19 | 35.11 | 37.38 | 40.26 | 44.79 | **54.66** |
| PIE5 vs PIE1 (17) | 18.49 | 18.85 | 20.35 | 26.98 | 33.28 | **46.46** |
| PIE5 vs PIE2 (18) | 24.19 | 23.39 | 24.62 | 29.90 | 34.13 | **42.05** |
| PIE5 vs PIE3 (19) | 28.31 | 27.21 | 28.49 | 29.90 | 36.58 | **53.31** |
| PIE5 vs PIE4 (20) | 31.24 | 30.34 | 31.33 | 33.64 | 38.75 | **57.01** |
| C→A (1) | 23.70 | 36.95 | 41.02 | 38.20 | 44.47 | **44.78** |
| C→W (2) | 25.76 | 32.54 | 40.68 | 38.64 | 34.24 | **41.69** |
| C→D (3) | 25.48 | 38.22 | 38.85 | 41.40 | 43.31 | **45.22** |
| A→C (4) | 26.00 | 34.73 | **40.25** | 37.76 | 37.58 | 39.36 |
| A→W (5) | 29.83 | 35.59 | **38.98** | 37.63 | 33.90 | 37.97 |
| A→D (6) | 25.48 | 27.39 | 36.31 | 33.12 | 26.11 | **39.49** |
| W→C (7) | 19.86 | 26.36 | 30.72 | 29.30 | 29.83 | **31.17** |
| W→A (8) | 22.96 | 31.00 | 29.75 | 30.06 | 30.27 | **32.78** |
| W→D (9) | 59.24 | 77.07 | 80.89 | 87.26 | 87.26 | **89.17** |
| D→C (10) | 26.27 | 29.65 | 30.28 | 31.70 | 28.50 | **31.52** |
| D→A (11) | 28.50 | 32.05 | 32.05 | 32.15 | 27.56 | **33.09** |
| D→W (12) | 63.39 | 75.93 | 75.59 | 86.10 | 85.42 | **89.49** |
| Average | 37.46 | 39.84 | 41.19 | 47.22 | 49.80 | **57.37** |

object) datasets are illustrated in Table 3. The results are visualized in Figure 3 for better interpretation. We observe that JDA achieves much better performance than the five baseline methods with statistical significance. The average classification accuracy of JDA on the 36 datasets is **57.37%** and the performance improvement is **7.57%** compared to the best baseline method TSL, *i.e.*, a significant error reduction of **15.07%**. Note that, the adaptation difficulty in the 36 datasets varies a lot, since the standard NN classifier can only achieve an average classification accuracy of 37.46%, and may perform very poorly on many of the datasets. This verifies that JDA can construct more effective and robust representation for cross-domain image classification tasks.

Secondly, GFK performs pretty well on Office+Caltech datasets, but poorly on the other datasets. In GFK, the subspace dimension should be small enough to ensure different subspaces transit smoothly along the geodesic flow, which, however, may not represent input data accurately. JDA performs much better by learning an accurate shared subspace.

Thirdly, JDA significantly outperforms TCA, which is a state-of-the-art transfer learning method based on feature extraction. A major limitation of TCA is that the difference in the conditional distributions is not explicitly reduced. J-DA avoids this limitation and achieves much better results.

