Unsupervised Domain Adaptation by Domain Invariant Projection

Mahsa Baktashmotlagh[1,3], Mehrtash T. Harandi[2,3], Brian C. Lovell[1], and Mathieu Salzmann[2,3]
[1]University of Queensland [2]Australian National University [3]NICTA, Canberra*
mahsa.baktashmotlagh@nicta.com.au

Abstract

Domain-invariant representations are key to addressing the domain shift problem where the training and test examples follow different distributions. Existing techniques that have attempted to match the distributions of the source and target domains typically compare these distributions in the original feature space. This space, however, may not be directly suitable for such a comparison, since some of the features may have been distorted by the domain shift, or may be domain specific. In this paper, we introduce a Domain Invariant Projection approach: An unsupervised domain adaptation method that overcomes this issue by extracting the information that is invariant across the source and target domains. More specifically, we learn a projection of the data to a low-dimensional latent space where the distance between the empirical distributions of the source and target examples is minimized. We demonstrate the effectiveness of our approach on the task of visual object recognition and show that it outperforms state-of-the-art methods on a standard domain adaptation benchmark dataset.

1. Introduction

Domain shift is a fundamental problem in visual recognition tasks as evidenced by the recent surge of interest in domain adaptation [22, 15, 16]. The problem typically arises when the training (source) and test (target) examples follow different distributions. This is a common scenario in modern visual recognition tasks, especially if images are acquired with different cameras, or in very different conditions (*e.g.*, commercial website versus home environment, images taken under different illuminations). Failing to model the distribution shift in the hope that the image features will be robust enough often yields poor recognition accuracy [26, 16, 15, 14]. On the other hand, labeling sufficiently many images from the target domain to train a discriminative classifier specific to this domain is prohibitively time-consuming and impractical in realistic scenarios.

To relate the source and target domains, several state-of-the-art methods have proposed to create intermediate representations [15, 16]. However, these representations do not explicitly try to match the probability distributions of the source and target data, which may make them sub-optimal for classification. Sample selection, or re-weighting, approaches [14, 21] explicitly attempt to match the source and target distributions by finding the most appropriate source examples for the target data. However, they fail to account for the fact that the image features themselves may have been distorted by the domain shift, and that some of the image features may be specific to one domain and thus irrelevant for classification in the other one.

In light of the above discussion, we propose to tackle the problem of domain shift by extracting the information that is invariant across the source and target domains. To this end, we introduce a Domain Invariant Projection (DIP) approach, which aims to learn a low-dimensional latent space where the source and target distributions are similar. Learning such a projection allows us to account for the potential distortions induced by the domain shift, as well as for the presence of domain-specific image features. Furthermore, since the distributions of the source and target data in the latent space are similar, we expect a classifier trained on the source examples to perform well on the target domain.

In this work, we make use of the Maximum Mean Discrepancy (MMD) [17] to measure the dissimilarity between the empirical distributions of the source and target examples. Learning the latent space that minimizes the MMD between the source and target domains can then be formulated as an optimization problem on a Grassmann manifold. This lets us utilize Grassmannian geometry to effectively obtain our domain invariant projection. Although designed to be fully unsupervised, our formalism naturally allows us to exploit label information from either domain during the training process. While not strictly necessary, this information can help boosting classification accuracy even further.

In short, we introduce the idea of finding a domain invariant representation of the data by matching the source and target distributions in a low-dimensional latent space, and propose an effective algorithm to learn our Domain In-

*NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the ARC through the ICT Centre of Excellence program.

variant Projection. We demonstrate the benefits of our approach on the task of visual object recognition and show that it outperforms state-of-the-art methods on the standard domain adaptation benchmark dataset [26].

2. Related Work

Existing domain adaptation methods can be divided into two categories: Semi-supervised approaches [12, 3, 26] that assume that a small number of labeled examples from the target domain are available during training, and unsupervised approaches [15, 14, 16, 21] that do not require any labels from the target domain.

In the former category, modifications of Support Vector Machines (SVM) [12, 3] and other statistical classifiers [10] have been proposed to exploit the availability of labeled and unlabeled data from the target domain. Co-regularization of similar classifiers was also introduced to utilize unlabeled target data during training [9]. For visual recognition, metric learning [26] and transformation learning [23] were shown to be effective at making use of the labeled target examples. Furthermore, semi-supervised methods have also been proposed to tackle the case where multiple source domains are available [11, 20]. While semi-supervised methods are often effective, in many applications, labeled target examples are not available and cannot easily be acquired.

To address this issue, unsupervised domain adaptation approaches that rely on purely unsupervised target data have been proposed [28, 7, 8]. In particular, two types of methods have proven quite successful at the task of visual object recognition: Subspace-based approaches and sample re-weighting approaches.

Subspace-based approaches [4, 16, 15] model the domain shift by representing the data with multiple subspaces. In particular, in [4], coupled subspaces are learned using Canonical Correlation Analysis (CCA). Rather than limiting the representation to one source and one target subspaces, several techniques exploit intermediate subspaces, which link the source data to the target data. This idea was originally introduced in [16], where the subspaces were modeled as points on a Grassmann manifold, and intermediate subspaces were obtained by sampling points along the geodesic between the source and target subspaces. This method was extended in [15], which showed that all intermediate subspaces could be taken into account by integrating along the geodesic. While this formulation nicely characterizes the change between the source and target data, it is not clear why all the subspaces along this path should yield meaningful representations. More importantly, these subspace-based methods do not explicitly exploit the statistical properties of the observed data.

