PUnDA: Probabilistic Unsupervised Domain Adaptation for Knowledge Transfer Across Visual Categories

Behnam Gholami[1], Ognjen (Oggi) Rudovic[2], Vladimir Pavlovic[1]
[1]Department of Computer Science, Rutgers University, Piscataway, NJ, USA
[2]MIT Media Lab, Cambridge, MA, USA
{bb510,vladimir}@cs.rutgers.edu, orudovic@mit.edu

Abstract

This paper introduces a probabilistic latent variable model to address unsupervised domain adaptation problems. Specifically, we tackle the task of categorization of visual input from different domains by learning projections from each domain to a latent (shared) space jointly with the classifier in the latent space, which simultaneously minimizes the domain disparity while maximizing the classifier's discriminative power. Furthermore, the non-parametric nature of our adaptation model makes it possible to infer the latent space dimension automatically from data. We also develop a novel regularized Variational Bayes (VB) algorithm for efficient estimation of the model parameters. We compare the proposed model with the state-of-the-art methods for the tasks of visual domain adaptation using both handcrafted and deep-net features. Our experiments show that even with a simple softmax classifier, our model outperforms several state-of-the-art methods that take advantage of more sophisticated classification schemes.

1. Introduction

Traditional machine learning algorithms assume that the training and test data are independent and identically distributed (i.i.d.), coming from the same underlying distribution [41]. However, in real-world data, this assumption rarely holds due to a number of artifacts, such as different types of noise, changes in object view, etc. This inevitably introduces different types of biases in the observed data sampled during the training and test stage.

Domain Adaptation (DA) approaches [11, 17, 31, 37, 2, 22, 26, 47] have been proposed to compensate for these effects. The goal of DA is to leverage the knowledge from one domain to improve the model's performance on another domain. One way to reduce the adverse effects of the domain shift is to use an extensive set of labeled training data (i.e., the source domain), hoping that these will eventually

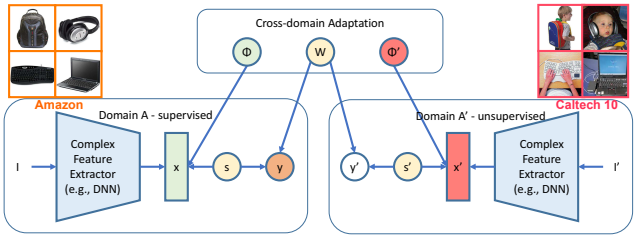


Figure 1. The overview of the proposed **PUnDA** approach. We jointly learn the mapping Φ and Φ' linking the original domain features x and x' to the matched subspaces s and s', automatically inferring the subspace dimension, and the cross-domain classifier W. Both domains use the same (pretrained) feature extractors (e.g., DNNs). Target domain A' does not rely on labels. Details of the model are specified in Fig. 2.

contain data from a distribution similar to that of the test data (i.e., the target domain). The most representative examples of this are the recent trends in deep learning, which have shown great improvements in the performance of supervised learning tasks due to the available substantial amount of labeled data [30, 13, 20]. However, obtaining labels is labor-intensive and expensive if one is to label training data across all possible domains. More importantly, due to the inherent bias within different datasets [43, 38], using a large amount of (labeled) training data does not warrant a better performance by these models. In this case, DA methods are a good choice as they can smartly leverage the available information across domains to correct for the domain shift, reducing the need for large amount of labeled data, while also correcting for differences in the distributions of the input features (covariate shift).

Based on their learning assumptions, existing DA methods can be divided into two categories: (semi)supervised DA [27, 9, 28, 45], and unsupervised DA [18, 16, 3, 13, 31, 32]. The former assumes that in addition to the labeled data of the source domain, some labeled data from the target domain are also available for training/adapting the classi-

fiers. By contrast, the latter does not require any labels from the target domain. While the labeled data should always be used when available (as this allows for more effective DA), there are many real-world applications where obtaining the labels is impractical and/or infeasible. This calls for unsupervised DA, the only feasible DA approach for many real-world unsupervised problems. In this paper, we address the unsupervised DA learning tasks for visual categories.

In most existing unsupervised DA methods, the first step is to project the source and target data onto a common space such that the source data is as close as possible to the target data in their distribution [18, 16, 3, 13, 31, 25, 34, 15]. Then, a classifier trained on the transformed source domain is applied to the target data, hoping that it will perform equally well across the domains as the domain mismatch is minimized through the learned projections. Hence, these methods have an underlying assumption that the shift in the two distributions (termed covariate shift [40]) can be reduced without relying on the labels from the target domain. However, most of these methods suffer from at least one or more of the following limitations that can adversely affect their performance and/or constrain their applicability to the target tasks. First, majority of existing methods are deterministic [18, 16, 3, 13, 31], relying on costly cross-validation procedures to find the size of the underlying manifold in which the mismatch between the source and target domains can effectively be reduced — increasing the computational complexity of the model and making it more prone to overfitting. Second, the minimization of the domain mismatch and learning of the target classifiers are done independently resulting in the joint feature space that is suboptimal for the main task, i.e., classification.

To overcome the above-mentioned limitations, we introduce a novel probabilistic framework that we call Probabilistic Unsupervised Domain Adaptation (**PUnDA**). In contrast to existing two-stage approaches where new feature spaces and classifiers are separately learned, our approach learns both the classifier and low dimensional subspace jointly via a newly introduced Bayesian learning framework. Moreover, the probabilistic nature of our **PUnDA** allows it to automatically infer the dimensionality of the common subspace. Since these benefits come with computational challenges if not addressed properly, we introduce an efficient learning and inference method based on the variational Bayes (VB) framework. Within this framework, we also propose an extension of the Maximum Mean Discrepancy (MMD) score [6], traditionally used to measure the domain mismatch, with the aim to align the source and target domains via estimated (variational) posteriors - thus, exploiting the model uncertainty — something the deterministic approaches fail to account for. Finally, the proposed VB learning in our **PUnDA** allows us to effectively incorporate unlabeled data of the target domain into the classifier learning via the regularizers specifically designed to minimize the expected classification loss in target domain. Our method is expected to bring most benefits in the DA cases when: (i) the data in both the source and target domains are tightly clustered, and (ii) the clusters from the two domains are geometrically close to each other. We show in our experiments on several benchmark datasets that the proposed approach significantly outperforms the state-of-the-art methods for unsupervised DA as they fail to account for the properties exploited in our **PUnDA** approach. An overview of this approach is shown in Fig. 1.