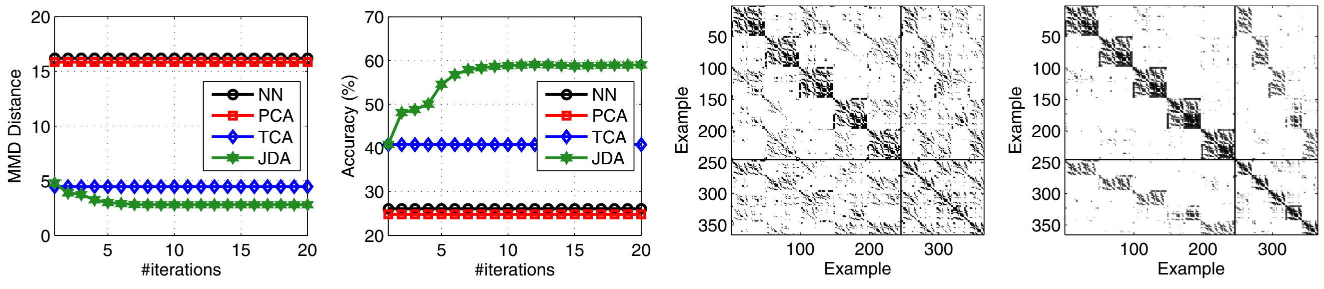
Lastly, JDA achieves much better performance than T-SL, which depends on the kernel density estimation (KDE) to match the marginal distributions. Theoretically, TSL can adapt the marginal distributions better than TCA, which is validated by the empirical results. However, TSL also does not explicitly reduce the difference in the conditional distributions. One possible difficulty for TSL to adapt the conditional distributions is that TSL relies on the distribution density, which is hard to manipulate with partially incorrect pseudo labels. JDA succeeds in matching the conditional distributions through exploring only the sufficient statistics.

## 4.5. Effectiveness Verification

We further verify the effectiveness of JDA by inspecting the distribution distance and the similarity of embeddings.

**Distribution Distance:** We run NN, PCA, TCA, and JDA on dataset *PIE1 vs PIE2* using their optimal parameter settings. Then we compute the aggregate MMD distance of each method on their induced embeddings by E-quation (7). Note that, in order to compute the true distance in both the marginal and conditional distributions between domains, we have to use the groundtruth labels instead of the pseudo labels. However, the groundtruth target labels are only used for verification, not for learning procedure.

Figure 4(a) shows the distribution distance computed for each method, and figure 4(b) shows the classification accuracy. We can have these observations. 1) Without learning a feature representation, the distribution distance of NN in the original feature space is the largest. 2) PCA can learn a new representation in which the distribution distance is s-lightly reduced, but not much, thus it cannot help much for cross-domain problems. 3) TCA can substantially reduce the distribution distance by explicitly reducing the differ-



(a) MMD distance w.r.t. #iterations (b) Accuracy (%) w.r.t. #iterations (c) Similarity of TCA embedding (d) Similarity of JDA embedding

Figure 4. Effectiveness verification: MMD distance, classification accuracy, and similarity of embeddings on the *PIE1 vs PIE2* dataset.

ence in the *marginal* distributions, so it can achieve better classification accuracy. 4) JDA can reduce the difference in *both* the marginal and conditional distributions, thus it can extract a most effective and robust representation for cross-domain problems. By iteratively refining the pseudo labels, JDA can reduce the difference in conditional distributions in each iteration to improve the classification performance.

**Similarity of Embeddings:** We run TCA and JDA on dataset *PIE1 vs PIE2* using their optimal parameter settings. Then we compute the 20-nearest neighbor similarity matrix on the embedding $\mathbf{Z} = \mathbf{A}^{\mathrm{T}}\mathbf{X}$ obtained by TCA and JDA respectively. For better illustration, we only use the 365 face images corresponding to the first 5 classes, in which the first 245 images are from the source data, the last 120 images are from the target data. Correspondingly, in the similarity matrix, the top-left and bottom-right submatrices indicate *within-domain* similarity, the top-right and bottom-left submatrices indicate *between-domain* similarity. Also, the diagonal blocks of the similarity matrix indicate *within-class* similarity in the same domain, the diagonal blocks of the top-right and bottom-left submatrices indicate *within-class* similarity across domains, while all other blocks of the similarity matrix indicate *between-class* similarity.

Figures 4(c) and 4(d) illustrate the similarity matrix of TCA embedding and JDA embedding respectively. To be an effective and robust embedding for cross-domain classification problems, 1) the *between-domain* similarity should be high enough to establish knowledge transfer, and 2) the *between-class* similarity should be low to facilitate category discrimination. In this sense, we see that TCA cannot extract a good embedding on which the *between-domain* similarity is high while the *between-class* similarity is low. This proves that only adapting the marginal distributions is not enough for transfer learning. It is interesting to observe that JDA can indeed learn the ideal embedding, which can lead to better generalization capability across different domains.