In contrast, sample re-weighting, or selection, approaches, have focused more directly on comparing the distributions of the source and target data. In particular, in [21, 18], the source examples are re-weighted so as to minimize the MMD between the source and target distributions. More recently, an approach to selecting landmarks among the source examples based on MMD was introduced [14]. This sample selection approach was shown to be very effective, especially for the task of visual object recognition, to the point that it outperforms state-of-the-art semi-supervised approaches. Despite their success, it is important to note that sample re-weighting and selection methods compare the source and target distributions directly in the original feature space. This space, however, may not be appropriate for this task, since the image features may have been distorted by the domain shift, and since some of the features may only be relevant to one specific domain. In contrast, in this work, we compare the source and target distributions in a low-dimensional latent space where these effects are removed, or reduced. This, in turn, yields a representation that significantly outperforms the recent landmark-based approach [14], as well as other state-of-the-art methods on the task of object recognition.

Transfer Component Analysis (TCA) [24] may be closest in spirit to our work. However, although motivated by MMD, in TCA, the distance between the sample means is measured in a lower-dimensional space rather than in Reproducing Kernel Hilbert Space (RKHS), which somewhat contradicts the intuition behind the use of kernels. Here, we follow the more intuitive idea of comparing the distributions of the transformed data using MMD. This, we believe and as suggested by our experiments, makes better use of the expressive power of the kernel in MMD.

3. Background

In this section, we review some concepts that will be used in our algorithm. In particular, we briefly discuss the idea of Maximum Mean Discrepancy and introduce some notions of Grassmann manifolds.

3.1. Maximum Mean Discrepancy

In this work, we are interested in measuring the dissimilarity between two probability distributions s and t. Rather than restricting these distributions to take a specific parametric form, we opt for a non-parametric approach to compare s and t. Non-parametric representations are very well-suited to visual data, which typically exhibits complex probability distributions in high-dimensional spaces.

We employ the maximum mean discrepancy [17] between two distributions s and t to measure their dissimilarity. The MMD is an effective non-parametric criterion that compares the distributions of two sets of data by mapping the data to RKHS. Given two distributions s and t, the MMD between s and t is defined as

$$D'(\mathfrak{F}, s, t) = \sup_{f \in \mathfrak{F}} \left(E_{\tilde{\boldsymbol{x}}_s \sim s}[f(\tilde{\boldsymbol{x}}_s)] - E_{\tilde{\boldsymbol{x}}_t \sim t}[f(\tilde{\boldsymbol{x}}_t)] \right) \ ,$$

where $E_{\tilde{\boldsymbol{x}} \sim s}[\cdot]$ is the expectation under distribution s. By defining $\mathfrak{F}$ as the set of functions in the unit ball in a universal RKHS $\mathcal{H}$, it was shown that $D'(\mathfrak{F}, s, t) = 0$ if and only if $s = t$ [17].

Let $\tilde{\boldsymbol{X}}_{\boldsymbol{s}} = \{\tilde{\boldsymbol{x}}_s^1, \cdots, \tilde{\boldsymbol{x}}_s^n\}$ and $\tilde{\boldsymbol{X}}_{\boldsymbol{t}} = \{\tilde{\boldsymbol{x}}_t^1, \cdots, \tilde{\boldsymbol{x}}_t^m\}$ be two sets of observations drawn i.i.d. from s and t, respectively. An empirical estimate of the MMD can be computed as

$$D(\tilde{\boldsymbol{X}}_s, \tilde{\boldsymbol{X}}_t) = \left\| \frac{1}{n} \sum_{i=1}^{n} \phi(\tilde{\boldsymbol{x}}_s^i) - \frac{1}{m} \sum_{j=1}^{m} \phi(\tilde{\boldsymbol{x}}_t^j) \right\|_{\mathcal{H}}$$

$$= \left(\sum_{i,j=1}^{n} \frac{k(\tilde{\boldsymbol{x}}_s^i, \tilde{\boldsymbol{x}}_s^j)}{n^2} + \sum_{i,j=1}^{m} \frac{k(\tilde{\boldsymbol{x}}_t^i, \tilde{\boldsymbol{x}}_t^j)}{m^2} - 2 \sum_{i,j=1}^{n,m} \frac{k(\tilde{\boldsymbol{x}}_s^i, \tilde{\boldsymbol{x}}_t^j)}{nm} \right)^{\frac{1}{2}},$$

where $\phi(\cdot)$ is the mapping to the RKHS $\mathcal{H}$, and $k(\cdot, \cdot) = \langle \phi(\cdot), \phi(\cdot) \rangle$ is the universal kernel associated with this mapping. In short, the MMD between the distributions of two sets of observations is equivalent to the distance between the sample means in a high-dimensional feature space.

3.2. Grassmann Manifolds

In our formulation, we model the projection of the source and target data to a low-dimensional space as a point $\boldsymbol{W}$ on a Grassmann manifold $\mathcal{G}(d, D)$. The Grassmann manifold $\mathcal{G}(d, D)$ consists of the set of all linear d-dimensional subspaces of $\mathbb{R}^D$. In particular, this lets us handle constraints of the form $\boldsymbol{W}^T \boldsymbol{W} = \boldsymbol{I}_d$. Learning the projection then involves non-linear optimization on the Grassmann manifold, which requires some notions of differential geometry reviewed below.