2. Related Work

Existing methods for DA can be divided into three main categories: 1) **Instance-based methods** (I-DA) [23, 12, 7], where the goal is to perform re-weighting of the source domain samples (in their loss function) in order to minimize the difference between the target and source domains. 2) **Feature learning-based methods** (FL-DA) [37, 35, 18, 17, 29] aims to transform the original source and target feature spaces to a shared subspace preserving the commonalities between the source and target domain. 3) **Model-based methods** operate directly on the model parameters (of the source classifier) by adjusting them based on the input distribution of the target domain (typically) without changing the feature space [46, 1, 24]. Since the unsupervised methods for DA are mainly based on the feature adaptation (due to the lack of target labels), in what follows, we review the methods for FL-DA. For a general overview of existing DA methods, see [36].

One of the first FL-DA approaches is the Transfer Component Analysis (TCA) [35]. The main idea is to find a low-dimensional linear transformation such that the source and target domains are as close as possible in their marginal distributions, while maintaining the intrinsic structure of the original domains. The latter is achieved by incorporating a local geometry (manifold) preserving regularization term into the TCA's objective function. Likewise, [37] proposed a metric learning-based DA method with cross-domain constraints. This method learns a symmetric transformation to map source and target domain samples onto a new domain invariant space. [18] proposed an feature alignment method for DA based on the Sampling Geodesic Flow (SGF) that exploits the geodesic distance between the source and target subspaces. Likewise, [41] proposed a simple but effective method for unsupervised DA called Correlation Alignment (CORAL), which minimizes domain shift by aligning the second-order statistics of source and target distributions.

Instead of aligning the source and target domains in a (low) dimensional manifold, a few works attempted to reduce the domain mismatch by expanding the source and target features in a non-parametric fashion using the notion of Reproducing Kernel Hilbert Spaces (RKHS). The main

assumption here is that in RKHS the domains can be brought together more easily compared to parametric (fix-dimension) transformations. Specifically, [3, 4] proposed the Domain Invariant Projection (DIP) method that compares the domain distributions in RKHS, while constraining the transformation to be orthogonal. More recently, [21] proposed a DA scheme to construct a RKHS using the Mahalanobis metric in the target space. This is achieved by simultaneously learning the projections from the source and target domains to RKHS, by minimizing a notion of domain distance while maximizing a measure of discriminatory power of RKHS.

The models reviewed lack the key properties of our **PUnDA** approach: the majority of the models that perform learning of the common subspace are deterministic, and therefore do not account for the uncertainty during feature adaptation - resulting in less robust measures of the domain mismatch, used to find the subspace. More importantly, because of their non-Bayesian treatment, most of these methods cannot automatically reveal the optimal subspace dimension. On the other hand, the non-parametric methods that use the notion of RKHS can easily lead to overfitting of the available target data. More importantly, in contrast to our approach, the learning of the target classifier and the domain alignment in these methods is done independently - rendering suboptimal models for the classification task.

3. Proposed Method

In this section, we present **PUnDA** for unsupervised DA. We consider a multi-class classification problem as the running example. Specifically, suppose we are given source-domain training examples $\boldsymbol{X} = [x_1, ..., x_N] \in \mathbb{R}^{d \times N}$, with labels $Y = [y_1, ..., y_N] \in \mathbb{R}^{1 \times N}$, $y \in \{1, 2, ..., C\}$ (we assume the shared set of class labels between the two domains), and target data $\boldsymbol{X}' = [x'_1, ..., x'_M] \in \mathbb{R}^{d \times M}$. Our goal is to assign the correct class label Y' to target data points $\boldsymbol{X}'$. Fig. 2 shows the model's representation as a Bayesian network. There are three observed variables represented by the shaded nodes: the source features $\{x_i\}$, the target features $\{x'_j\}$, and the source labels Y. Note that we assume that we do not have access to target labels, hence, $Y' = [y'_1, ..., y'_M]$ are unobserved. By assuming the existence of a low-dimensional latent space where the source and target distributions are similar, we model each feature x_i/x'_i, as a linear transformation $\boldsymbol{\Phi}/\boldsymbol{\Phi}'$ of their latent representations s_i/s'_i in the source/target domain, corrupted with an additive Gaussian noise ϵ/ϵ', as

$$x_i = \boldsymbol{\Phi}^\top s_i + \epsilon, \quad x'_i = \boldsymbol{\Phi}'^\top s'_i + \epsilon', \tag{1}$$

where $\boldsymbol{\Phi} = [\phi_1, ..., \phi_K] \in \mathbb{R}^{d \times K}$, and $\boldsymbol{\Phi}' = [\phi'_1, ..., \phi'_K] \in \mathbb{R}^{d \times K}$ are the transformation matrices for source and target domains, respectively. $\epsilon \sim \mathcal{N}(0, \gamma_s^{-1} \boldsymbol{I}_d)$ and $\epsilon' \sim \mathcal{N}(0, \gamma_t^{-1} \boldsymbol{I}_d)$ are the zero-mean Gaussian noise with precision values γ_s and γ_t, respectively ($\boldsymbol{I}_d$ denotes a $d \times d$ identity matrix). To keep the exponential family conjugacy between the prior and likelihood distributions, we place non-informative gamma hyper-priors on γ_s and γ_t, as

$$\gamma_s \sim Ga(c_1, c_2), \quad \gamma_t \sim Ga(c'_1, c'_2),$$

where Ga denotes the Gamma distribution.