## 4.6. Parameter Sensitivity

We conduct sensitivity analysis to validate that JDA can achieve optimal performance under a wide range of parameter values. We only report the results on *USPS vs MNIST*, *PIE1 vs PIE2*, and $A \rightarrow D$ datasets, while similar trends on all other datasets are not shown due to space limitation.

Table 4. Time complexity of JDA and all the baseline methods.

| Method | Runtime (s) | Method | Runtime (s) | Method | Runtime (s) |
|---|---|---|---|---|---|
| NN | 4.85 | PCA | 2.63 | GFK | 4.58 |
| TCA | 3.80 | TSL | 1789 | JDA | 46.32 |

We run JDA with varying values of $k$. It can be chosen such that the low-dimensional representation is accurate for data reconstruction. We plot classification accuracy w.r.t. different values of $k$ in Figure 5(a), and choose $k \in [60, 200]$.

We run JDA with varying values of $\lambda$. Theoretically, larger values of $\lambda$ can make shrinkage regularization more important in JDA. When $\lambda \rightarrow 0$, the optimization problem is ill-defined. When $\lambda \rightarrow \infty$, distribution adaptation is not performed, and JDA cannot construct robust representation for cross-domain classification. We plot classification accuracy w.r.t. different values of $\lambda$ in Figure 5(b), which indicates that $\lambda \in [0.001, 1.0]$ can be optimal parameter values, where JDA generally does much better than the baselines.

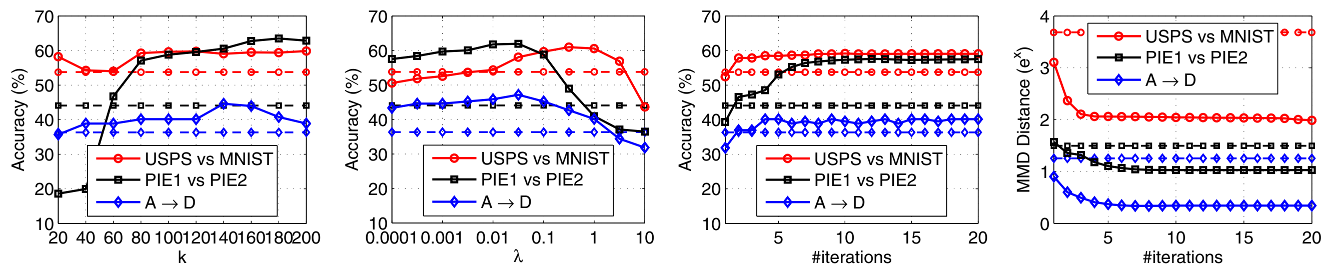
## 4.7. Convergence and Time Complexity

We also empirically check the convergence property of JDA. Figures 5(c) and 5(d) show that classification accuracy (distribution distance) increases (decreases) steadily with more iterations and converges within only 10 iterations.

We check the time complexity by running all algorithms on the *PIE1 vs PIE2* dataset with 1,024 features and 4,961 images, and show the results in Table 4. We observe that JDA is $T$-times worse than TCA but much better than TSL.

## 5. Conclusion and Future Work

In this paper, we propose a Joint Distribution Adaptation (JDA) approach for robust transfer learning. JDA aims to simultaneously adapt both marginal and conditional distributions in a principled dimensionality reduction procedure. Extensive experiments show that JDA is effective and robust for a variety of cross-domain problems, and can significantly outperform several state-of-the-art adaptation methods even if the distribution difference is substantially large.



(a) #subspace bases $k$ (b) regularization parameter $\lambda$ (c) accuracy w.r.t. #iterations (d) distance w.r.t. #iterations

Figure 5. Parameter sensitivity and convergence study for JDA on three types of datasets (dashed lines show the best baseline results).

In the future, we plan to extend our distance measure to other representation learning methods, *e.g.*, Sparse Coding.