In differential geometry, the shortest path between two points on a manifold is a curve called a *geodesic*. The *tangent space* at a point on a manifold is a vector space that consists of the tangent vectors of all possible curves passing through this point. *Parallel transport* is the action of transferring a tangent vector between two points on a manifold. Unlike in flat spaces, this cannot be achieved by simple translation, but requires subtracting a normal component at the end point [13].

On a Grassmann manifold, the above-mentioned operations have efficient numerical forms and can thus be used to perform optimization on the manifold. In particular, we make use of a conjugate gradient (CG) algorithm on the Grassmann manifold [13]. CG techniques are popular non-linear optimization methods with fast convergence rates. These methods iteratively optimize the objective function in linearly independent directions called conjugate directions [25]. CG on a Grassmann manifold can be summarized by the following steps:

(i) Compute the gradient $\nabla f_{\boldsymbol{W}}$ of the objective function f on the manifold at the current estimate $\boldsymbol{W}$ as

$$\nabla f_{\boldsymbol{W}} = \partial f_{\boldsymbol{W}} - \boldsymbol{W}\boldsymbol{W}^T \partial f_{\boldsymbol{W}} , \tag{1}$$

with $\partial f_{\boldsymbol{W}}$ the matrix of usual partial derivatives.

(ii) Determine the search direction $\boldsymbol{H}$ by parallel transporting the previous search direction and combining it with $\nabla f_{\boldsymbol{W}}$.

(iii) Perform a line search along the geodesic at $\boldsymbol{W}$ in the direction $\boldsymbol{H}$.

These steps are repeated until convergence to a local minimum, or until a maximum number of iterations is reached.

4. Domain Invariant Projection (DIP)

In this section, we introduce our approach to unsupervised domain adaptation. We first derive the optimization problem at the heart of our approach, and then discuss the details of our Grassmann manifold optimization method.

4.1. Problem Formulation

Our goal is to find a representation of the data that is invariant across different domains. Intuitively, with such a representation, a classifier trained on the source domain should perform equally well on the target domain. To achieve invariance, we search for a projection to a low-dimensional subspace where the source and target distributions are similar, or, in other words, a projection that minimizes a distance measure between the two distributions.

More specifically, let $\boldsymbol{X}_{\boldsymbol{s}} = \left[\boldsymbol{x}_s^1, \cdots, \boldsymbol{x}_s^n\right]$ be the $D \times n$ matrix containing n samples from the source domain and $\boldsymbol{X}_{\boldsymbol{t}} = \left[\boldsymbol{x}_t^1, \cdots, \boldsymbol{x}_t^m\right]$ be the $D \times m$ matrix containing m samples from the target domain. We search for a $D \times d$ projection matrix $\boldsymbol{W}$, such that the distributions of the source and target samples in the resulting d-dimensional subspace are as similar as possible. In particular, we measure the distance between these two distribution with the MMD discussed in Section 3.1. This distance can be expressed as

$$D(\boldsymbol{W}^T\boldsymbol{X}_{\boldsymbol{s}}, \boldsymbol{W}^T\boldsymbol{X}_{\boldsymbol{t}}) = \left\| \frac{1}{n} \sum_{i=1}^{n} \phi(\boldsymbol{W}^T\boldsymbol{x}_s^i) - \frac{1}{m} \sum_{j=1}^{m} \phi(\boldsymbol{W}^T\boldsymbol{x}_t^j) \right\|_{\mathcal{H}}, \tag{2}$$

with $\phi(\cdot)$ the mapping from $\mathbb{R}^D$ to the high-dimensional RKHS $\mathcal{H}$. Note that, here, $\boldsymbol{W}$ appears inside $\phi(\cdot)$ in order to measure the MMD of the projected samples. This is in contrast with sample re-weighting, or selection methods [21, 18, 14, 24] that place weights outside $\phi(\cdot)$. Therefore, these methods ultimately still compare the distributions in the original image feature space and may suffer from the presence of domain-specific features.

Using the MMD, learning $\boldsymbol{W}$ can be expressed as the optimization problem

$$\begin{aligned} \boldsymbol{W}^* \quad &= \quad \underset{\boldsymbol{W}}{\operatorname{argmin}} \; D^2(\boldsymbol{W}^T\boldsymbol{X}_{\boldsymbol{s}}, \boldsymbol{W}^T\boldsymbol{X}_{\boldsymbol{t}}) \\ &\text{s.t.} \quad \boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}_d \, , \end{aligned} \tag{3}$$

where the constraints enforce $\boldsymbol{W}$ to be orthogonal. Such constraints prevent our model from wrongly matching the two distributions by distorting the data, and make it very unlikely that the resulting subspace only contains the noise of both domains. Orthogonality constraints have proven effective in many subspace methods, such as PCA or CCA.