To automatically infer the dimensionality K of the shared latent space, we introduce an auxiliary binary vector $Z \in \{0, 1\}^K$ for the latent features $\{s_i\}$, $\{s'_j\}$, where the non-zero entries of Z specify which latent features are used to represent the observations. Consequently, the model in Eq. 1 is reformulated as

$$x_i = \boldsymbol{\Phi}^\top (Z \odot s_i) + \epsilon, \quad x'_i = \boldsymbol{\Phi}'^\top (Z \odot s'_i) + \epsilon', \tag{2}$$

where $\odot$ denotes the element-wise multiplication operator. Note that all the source and target data points (x_i/x'_j) share the same set of important latent features defined by Z, but each have their unique weights (s_i/s_j).

Using the notion of the probabilistic hierarchical framework as in [10], we place a non-parametric prior on the binary vector Z by introducing auxiliary variables $\boldsymbol{\Pi} = \{\pi_k\}_{k=1}^K$ drawn from the Beta distribution as

$$\pi_k \sim Beta(a/K, b(K-1)/K),$$

where a, b are the hyper-parameters and the integer K is the largest possible dimension for Z (by letting $K \to \infty$, the length of the binary code Z can be learned from the observed

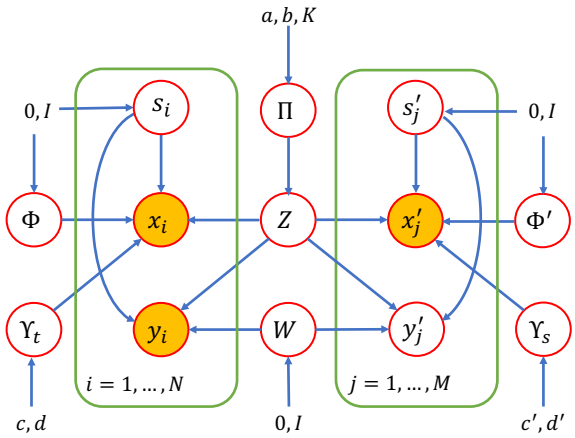


Figure 2. The graphical representation of **PUnDA** (the shaded circles denote the observed data). $\{x_i, x'_j\}$ are the source and the target variables in the observation space, $\{y_i, y'_j\}$ are the labels of the source and the target data, and $\{s_i, s'_j\}$ are the representation of two domains in the shared space. $\boldsymbol{\Phi}, \boldsymbol{\Phi}'$ are the projection matrices. The elements of $\boldsymbol{W}$ are the classifier parameters that are shared between both the source and target domains. Z defines the underlying dimension of the shared space, and γ_s and γ_t are the noise parameters of the source and target domain, respectively.

data [42]). Then, we model the binary vector Z as a random sample from the Bernoulli process parameterized by $\mathbf{\Pi}$ as

$$Z \sim \prod_{k=1}^{K} Ber(z_k; \pi_k), \quad k = 1, ..., K,$$

where z_k denotes the k-th element of the binary vector Z and Ber denotes the Bernoulli distribution (we obtain the Indian Buffet Process (IBP) prior[19] on Z by integrating out $\mathbf{\Pi}$ and letting $K \to \infty$). For computational simplicity, we model the latent features $\boldsymbol{S} = [s_1, ..., s_N] \in \mathbb{R}^{K \times N}$ and $\boldsymbol{S}' = [s'_1, ..., s'_M] \in \mathbb{R}^{K \times M}$ using a multivariate zero-mean Gaussian distribution:

$$P(s_i) \sim \mathcal{N}(0, \boldsymbol{I}_K), \; P(s'_j) \sim \mathcal{N}(0, \boldsymbol{I}_K).$$

Similarly, we also assume that the elements of the transformation matrices are drawn from a multivariate zero-mean Gaussian distribution:

$$P(\phi_i) \sim \mathcal{N}(0, \boldsymbol{I}_d), \; P(\phi'_j) \sim \mathcal{N}(0, \boldsymbol{I}_d).$$

In order to make the latent representations discriminative for the classification task, we employ the softmax regression classifier. More precisely, for the shared space representation $Z \odot s$ of a sample x, the probability of the x's label y belonging to class $c = 1, \ldots, C$ is computed as:

$$P(y = c|\boldsymbol{W}, s, Z) = \frac{exp\big(w_c^\top (Z \odot s)\big)}{\sum_{c'=1}^{C} exp\big(w_{c'}^\top (Z \odot s)\big)},$$

where $\boldsymbol{W} = [w_1, ..., w_C] \in \mathbb{R}^{(K+1) \times C}$ contains the class projection vectors. Again, within our probabilistic framework, we assume that elements of $\boldsymbol{W}$ are drawn from a multivariate zero-mean Gaussian distribution ($w_c \sim \mathcal{N}(0, \boldsymbol{I}_{K+1})$). It is worth noting that $\boldsymbol{W}$ includes a bias by having an extra dimension $s_0 = 1$ and $z_0 = 1$ for s and Z, respectively.

3.1. Posterior Inference

Because computing the exact posterior distribution of the latent variables $\Omega = \{\boldsymbol{S}, \boldsymbol{S}', \boldsymbol{W}, \mathbf{\Phi}, \mathbf{\Phi}', Z, \mathbf{\Pi}, \gamma_s, \gamma_t\}$ is intractable, we derive a Variational Bayes (VB) algorithm [14] to approximate this posterior distribution in the proposed **PUnDA** approach.