## 6. Acknowledgments

This work is supported by the National HGJ Key Project (2010ZX01042-002-002-01), the National High-Tech R&D Program (863 Program) (2012AA040911), the National Basic Research Program (973 Program) (2009CB320700), and the National Natural Science Foundation of China (NSFC) (61073005, 61271394). Philip S. Yu is supported in part by US NSF through grants IIS-0905215, CNS-1115234, IIS-0914934, DBI-0960443, OISE-1129076, and US Department of Army through grant W911NF-12-1-0066.

## References

[1] A. Arnold, R. Nallapati, and W. W. Cohen. A comparative study of methods for transductive transfer learning. In *Proceedings of ICDMW*, 2007.

[2] Y. Aytar and A. Zisserman. Tabula rasa: Model transfer for object category detection. In *Proceedings of ICCV*, 2011.

[3] L. Bruzzone and M. Marconcini. Domain adaptation problems: A dasvm classification technique and a circular validation strategy. *IEEE TPAMI*, 32(5), 2010.

[4] M. Chen, K. Q. Weinberger, and J. C. Blitzer. Co-training for domain adaptation. In *Proceedings of NIPS*, 2011.

[5] N. FarajiDavar, T. de Campos, J. Kittler, and F. Yan. Transductive transfer learning for action recognition in tennis games. In *Proceedings of ICCVW*, 2011.

[6] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Proceedings of CVPR*, 2012.

[7] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *Proceedings of ICCV*, 2011.

[8] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Scholkopf, and A. J. Smola. A kernel method for the two-sample problem. In *Proceedings of NIPS*, 2006.

[9] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical report, Caltech, 2007.

[10] M. Guillaumin and V. Ferrari. Large-scale knowledge transfer for object localization in imagenet. In *Proceedings of CVPR*, 2012.

[11] I.-H. Jhuo, D. Liu, D.-T. Lee, and S.-F. Chang. Robust visual domain adaptation with low-rank reconstruction. In *Proceedings of CVPR*, 2012.

[12] L. Jie, T. Tommasi, and B. Caputo. Multiclass transfer learning from unconstrained priors. In *Proc. of ICCV*, 2011.

[13] C. H. Lampert and O. Krömer. Weakly-paired maximum covariance analysis for multimodal dimensionality reduction and transfer learning. In *Proceedings of ECCV*, 2010.

[14] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *Proceedings of CVPR*, 2009.

[15] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. *IEEE TNN*, 22(2):199–210, 2011.

[16] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE TKDE*, 22:1345–1359, 2010.

[17] Q. Qiu, V. M. Patel, P. Turaga, and R. Chellappa. Domain adaptive dictionary learning. In *Proceedings of ECCV*, 2012.

[18] B. Quanz, J. Huan, and M. Mishra. Knowledge transfer with low-quality data: A feature extraction issue. *IEEE TKDE*, 24(10), 2012.

[19] S. D. Roy, T. Mei, W. Zeng, and S. Li. Socialtransfer: Cross-domain transfer learning from social streams for media applications. In *Proceedings of ACM MM*, 2012.

[20] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *Proceedings of ECCV*, 2010.

[21] S. Satpal and S. Sarawagi. Domain adaptation of conditional probability models via feature subsetting. In *Proceedings of PKDD*, 2007.

[22] S. Si, D. Tao, and B. Geng. Bregman divergence-based regularization for transfer subspace learning. *IEEE TKDE*, 2010.

[23] Q. Sun, R. Chattopadhyay, S. Panchanathan, and J. Ye. A two-stage weighting framework for multi-source domain adaptation. In *Proceedings of NIPS*, 2011.

[24] H. Wang, F. Nie, H. Huang, and C. Ding. Dyadic transfer learning for cross-domain image classification. In *Proceedings of ICCV*, 2011.

[25] S. Wang, S. Jiang, Q. Huang, and Q. Tian. Multi-feature metric learning with knowledge transfer among semantics and social tagging. In *Proceedings of CVPR*, 2012.

[26] E. Zhong, W. Fan, J. Peng, K. Zhang, J. Ren, D. Turaga, and O. Verscheure. Cross domain distribution adaptation via kernel mapping. In *Proceedings of KDD*, 2009.