As shown in Section 3.1, the MMD in the RKHS $\mathcal{H}$ can be expressed in terms of a kernel function $k(\cdot,\cdot)$. In particular here, we exploit the Gaussian kernel function, which is known to be universal [27]. This lets us rewrite our objective function as

$$\begin{aligned} D^2(\boldsymbol{W}^T\boldsymbol{X_s}, \boldsymbol{W}^T\boldsymbol{X_t}) &= \\ &\frac{1}{n^2}\sum_{i,j=1}^{n} \exp\left(-\frac{(\boldsymbol{x}_s^i - \boldsymbol{x}_s^j)^T\boldsymbol{W}\boldsymbol{W}^T(\boldsymbol{x}_s^i - \boldsymbol{x}_s^j)}{\sigma}\right) \\ &+\frac{1}{m^2}\sum_{i,j=1}^{m} \exp\left(-\frac{(\boldsymbol{x}_t^i - \boldsymbol{x}_t^j)^T\boldsymbol{W}\boldsymbol{W}^T(\boldsymbol{x}_t^i - \boldsymbol{x}_t^j)}{\sigma}\right) \\ &-\frac{2}{mn}\sum_{i,j=1}^{n,m} \exp\left(-\frac{(\boldsymbol{x}_s^i - \boldsymbol{x}_t^j)^T\boldsymbol{W}\boldsymbol{W}^T(\boldsymbol{x}_s^i - \boldsymbol{x}_t^j)}{\sigma}\right) . \end{aligned} \tag{4}$$

Since the Gaussian kernel satisfies the universality condition of the MMD, it is a natural choice for our approach. However, it was shown that, in practice, choices of non-universal kernels may be more appropriate to measure the MMD [6]. In particular, the more general class of characteristic kernels can also be employed. This class incorporates all strictly positive definite kernels, such as the well-known polynomial kernel. Therefore, here, we also consider using the polynomial kernel of degree two. The fact that this kernel yields a distribution distance that only compares the first and second moment of the two distributions [17] will be shown to have little impact on our experimental results, thus showing the robustness of our approach to the choice of kernel. Replacing the Gaussian kernel with this polynomial kernel in our objective function yields

$$\begin{aligned} D^2(\boldsymbol{W}^T\boldsymbol{X_s}, \boldsymbol{W}^T\boldsymbol{X_t}) &= \\ &\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}(1+{\boldsymbol{x}_s^i}^T\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{x}_s^j)^2 \\ &+\frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}(1+{\boldsymbol{x}_t^i}^T\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{x}_t^j)^2 \\ &-\frac{2}{mn}\sum_{i=1}^{n}\sum_{j=1}^{m}(1+{\boldsymbol{x}_s^i}^T\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{x}_t^j)^2. \end{aligned} \tag{5}$$

The two definitions of MMD introduced in Eqs. 4 and 5 can be computed efficiently in matrix form as

$$D^2(\boldsymbol{W}^T\boldsymbol{X_s}, \boldsymbol{W}^T\boldsymbol{X_t}) = Tr(\boldsymbol{K_W}\boldsymbol{L}) \ , \tag{6}$$

where

$$\begin{aligned} \boldsymbol{K_W} &= \begin{bmatrix} \boldsymbol{K}_{s,s} & \boldsymbol{K}_{s,t} \\ \boldsymbol{K}_{t,s} & \boldsymbol{K}_{t,t} \end{bmatrix} \in \mathbb{R}^{(n+m)\times(n+m)} \ , \text{ and} \\ L_{ij} &= \begin{cases} 1/n^2 & i,j \in \mathcal{S} \\ 1/m^2 & i,j \in \mathcal{T} \\ -1/(nm) & \text{otherwise} \end{cases} , \end{aligned}$$

with $\mathcal{S}$ and $\mathcal{T}$ the sets of source and target indices, respectively. Each element in $\boldsymbol{K_W}$ is computed using the kernel function (either Gaussian, or polynomial), and thus depends on $\boldsymbol{W}$. Note that, with both kernels, $\boldsymbol{K_W}$ can be computed efficiently in matrix form (*i.e.*, without looping over its elements). This yields the optimization problem

$$\begin{aligned} \boldsymbol{W}^* &= \operatorname*{argmin}_{\boldsymbol{W}} \ Tr\left(\boldsymbol{K_W}\boldsymbol{L}\right) \\ &\quad \text{s.t. } \boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}_d \ , \end{aligned} \tag{7}$$

which is a nonlinear constrained problem. In practice, we represent $\boldsymbol{W}$ as a point on a Grassmann manifold, which yields an unconstrained optimization problem on the manifold. As mentioned in Section 3.2, we make use of a conjugate gradient method on the manifold to obtain $\boldsymbol{W}^*$.

4.1.1 Encouraging Class Clustering (DIP-CC)

In the DIP formulation described above, learning the projection $\boldsymbol{W}$ is done in a fully unsupervised manner. Note, however, that even in the so-called unsupervised setting, domain adaptation methods have access to the labels of the source examples. Here, we show that our formulation naturally allows us to exploit these labels while learning the projection.

Intuitively, we are interested in finding a projection that not only minimizes the distance between the distribution of the projected source and target data, but also yields good classification performance. To this end, we search for a projection that encourages samples with the same labels to form a more compact cluster. This can be achieved by minimizing the distance between the projected samples of each class and their mean. This yields the optimization problem

$$\begin{aligned} \boldsymbol{W}^* = &\operatorname*{argmin}_{\boldsymbol{W}} \ Tr(\boldsymbol{K_W}\boldsymbol{L}) + \lambda\sum_{c=1}^{C}\sum_{i=1}^{n_c}\left\|\boldsymbol{W}^T(\boldsymbol{x}_s^{i,c} - \boldsymbol{\mu}_c)\right\|^2 \\ &\text{s.t.} \quad \boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I} \ , \end{aligned} \tag{8}$$

where C is the number of classes, n_c the number of examples in class c, $\boldsymbol{x}_s^{i,c}$ denotes the i^{th} example of class c, and $\boldsymbol{\mu}_c$ the mean of the examples in class c. Note that in our formulation, the mean of the projected examples is equivalent to the projection of the mean. Note also that the regularizer in Eq. 8 is related to the intra-class scatter in the objective function of Linear Discriminant Analysis (LDA). While we also tried to incorporate the other LDA term, which encourages the means of different classes to be spread apart, we found no benefits in doing so in our results.