The goal of the VB is to approximate the true posterior distribution over the latent variables $P(\Omega|\boldsymbol{X}, \boldsymbol{X}', Y)$ with a variational distribution $q(\Omega)$, which is closest in KL divergence to the true posterior distribution. It is easy to show that this equals to maximizing the lower bound of the marginal likelihood $P(\boldsymbol{X}, Y, \boldsymbol{X}'|\mathbf{\Theta})$

$$q^*(\mathbf{\Omega}) = \arg\max_{q(\mathbf{\Omega})} \mathbb{E}_q\big[\log(\boldsymbol{X}, Y, \boldsymbol{X}', \mathbf{\Omega}|\mathbf{\Theta})\big] + \boldsymbol{H}[q(\mathbf{\Omega})],$$

where $\mathbf{\Theta} = \{a, b, K, c, d, c', d'\}$ denotes the set of hyper-parameters, $\mathbb{E}_q[.]$ denotes the expectation operator under the distribution q, and $\boldsymbol{H}[.]$ the entropy operator. For our framework to yield a computationally effective inference method, we employ a factorized variational distribution:

$$\begin{aligned} q(\mathbf{\Omega}) =& \Pi_{i=1}^{N} q(s_i) \Pi_{j=1}^{M} q(s'_j) \Pi_{k=1}^{K} q(\phi_k) q(\phi'_k) \\ & \Pi_{c=1}^{C} q(w_c) q(\gamma_s) q(\gamma_t). \end{aligned}$$

For simplicity, we also fix K and set it to a finite but large number. If K is large enough (see Sec. 4), the observed data will reveal fewer than K components for shared space features.

Apart from maximizing the marginal likelihood, we also need the shared latent features to be invariant to differences between the source and target domains, i.e., to be robust to the covariate shift that may exist in the target space. To this end, we introduce a regularizer $\mathcal{L}(\boldsymbol{S}, \boldsymbol{S}')$, based on on the Maximum Mean Discrepancy (MMD) [6], designed to minimize the distance between the distributions of the source and target representations. Specifically, given two sets of source/target samples, the MMD measures the distance between the mean of the two sets after mapping each sample to a RKHS:

$$MMD^2(\boldsymbol{S}, \boldsymbol{S}') = \left\| \sum_{i=1}^{N} \frac{\mathcal{F}(s_i)}{N} - \sum_{j=1}^{M} \frac{\mathcal{F}(s'_j)}{M} \right\|^2, \quad (3)$$

where $\mathcal{F}(.)$ denotes the target mapping. In practice, this mapping is typically unknown. By expanding Eq. 3, and using the kernel trick to replace the inner products by their kernel values, we rewrite the squared MMD, leading to the following regularizer:

$$\mathcal{L}(\boldsymbol{S}, \boldsymbol{S}') = \sum_{i,i'} \frac{\mathcal{K}(s_i, s_j)}{N^2} - 2 \sum_{i,j} \frac{\mathcal{K}(s_i, s'_j)}{NM} + \sum_{j,j'} \frac{\mathcal{K}(s'_j, s'_{j'})}{M^2},$$

where $\mathcal{K}(.,.)$ denotes the kernel function. In contrast to most existing DA methods that measure the domain distance directly in the learned RKHS [35, 3, 30, 44, 29], **PUnDA** encodes this distance using the posterior distributions of the shared features $\boldsymbol{S}$ and $\boldsymbol{S}'$ – thus, accounting also for uncertainty of the projections from the two domains. To this end, we use the Bhattacharyya kernel [8] to measure the posterior similarity as

$$\mathcal{K}(q(s), q(s')) = \log \int_{\mathbb{R}^K} q(s)^{1/2} q(s')^{1/2} \, ds \, ds'.$$

The intuition behind this kernel is that it measures the amount of overlap (similarity) between two distributions $q(s)$ and $q(s')$, by integrating the square root of their product over the whole space [8].

To learn a good classifier, we also leverage knowledge of the target domain samples by minimizing the uncertainty of

the classifier over the target samples. To this end, we introduce a regularizer $\mathcal{L}'(\boldsymbol{W}, \boldsymbol{S}', Z)$ designed to minimize the Shanon Entropy of the probability vectors $P(y'_j|\boldsymbol{W}, s'_j, Z)$ over the target domain samples:

$$\mathcal{L}'(\boldsymbol{W}, \boldsymbol{S}', Z) = \sum_{j=1}^{M} \sum_{c=1}^{C} \mathbb{E}_{P(y'_j|\boldsymbol{W}, s'_j, Z)} \log P(y'_j = c).$$

Intuitively, if our assumptions about two sets of clusters being geometrically close indeed hold in the used datasets, the probability vector $P(y'_j|\boldsymbol{W}, Z, s_j) = [p_j^1, ..., p_j^C]$ should ideally look like a posterior probability vector $[0, 0, ..., 1, ..., 0]$ (using 1-of-many coding). Since we do not know the true label, we cannot measure directly the similarity of $P(y'_j|\boldsymbol{W}, Z, s_j)$ and the correct label. However, we can minimize the entropy of $P(y'_j|\boldsymbol{W}, Z, s_j)$ by which we can reduce the amount of information that $P(y'_j|\boldsymbol{W}, Z, s_j)$ contains about the confusing labels.

By defining the regularizers $\mathcal{L}(\boldsymbol{S}, \boldsymbol{S}')$ and $\mathcal{L}'(\boldsymbol{S}', \boldsymbol{W}, Z)$, the proposed regularized VB algorithm can be written as the following optimization problem:

$$q^*(\boldsymbol{\Omega}) = \arg\max_{q(\boldsymbol{\Omega})} \mathbb{E}_q\big[\log(\boldsymbol{X}, Y, \boldsymbol{X}', \boldsymbol{\Omega}|\boldsymbol{\Theta})\big] + \boldsymbol{H}[q(\boldsymbol{\Omega})] - \lambda\mathcal{L}(\boldsymbol{S}, \boldsymbol{S}') + \lambda'\mathcal{L}'(\boldsymbol{S}', \boldsymbol{W}, Z),$$

where $\lambda \geq 0$ and $\lambda' \geq 0$ denote the regularization parameters. The VB algorithm solves the above optimization problem using the Coordinate Descent algorithm. The computational complexity of each iteration of the proposed VB algorithm, for training, in one iteration is $O((N+M)dK^2)$, i.e., linear in the size of the source+target data $N+M$, the data dimensionality d, and quadratic in the dimensionality of the shared space $K(K << d)$. Details of the proposed VB algorithm and its computational complexity analysis, along with other derivations, are available in the Supplementary Material.