4.1.2 Semi-Supervised DIP (SS-DIP)

The formulations of DIP given in Eqs. 7 and 8 fall into the unsupervised domain adaptation category, since they do not exploit any labeled target examples. However, our formulation can very naturally be extended to the semi-supervised settings. To this end, it must first be noted that, after learning $\boldsymbol{W}$, we train a classifier in the resulting latent space (*i.e.*, on $\boldsymbol{W}^{*T}\boldsymbol{x}$). In the unsupervised setting, this classifier is only trained using the source examples.

With Semi-Supervised DIP (SS-DIP), the labeled target examples can be taken into account in two different manners. In the unregularized formulation of Eq. 7, since no labels are used when learning $\boldsymbol{W}$, we only employ the labeled target examples along with the source ones to train the final classifier. With the class-clustering regularizer of Eq. 8, we utilize the target labels in the regularizer when learning $\boldsymbol{W}$, as well as when learning the final classifier.

4.2. Optimization on a Grassmann Manifold

All versions of our DIP formulation yield nonlinear, constrained optimization problems. To tackle this challenging scenario, we first note that the constraints on $\boldsymbol{W}$ make it a point on a Grassmann manifold. This lets us rewrite our constrained optimization problem as an unconstrained problem on the manifold $\mathcal{G}(d, D)$. Optimization on Grassmann manifolds has proven effective at avoiding bad local minima [1]. More specifically, manifold optimization methods often have better convergence behavior than iterative projection methods, which can be crucial with a nonlinear objective function [1].

While our optimization problem has become unconstrained, it remains nonlinear. To effectively address this, we make use of a conjugate gradient method on the manifold. Recall from Section 3.2 that CG on a Grassmann manifold involves **(i)** computing the gradient on the manifold $\nabla f_{\boldsymbol{W}}$, **(ii)** estimating the search direction $\boldsymbol{H}$, and **(iii)** performing a line search along a geodesic. Eq. 1 shows that the gradient on the manifold depends on the partial derivatives of the objective function w.r.t. $\boldsymbol{W}$, *i.e.*, $\partial f / \partial \boldsymbol{W}$. The general form of $\partial f / \partial \boldsymbol{W}$ in our formulation is

$$\frac{\partial f}{\partial \boldsymbol{W}} = \sum_{i,j=1}^{n} \frac{\boldsymbol{G}_{ss}(i,j)}{n^2} + \sum_{i,j=1}^{m} \frac{\boldsymbol{G}_{tt}(i,j)}{m^2} - 2 \sum_{i,j=1}^{n,m} \frac{\boldsymbol{G}_{st}(i,j)}{mn} ,$$

where $\boldsymbol{G}_{ss}(\cdot,\cdot)$, $\boldsymbol{G}_{tt}(\cdot,\cdot)$ and $\boldsymbol{G}_{st}(\cdot,\cdot)$ are matrices of size $D \times d$. With the definition of MMD in Eq. 4 based on the Gaussian kernel $k_G(\cdot,\cdot)$, the matrix, e.g., $\boldsymbol{G}_{ss}(i,j)$ takes the form

$$\boldsymbol{G}_{ss}(i,j) = -\frac{2}{\sigma} k_G(\boldsymbol{x}_s^i, \boldsymbol{x}_s^j)(\boldsymbol{x}_s^i - \boldsymbol{x}_s^j)(\boldsymbol{x}_s^i - \boldsymbol{x}_s^j)^T \boldsymbol{W} ,$$

and similarly for $\boldsymbol{G}_{tt}(\cdot,\cdot)$ and $\boldsymbol{G}_{st}(\cdot,\cdot)$. With the MMD of Eq. 5 based on the degree 2 polynomial kernel $k_P(\cdot,\cdot)$, $\boldsymbol{G}_{ss}(i,j)$ becomes

$$\boldsymbol{G}_{ss}(i,j) = 2k_P(\boldsymbol{x}_s^i, \boldsymbol{x}_s^j)(\boldsymbol{x}_s^i \boldsymbol{x}_s^{j\,T} + \boldsymbol{x}_s^j \boldsymbol{x}_s^{i\,T})\boldsymbol{W} ,$$

and similarly for $\boldsymbol{G}_{tt}(\cdot,\cdot)$ and $\boldsymbol{G}_{st}(\cdot,\cdot)$. As f itself, $\partial f / \partial \boldsymbol{W}$ can be efficiently computed in matrix form.

In our experiments, we first applied PCA to the concatenated source and target data, kept all the data variance, and initialized $\boldsymbol{W}$ to the truncated identity matrix. We observed that learning $\boldsymbol{W}$ typically converges in only a few iterations.

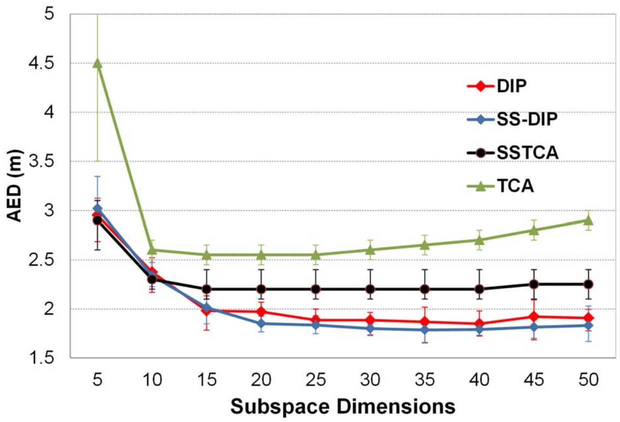

Figure 1. Comparison of our approach with TCA on the task of indoor WiFi localization.

5. Experiments

We evaluated our approach on the tasks of indoor WiFi localization and visual object recognition, and compare its performance against the state-of-the art methods in each task. In all our experiments, we set the variance σ of the Gaussian kernel to the median squared distance between all source examples, and the weight λ of the regularizer to $4/\sigma$ when using the regularizer.

5.1. Cross-domain WiFi Localization

We first evaluated our approach on the task of indoor WiFi localization using the public wifi data set published in the 2007 IEEE ICDM Contest for domain adaptation [29]. The goal of indoor WiFi localization is to predict the location (labels) of WiFi devices based on received signal strength (RSS) values collected during different time periods (domains). The dataset contains 621 labeled examples collected during time period A (*i.e.*, source) and 3128 unlabeled examples collected during time period B (*i.e.*, target).

We followed the transductive evaluation setting introduced in [24] to compare our DIP methods with TCA and SSTCA, which are considered state-of-the-art on this dataset. Nearest-neighbor was employed as the final classifier for our algorithms and for the baselines. In our experiments, we used all the source data and 400 randomly sampled target examples. In Fig. 1, we report the mean Average

Method	$A \rightarrow C$	$A \rightarrow D$	$A \rightarrow W$	$C \rightarrow A$	$C \rightarrow D$	$C \rightarrow W$	$W \rightarrow A$	$W \rightarrow C$	$W \rightarrow D$
NO ADAPT-1NN	26	25.5	29.8	23.7	25.5	25.8	23	20	59.2
NO ADAPT-SVM	41.7	41.4	34.2	51.8	54.1	46.8	31.1	31.5	70.7
TCA[24]	35.0	36.3	27.8	41.4	45.2	32.5	24.2	22.5	80.2
GFK[15]	42.2	42.7	40.7	44.5	43.3	44.7	31.8	30.8	75.6
SCL[5]	42.3	36.9	34.9	49.3	42.0	39.3	34.7	32.5	83.4
KMM[18]	42.2	42.7	42.4	48.3	53.5	45.8	31.9	29.0	72.0
LM[14]	45.5	47.1	46.1	56.7	57.3	49.5	40.2	35.4	75.2
DIP	**47.4**	**50.3**	47.5	55.7	60.5	**58.3**	**42.6**	34.2	88.5
DIP-CC	47.2	49.04	**47.8**	**58.7**	**61.2**	58	40.9	**37.2**	**91.7**
DIP(Poly)	47.3	49.1	45.1	56.1	58.6	57	42.8	36.5	89.8
DIP-CC(Poly)	47.4	48.4	46.1	56.4	58.6	58	42.7	36.5	89.8

Table 1. Recognition accuracies on 9 pairs of source/target domains using the evaluation protocol of [14]. C**: Caltech,** A**: Amazon,** W**: Webcam,** D**: DSLR**.

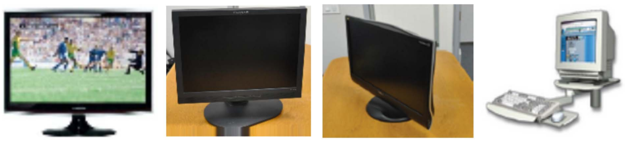

Figure 2. Sample images from the ***monitor*** *category. From left to right:* ***Amazon***, ***Webcam***, ***DSLR***, *and* ***Caltech***.

Error Distance (AED) over 10 different random samples for different subspace dimensionalities. $AED = \frac{\sum_i l(\boldsymbol{x}_i) - y_i}{N}$ where $\boldsymbol{x}_i$ is a vector of RSS values, $l(\boldsymbol{x}_i)$ is the predicted location and y_i is the corresponding ground truth location. Note that our algorithms outperform TCA in both unsupervised and supervised settings.

5.2. Visual Object Recognition

We then evaluated our approach on the task of visual object recognition using the benchmark domain adaptation dataset introduced in [26]. This dataset contains images from four different domains: Amazon, DSLR, Webcam, and Caltech. The Amazon domain consists of images acquired in a highly-controlled environment with studio lighting conditions. These images capture the large intra-class variations of 31 classes, but typically show the objects only from one canonical viewpoint. The DSLR domain consists of high resolution images of 31 categories that are taken with a digital SLR camera in a home environment under natural lighting. The Webcam images were acquired in a similar environment as the DSLR ones, but have much lower resolution and contain significant noise, as well as color and white balance artifacts. The last domain, Caltech [19], consists of images of 256 object classes downloaded from Google images. Following [15], we use the 10 object classes common to all four datasets. This yields 2533 images in total, with 8 to 151 images per category per domain. Fig. 2 depicts sample images from the four domains.

For our evaluation, we used the features provided by [15], which were obtained using the protocol described in [26]. More specifically, all images were converted to grayscale and resized to have the same width. Local scale-invariant interest points were detected by the SURF detector [2], and a 64-dimensional rotation invariant SURF descriptor was extracted from the image patch around each interest point. A codebook of size 800 was then generated from a subset of the Amazon dataset using k-means clustering on the SURF descriptors. The final feature vector for each image is the normalized histogram of visual words obtained from this codebook.