3.2. Target Class Label Prediction

After computing the posterior distribution $q^*(\boldsymbol{\Omega})$, to determine the target class-label y'_j of a given target domain instance x'_j, we first compute the distribution of y'_j given x'_j by integrating out the latent variables $\{\boldsymbol{W}, Z, s'_j\}$. Then, we select the most likely label as

$$\hat{y}'_j = \arg\max_{y'_j \in \{1,...,C\}} P(y'_j|x'_j),$$

where $P(y'_j|x'_j)$ can be computed as

$$P(y'_j = c|x'_j) = \sum_{Z} \int P(y'_j|\boldsymbol{W}, Z, s'_j) q^*(Z) q^*(s'_j) q^*(\boldsymbol{W})\, dZ\, ds'_j\, d\boldsymbol{W}.$$

Since the above expression cannot be computed in a closed form, we approximate $q^*(Z), q^*(s'_j)$, and $q^*(\boldsymbol{W})$ with their mean values: $\mathbb{E}_{q^*(Z)}[Z], \mathbb{E}_{q^*(\boldsymbol{W})}[\boldsymbol{W}]$ and $\mathbb{E}_{q^*(s'_j)}[s'_j]$, respectively. Using this approximation, we compute y'_j as:

$$\hat{y}'_j = \arg\max_{c \in \{1,...,C\}} \mathbb{E}_{q^*(w_c)}[w_c]^\top (\mathbb{E}_{q^*(Z)}[Z] \odot \mathbb{E}_{q^*(s'_j)}[s'_j]).$$

4. Experimental Results

We run extensive experiments on unsupervised DA tasks, where we use the handcrafted features (SURF) [5] and the current state-of-the-art deep-net features (VGG-Net) [39], employing the same datasets/features as in [21]. We compare our approach to several state-of-the-art unsupervised DA methods on two DA benchmark datasets: the Office+Caltech10[1] and Multi-PIE[2] datasets.

Office+Caltech10 dataset contains images collected from four different sources (see Fig. 3) and 10 object classes. The corresponding domains are **A**mazon, **W**ebcam, **DSLR**, and **C**altech. The Multi-PIE dataset includes face images of 67 individuals captured from different expressions, views, and illumination conditions. We compare the performance of the proposed **PUnDA** approach to the following benchmarks:

- **1-NN** and **SVM**: original features are used without any adaptation, a basic 1-nearest neighbor (1-NNs) and linear SVM is found by comparing the target samples to the training data from the source domain.
- **GFK**[17]: The geodesic flow kernel algorithm. Results are evaluated using the kernel-NNs.
- **SA**[11]: The subspace alignment algorithm. Results are evaluated using 1-NN.
- **CORAL** [41]: The correlation alignment algorithm that uses a linear SVM on the similarity matrix formed by the correlation matching.
- **ILS** [21]: Invariant Latent Space algorithm. Results are evaluated using 1-NN.

To have a fair comparison, we use the accuracy reported by other authors with exactly the same experimental settings and source codes provided by the authors.

4.1. Implementation Details

In our experiments, we follow the standard setup in both datasets with the train/test splits provided by [21]. For the VB algorithm, we set the truncation level for the dimensionality of the latent space to $(K = 100)$ for both datasets. The

[1] https://cs.stanford.edu/~jhoffman/domainadapt

[2] http://www.cs.cmu.edu/afs/cs/project/PIE/MultiPie/Multi-Pie/Home.html

Figure 3. Exemplary images from the Office+Caltech10 dataset.

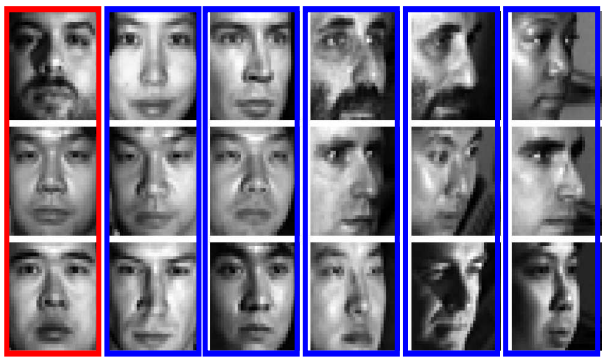


Figure 4. Three instances of the Multi-PIE face data. Here, the view from $C27$ is used as the source domain (first column). Remaining views (columns $2-6$ represent samples from $C09$, $C05$, $C37$, $C25$, and $C02$ respectively) are considered to be the target for each transformation.

hyper-parameters a, b of the Beta distributions are set with $a = 1$ and $b = 1$ (other settings of a and b yield similar results). All Gamma priors are set as $Ga(10^{-6}, 10^{-6})$ to make the prior distributions uninformative. In all our experiments, we set $\lambda = 0.1$ and $\lambda' = 1$. We use the classification accuracy for the target data as the evaluation metric:

4.2. Results for OFFICE+CALTECH10

For this dataset, we used 4096 dimensional VGG-fc6 and VGG-fc7 features extracted with the network model of [39] for the deep-net feature experiments. Following the experimental protocol in [21], we also use SURF features [5] (each image is encoded with an 800-bin histogram and the histograms are then normalized to have zero mean and unit standard deviation in each dimension) as hand-crafted features. We set the latent space dimensionality to 20 for VGG features and to 100 for SURF features in all compared methods, as these were empirically found to be the best for the competing methods [21]. For each pair of the source and target domains, we conduct experiments using 20 random train/test splits.

In Tables 1&3, we report the performance using VGG-FC6, VGG-FC7 and SURF features, respectively. As can be seen, for all the feature types, **PUnDA** outperforms the state-of-the-art methods in most of domain transformations, and, generally, provides the highest overall classification accuracies for all the feature types. We also note that the VGG-fc7 is less favorable than VGG-fc6 for majority of the DA algorithms compared.

The higher performance of **PUnDA** compared to other methods is mainly attributed to the joint learning of the discriminative classifier and low-dimensional feature spaces. The key observation is that good representations are beneficial to data classification, with classification results providing supervisory signals to representation learning. Furthermore, from the results obtained, it is obvious that it is more beneficial to make the use of information coming from unlabeled target data during classifier learning process compared to when no data from target domain is used. Indeed, using the proposed learning scheme, we find a representation space in which we embed the knowledge from the target domain into the learned classifier.