In all our experiments, we used the subspace disagreement measure of [15] to automatically determine the dimensionality of the projection matrix $\boldsymbol{W}$. For recognition, we trained an SVM classifier with a polynomial kernel of degree 2 on the projected source examples. Our results are presented as DIP for the original model and DIP-CC for the class-clustering regularized one.

In a first experiment on this dataset, we used the evaluation protocol introduced in [14]: For each source/target pair, all the available examples in both domains are exploited at once, rather than splitting the datasets into multiple training/testing partitions.[1] This protocol was motivated by the fact that, in [14], selecting landmarks requires a sufficient number of source examples to be available. For the same reason, the DSLR dataset is never used as source domain, since it contains too few examples per class. We compare our DIP and DIP-CC results, with Gaussian or polynomial kernel in MMD, with those obtained by several state-of-the-art methods: transfer component analysis (TCA) [24], geodesic flow kernel (GFK) [15], geodesic flow sampling (GFS) [16], structural correspondence learning (SCL) [5], kernel mean matching (KMM) [18] and landmark selection (LM) [14]. Table 1 shows the recognition accuracies on the target examples for the 9 pairs of source and target domains. For this protocol, our method (with and without class-clustering regularizer) outperforms the state-of-the-art techniques in all cases. Note that, in this case, our class-clustering regularizer is not crucial to achieve good accu-

[1] This evaluation protocol was explained to us by the authors of [14].

Method	$A \rightarrow C$	$A \rightarrow D$	$A \rightarrow W$	$C \rightarrow A$	$C \rightarrow D$	$C \rightarrow W$
NO ADAPT-1NN	22.6 ± 0.3	22.2 ± 0.4	23.5 ± 0.6	20.8 ± 0.4	22 ± 0.6	19.4 ± 0.7
NO ADAPT-SVM	38.7 ± 1.6	36.7 ± 2.3	37.2 ± 2.8	44.3 ± 2.4	41.1 ± 3.9	39.9 ± 3.2
GFS[16]	35.6 ± 0.4	34.9 ± 0.9	34.4 ± 0.9	36.9 ± 0.5	35.2 ± 1	33.9 ± 1.2
GFK-1NN[15]	37.9 ± 0.4	35.2 ± 0.9	35.7 ± 0.9	40.4 ± 0.7	41.1 ± 1.3	35.8 ± 1
GFK-SVM[15]	39 ± 1.7	34.1 ± 2.6	40.7 ± 3.7	47.2 ± 2.3	38.5 ± 2.7	38.8 ± 3.2
TCA[24]	40 ± 1.3	39.1 ± 1.5	40.1 ± 1.2	46.7 ± 1.1	41.4 ± 1.2	36.2 ± 1.0
DIP	$\mathbf{43.3 \pm 1.4}$	42.8 ± 2.5	46.7 ± 2.7	50 ± 3.2	49 ± 2.9	47.6 ± 3.5
DIP-CC	43.2 ± 2.8	$\mathbf{43.3 \pm 3.3}$	$\mathbf{47.8 \pm 4.8}$	$\mathbf{51.8 \pm 2.6}$	$\mathbf{51.4 \pm 4.1}$	$\mathbf{47.7 \pm 4.4}$

Table 2. Recognition accuracies on 6 pairs of source/target domains using the evaluation protocol of [26]. C**: Caltech,** A**: Amazon,** W**: Webcam,** D**: DSLR**.

Method	$D \rightarrow A$	$D \rightarrow C$	$D \rightarrow W$	$W \rightarrow A$	$W \rightarrow C$	$W \rightarrow D$
NO ADAPT-1NN	27.7 ± 0.4	24.8 ± 0.4	53.1 ± 0.6	20.7 ± 0.6	16.1 ± 0.4	37.3 ± 1.2
NO ADAPT-SVM	33.6 ± 1.7	31.1 ± 0.9	75.2 ± 2.6	36.9 ± 1.2	33.4 ± 1.1	80.2 ± 2.5
GFS[16]	32.6 ± 0.5	30 ± 0.2	74.9 ± 0.6	31.3 ± 0.7	27.3 ± 0.5	70.7 ± 0.9
GFK-1NN [15]	36.2 ± 0.4	32.7 ± 0.4	79.1 ± 0.7	35.5 ± 0.7	29.3 ± 0.4	71.2 ± 0.9
GFK-SVM [15]	39 ± 1.1	34.5 ± 0.8	76.2 ± 1.2	40.8 ± 1.2	36.1 ± 0.9	72.4 ± 2.2
TCA[24]	39.6 ± 1.2	34 ± 1.1	80.4 ± 2.6	40.2 ± 1.1	33.7 ± 1.1	77.5 ± 2.5
DIP	40.5 ± 1	$\mathbf{39 \pm 0.5}$	$\mathbf{86.7 \pm 1.2}$	$\mathbf{42.5 \pm 1.5}$	37 ± 0.9	$\mathbf{86.4 \pm 1.8}$
DIP-CC	$\mathbf{41 \pm 0.9}$	35.8 ± 0.6	84.02 ± 0.9	41.1 ± 1.1	$\mathbf{37.1 \pm 0.9}$	85.3 ± 2.5

Table 3. Recognition accuracies on the remaining 6 pairs of source/target domains using the evaluation protocol of [26]. C**: Caltech,** A**: Amazon,** W**: Webcam,** D**: DSLR**.

racy. Note also that our approach is robust to the choice of kernel used in MMD. Therefore, in the remaining experiments, we only report results with the Gaussian kernel.