4.2.1 Sensitivity Analysis

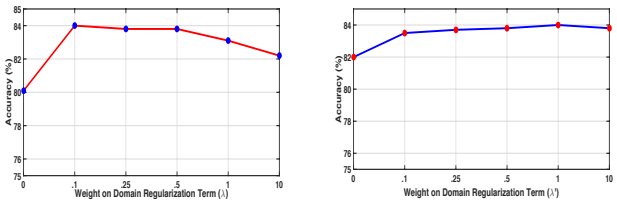
In the experiments above, we keep $\lambda = 0.1, \lambda' = 1$. To analyze the sensitivity of our method to changes in parameters λ and λ', we conducted additional experiments to analyze the parameter sensitivity of **PUnDA** w.r.t. the various values of λ and λ'. To this end, we consider random splits from each of the Office+Caltech10 dataset along VGG-FC6 features here. Fig. 5 shows the sensitivity analysis for the parameters of **PUnDA** on these random splits. Sensitivity analysis is performed by varying one parameter at the time over a given range, while for the other parameters we set them to their final values ($\lambda = 0.1, \lambda' = 1$). From Fig. 5 (a), we see that when $\lambda = 0$ (no domain mismatch regularization term is considered), the performance drops considerably. For other values of λ, the performance is superior and there is little

Table 1. Unsupervised domain adaptation results using VGG-FC6 features on Office+Caltech10 dataset with the evaluation setup of [21]. The best (bold red), the second best (red).

method	A → W	A → D	A → C	W → A	W → D	W → C	D → A	D → W	D → C	C → A	C → W	C → D	Ave.
1-NN	60.9	52.3	70.1	66.4	91.3	60.2	57.0	86.7	48.0	81.9	65.9	55.6	66.4
SVM	63.1	51.7	74.2	73.3	94.2	68.2	58.7	91.8	55.5	86.7	74.8	61.5	71.1
GFK[17]	74.1	63.5	77.7	81.1	**96.6**	73.5	69.9	92.4	64.0	86.2	76.5	66.5	76.8
SA[11]	76.0	64.9	77.1	80.2	94.2	71.9	69.0	90.5	62.3	83.9	76.0	66.2	76.0
CORAL[41]	74.8	67.1	79.0	82.3	96.0	75.9	75.8	**94.6**	64.7	89.4	77.6	67.6	78.7
ILS[21]	82.4	72.5	78.9	**87.2**	89.3	79.9	79.2	94.2	66.5	87.6	84.4	73.0	81.3
PUnDA	**82.7**	**76.2**	**82.3**	86.9	89.8	**82.6**	**83.1**	93.4	**69.2**	**90.3**	**88.3**	**76.2**	**83.4**

Table 2. Unsupervised domain adaptation results using VGG-FC7 features on Office+Caltech10 dataset with the evaluation setup of [21]. The best (in bold red), the second best (in red).

method	A → W	A → D	A → C	W → A	W → D	W → C	D → A	D → W	D → C	C → A	C → W	C → D	Ave.
1-NN	64.0	50.8	72.6	67.8	88.8	64.2	61.2	88.2	52.8	82.6	65.3	54.9	67.8
SVM	68.0	51.8	76.2	74.6	93.0	70.6	58.7	91.2	56.0	86.7	74.8	61.3	71.9
GFK[17]	74.0	57.6	76.6	76.0	92.9	69.5	67.5	91.9	62.9	84.1	73.6	63.4	74.2
SA[11]	75.0	60.7	76.2	76.4	94.0	69.0	66.0	89.5	59.4	82.6	73.6	63.2	73.8
CORAL[41]	71.8	61.3	78.6	82.0	**94.6**	73.7	71.2	**93.5**	63.0	88.6	76.0	63.8	76.5
ILS[21]	80.9	71.3	78.4	**86.7**	88.2	76.3	76.5	91.8	66.2	87.1	80.1	67.1	79.2
PUnDA	**81.4**	**75.8**	**81.0**	85.7	90.1	**80.1**	**80.4**	92.0	**69.1**	**91.1**	**83.8**	**70.8**	**81.7**



(a) Sensitivity to λ (b) Sensitivity to λ'

Figure 5. Sensitivity analysis of **PUnDA**.

variation in the model performance, evidencing the robustness of **PUnDA** w.r.t. λ. Similarly, from Fig. 5 (b), **PUnDA** is largely insensitive to the parameter λ' over the specified range of its values. Moreover, it is clear that using the unlabeled target data improves the discriminative power of the classifier.

4.3. Results on Multi-PIE Faces

In this experiment, we follow the setting in [21] and use the views: $C27$ (looking forward) and $C09$ (looking down), as the source domain, and the views: $C05, C37, C02, C25$ (looking towards left in an increasing angle, see Fig. 4), as target domains. We expect the face inclination angle to reflect the complexity of transfer learning. We normalize the images to 32×32 pixels and use the vectorized gray-scale images as features. The dimensionality of the common feature space for all the feature learning-based methods is set to 100.

Table 4 shows the classification accuracy w.r.t. the increasing angle of inclination. As can be seen, **PUnDA** achieves the best performance (on average) as well as the best scores for the 3 views and the second best for the $C02$. Clearly, with the increasing camera angle, the feature structure changes up to a certain extent (the features become heterogeneous). However, our method produces good accuracies even under such challenging conditions.

4.4. Model Selection

To demonstrate the ability of the proposed method to learn the dimensionality of the latent space automatically, we conduct experiments on both Office+Caltech10 and Multi-PIE datasets. We consider a random split from A → W of the Office+Caltech10 dataset along VGG-FC6 features, and $C27 \rightarrow C25$ from the Multi-PIE dataset.

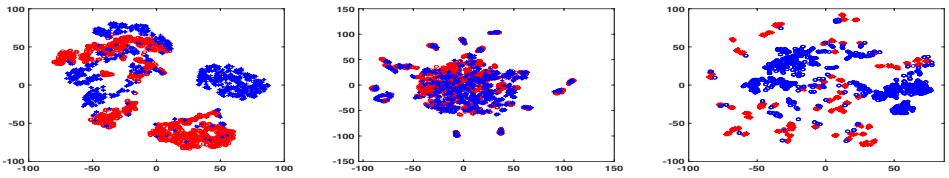
We plot the sorted values of $\mathbb{E}[q^*(Z)]$ for the selected source/target datasets, inferred by the algorithm in Fig. 7. As can be seen, the **PUnDA** inferred approximately $25 - 30$ dimensions for the learned latent space for the selected domain transformations of Office+Caltech10, and $80 - 85$ dimensions for the learned latent space for domain transformations in the Multi-PIE dataset, fewer than 100, as initially provided. It is worth noting that since the number of data points in the $C27, C25$ datasets is much larger than the number of samples in the A, W dataset, we need more latent dimensions for $C27, C25$ than for A, W to capture the variations in these datasets.

4.5. Conclusions

In the experiments conducted, we showed that our approach is able to achieve better performance than the competing methods. Namely, as stated in Sec. 1, our method is expected to bring most benefits in the DA cases when

Table 3. Unsupervised domain adaptation results using SURF features on the Office+Caltech10 dataset with the evaluation setup from [21]. The best and the second best are depicted in bold red and red, respectively.

method	A → W	A → D	A → C	W → A	W → D	W → C	D → A	D → W	D → C	C → A	C → W	C → D	Ave.
1-NN	23.1	22.3	20.0	13.8	40.6	12.2	23.0	51.7	19.9	21.0	19.0	23.6	24.2
SVM	25.6	33.4	35.9	32.1	78.9	25.2	34.6	70.2	31.2	43.8	30.5	40.3	40.1
GFK[17]	35.7	35.1	37.9	35.5	71.2	29.3	36.2	79.1	32.7	40.4	35.8	41.1	42.5
SA[11]	38.6	37.6	35.3	37.4	80.3	32.3	38.0	83.6	32.4	39.0	36.8	39.6	44.2
CORAL[41]	38.7	38.3	**40.3**	37.8	84.9	34.6	38.1	**85.9**	34.2	47.2	39.2	40.7	46.7
ILS[21]	40.6	**41.0**	37.1	39.0	78.7	34.2	38.9	79.1	36.9	48.6	**42.0**	44.1	46.7
PUnDA	**42.5**	40.3	39.5	**42.4**	**85.2**	**36.5**	**40.3**	83.2	**38.9**	**50.1**	41.7	**45.8**	**48.8**

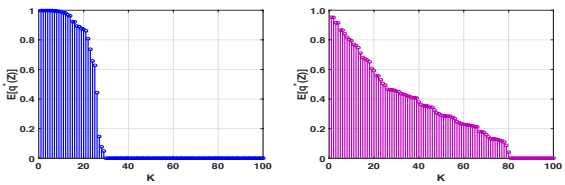


(a) (b) (c)

Figure 6. Feature visualization. The embedding of Multi-PIE $C05$ data using t-sne algorithm [33]. (a) Original features. (b) **PUnDA** features. (c) **ILS** features.

Table 4. Multi-PIE results. The changes in performance w.r.t. the changing face orientations when frontal face images ($C27$) are considered as the source domain. The best and the second best are depicted in bold red and red, respectively.

method	C09	C05	C37	C25	C02	Ave.
1-NN	92.5	55.7	28.5	14.8	11.0	40.5
SVM	87.8	65.0	35.8	15.7	16.7	44.2
GFK[17]	92.5	74.0	32.1	14.1	12.3	45.0
SA[11]	**97.9**	85.9	47.9	16.6	13.9	52.4
CORAL[41]	91.4	74.8	35.3	13.4	13.2	45.6
ILS[21]	96.6	88.3	72.9	28.4	**34.8**	64.2
PUnDA	94.3	**92.2**	**78.8**	**28.9**	34.7	**65.7**



(a) $A \rightarrow W$ (b) $C27 \rightarrow C25$

Figure 7. Inferred $\mathbb{E}[q^*(Z)]$ for the Office+Caltech10 and Multi-PIE datasets.

data in both domains are tightly clustered, with the clusters being geometrically proximal. Indeed, Fig. 6 depicts the embedding of the learned features s/s', and those of **ILS** and the original features x. Colors indicate source (red) and target (blue) domains. Notice that **PUnDA** significantly reduces the domain mismatch, resulting in the expected tight clustering. This is partially due to the use of the proposed probabilistic MMD with Bhattacharyya kernel, which penalizes the domain mismatch while exploiting the uncertainty in the shared feature space - something the **ILS** fails to account for. Further examples are provided in the supplementary material.

In summary, we proposed a novel *probabilistic* approach for unsupervised DA that learns an efficient domain-adaptive classifier that can generalize well on target domains. The key to the proposed approach is that it jointly learns a latent space along with its size, and a softmax classifier, by exploiting both labeled source and unlabeled target data in Bayesian fashion. To tackle the intractability of computing the exact posteriors in our model, we proposed a novel Bayesian approximation to efficiently approximate the target distributions. We showed on two benchmark datasets for image classification, using both hand-crafted and deep-net features, the superiority of the proposed method compared to the state-of-the-art methods for unsupervised domain adaptation of visual domain categories.

5. Acknowledgments

The work of O. Rudovic is funded by the European Union H2020, Marie Curie Action - Individual Fellowship no. 701236 (EngageMe). The work of V. Pavlovic is funded by the National Science Foundation under Grant no. IIS1555408.

References

[1] Y. Aytar and A. Zisserman. Tabula rasa: Model transfer for object category detection. In *ICCV*, pages 2252–2259, 2011. 2

[2] M. Baktashmotlagh, M. Harandi, and M. Salzmann. Distribution-matching embedding for visual domain adaptation. *Journal of Machine Learning Research*, 17(108):1–30, 2016. 1

[3] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann. Unsupervised domain adaptation by domain invariant projection. In *ICCV*, pages 769–776, 2013. 1, 2, 3, 4

[4] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann. Domain adaptation on the statistical manifold. In *CVPR*, pages 2481–2488, 2014. 3

[5] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *ECCV*, pages 404–417, 2006. 5, 6

[6] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):e49–e57, 2006. 2, 4

[7] M. Chen, K. Q. Weinberger, and J. Blitzer. Co-training for domain adaptation. In *NIPS*, pages 2456–2464, 2011. 2

[8] E. Choi and C. Lee. Feature extraction based on the bhattacharyya distance. *Pattern Recognition*, 36(8):1703–1709, 2003. 4

[9] J. Donahue, J. Hoffman, E. Rodner, K. Saenko, and T. Darrell. Semi-supervised domain adaptation with instance constraints. In *CVPR*, pages 668–675, 2013. 1

[10] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, volume 2, pages 524–531, 2005. 3

[11] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *ICCV*, pages 2960–2967, 2013. 1, 5, 7, 8

[12] G. Foster, C. Goutte, and R. Kuhn. Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proceedings of the conference on empirical methods in natural language processing*, pages 451–459, 2010. 2

[13] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. *ICML*, pages 27–35, 2014. 1, 2

[14] Z. Ghahramani, M. J. Beal, et al. Variational inference for bayesian mixtures of factor analysers. In *NIPS*, volume 12, pages 449–455, 1999. 4

[15] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *ECCV*, pages 597–613, 2016. 2

[16] B. Gong, K. Grauman, and F. Sha. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *ICML*, pages 222–230, 2013. 1, 2

[17] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, pages 2066–2073, 2012. 1, 2, 5, 7, 8

[18] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *ICCV*, pages 999–1006, 2011. 1, 2

[19] T. L. Griffiths and Z. Ghahramani. Infinite latent feature models and the indian buffet process. In *NIPS*, volume 18, pages 475–482, 2005. 4

[20] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1

[21] S. Herath, M. Harandi, and F. Porikli. Learning an invariant hilbert space for domain adaptation. *CVPR*, 2017. 3, 5, 6, 7, 8

[22] Y.-H. Hubert Tsai, Y.-R. Yeh, and Y.-C. Frank Wang. Learning cross-domain landmarks for heterogeneous domain adaptation. In *CVPR*, pages 5081–5090, 2016. 1

[23] J. Jiang and C. Zhai. Instance weighting for domain adaptation in nlp. In *ACL*, volume 7, pages 264–271, 2007. 2

[24] W. Jiang, E. Zavesky, S.-F. Chang, and A. Loui. Cross-domain learning methods for high-level visual concept classification. In *ICIP*, pages 161–164, 2008. 2

[25] M. Kan, S. Shan, and X. Chen. Bi-shifting auto-encoder for unsupervised domain adaptation. In *ICCV*, pages 3846–3854, 2015. 2

[26] E. Kodirov, T. Xiang, Z. Fu, and S. Gong. Unsupervised domain adaptation for zero-shot learning. In *ICCV*, pages 2452–2460, 2015. 1

[27] A. Kumar, A. Saha, and H. Daume. Co-regularization based semi-supervised domain adaptation. In *NIPS*, pages 478–486, 2010. 1

[28] W. Li, L. Duan, D. Xu, and I. W. Tsang. Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 36(6):1134–1148, 2014. 1

[29] M. Long, G. Ding, J. Wang, J. Sun, Y. Guo, and P. S. Yu. Transfer sparse coding for robust image representation. In *CVPR*, pages 407–414, 2013. 2, 4

[30] M. Long, J. Wang, Y. Cao, J. Sun, and S. Y. Philip. Deep learning of transferable representation for scalable domain adaptation. *IEEE Transactions on Knowledge and Data Engineering*, 28(8):2027–2040, 2016. 1, 4

[31] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu. Transfer joint matching for unsupervised domain adaptation. In *CVPR*, pages 1410–1417, 2014. 1, 2

[32] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *NIPS*, pages 136–144, 2016. 1

[33] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008. 8

[34] T. Ming Harry Hsu, W. Yu Chen, C.-A. Hou, Y.-H. Hubert Tsai, Y.-R. Yeh, and Y.-C. Frank Wang. Unsupervised domain adaptation with imbalanced cross-domain data. In *ICCV*, pages 4121–4129, 2015. 2

[35] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2011. 2, 4

[36] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa. Visual domain adaptation: A survey of recent advances. *IEEE signal processing magazine*, 32(3):53–69, 2015. 2

[37] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *ECCV*, pages 213–226, 2010. 1, 2

[38] H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000. 1

[39] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5, 6

[40] M. Sugiyama, S. Nakajima, H. Kashima, P. V. Buenau, and M. Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *NIPS*, pages 1433–1440, 2008. 2

[41] B. Sun, J. Feng, and K. Saenko. Return of frustratingly easy domain adaptation. *arXiv preprint arXiv:1511.05547*, 2015. 1, 2, 5, 7, 8

[42] R. Thibaux and M. I. Jordan. Hierarchical beta processes and the indian buffet process. In *AISTATS*, volume 2, pages 564–571, 2007. 4

[43] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *CVPR*, pages 1521–1528, 2011. 1

[44] S. Uguroglu and J. Carbonell. Feature selection for transfer learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 430–442, 2011. 4

[45] M. Xiao and Y. Guo. Feature space independent semi-supervised domain adaptation via kernel matching. *IEEE transactions on pattern analysis and machine intelligence*, 37(1):54–66, 2015. 1

[46] J. Yang, R. Yan, and A. G. Hauptmann. Adapting svm classifiers to data with shifted distributions. In *Seventh IEEE International Conference on Data Mining Workshops*, pages 69–76, 2007. 2

[47] D. Yoo, N. Kim, S. Park, A. S. Paek, and I. S. Kweon. Pixel-level domain transfer. In *ECCV*, pages 517–532, 2016. 1