In a second experiment, we used the more conventional evaluation protocol introduced in [26], which consists of splitting the data into multiple partitions. For each source/target pair, we report the average recognition accuracy and standard deviation over the 20 partitions provided with GFK[2]. With this protocol, all possible combinations of source and target domains were evaluated. In Tables 2 and 3, we compare our results with GFK and the baseline results reported in [15]. As before, both our DIP and DIP-CC approaches consistently outperform the baselines.

Finally, we evaluated our approach in the semi-supervised setting. Following the evaluation protocol of [26], we made use of 3 labeled samples per category from the target domain. We compare our method against the state-of-the-art semi-supervised GFK approach of [15] and metric learning approach of [26]. Table 4 and 5 show the results of all methods on all pairs of domains. Similarly as in the unsupervised scenario, our semi-supervised DIP-CC and DIP approaches achieve the highest accuracies. Here however, the class-clustering regularizer boosts the accuracy more consistently, which suggests the importance of such a regularizer in the presence of small amounts of labeled data.

6. Conclusion and Future Work

In this paper, we have introduced an approach to unsupervised domain adaptation that focuses on extracting a domain-invariant representation of the source and target data. To this end, we have proposed to match the source and target distributions in a low-dimensional latent space, rather than in the original feature space. Our experiments have evidenced the importance of exploiting distribution invariance for domain adaptation by revealing that our DIP approach consistently outperformed the state-of-the-art methods in the task of visual object recognition. A current limitation of our approach is the non-convexity of the resulting optimization problem. Although, in practice, optimization on the Grassmann manifold has proven well-behaved, we intend to study if the use of other characteristic kernels in conjunction with different optimization strategies, such as the convex-concave procedure, could yield theoretical convergence guarantees within our formalism. The use of a nonlinear mapping would intuitively also seem more effective than our current linear transformation. However, it is unclear how to regularize nonlinear transformations to prevent them from deteriorating the data distribution to the point of making two inherently dissimilar distributions similar. Finally, we also plan to investigate how ideas from the deep learning literature could be employed to obtain domain invariant features.

[2] `www-scf.usc.edu/~boqinggo/domainadaptation.html`

Method	$A \to C$	$A \to D$	$A \to W$	$C \to A$	$C \to D$	$C \to W$
NO ADAPT-1NN	24 ± 0.3	28.1 ± 0.6	31.6 ± 0.6	23.1 ± 0.4	26.6 ± 0.7	25.2 ± 0.8
NO ADAPT-SVM	43.5 ± 1.6	57.9 ± 3.2	55.6 ± 2.8	51.3 ± 2	61.2 ± 2.7	58.8 ± 2.4
Metric [26]	27.3 ± 0.7	33.7 ± 0.9	36 ± 1	33.7 ± 0.8	35 ± 1.1	34.7 ± 1
GFK-1NN[15]	39.6 ± 0.4	50.9 ± 0.9	56.9 ± 1	46.1 ± .06	55 ± 0.9	57 ± 0.9
GFK-SVM[15]	42.9 ± 1.8	55.1 ± 3.6	55.1 ± 2.9	51.4 ± 2.1	49.8 ± 3.6	54.6 ± 2.5
SSTCA[24]	40.4 ± 1.0	39.0 ± 1.2	41.1 ± 1.1	47.1 ± 1.1	41.7 ± 1.1	36.2 ± 1.0
SS-DIP	47.4 ± 1.5	60.8 ± 3.1	60.3 ± 3.9	57.1 ± 2.5	59.6 ± 4.1	66.1 ± 3.2
SS-DIP-CC	**47.8 ± 1.5**	**67.5 ± 4**	**72.5 ± 3.1**	**61.8 ± 2.5**	**65.8 ± 3.5**	**69.9 ± 2.9**

Table 4. Recognition accuracies on 6 pairs of source/target domains using the semi-supervised evaluation protocol of [26]. C**: Caltech,** A**: Amazon,** W**: Webcam,** D**: DSLR**.

Method	$D \to A$	$D \to C$	$D \to W$	$W \to A$	$W \to C$	$W \to D$
NO ADAPT-1NN	30.8 ± 0.6	20.8 ± 0.5	44.3 ± 1	31.3 ± 0.7	22.4 ± 0.5	55.5 ± 0.7
NO ADAPT-SVM	46.6 ± 1.9	37.8 ± 1.2	82.8 ± 2.2	46.2 ± 1.6	38.6 ± 0.9	84.8 ± 2.3
Metric [26]	30.3 ± 0.8	22.5 ± 0.6	55.6 ± 0.7	32.3 ± 0.8	21.7 ± 0.5	51.3 ± 0.9
GFK-1NN [15]	46.2 ± 0.6	33.9 ± 0.6	80.2 ± 0.4	46.2 ± 0.7	32.8 ± 0.7	75 ± 0.7
GFK-SVM [15]	48.5 ± 1.9	39.2 ± 1.3	79.6 ± 1.1	46.6 ± 1.3	39.3 ± 1.5	75.4 ± 1.9
SSTCA[24]	40.1 ± 1.2	34.2 ± 1.0	80.5 ± 2.0	41.5 ± 1.3	33.5 ± 1.1	77.8 ± 3.1
SS-DIP	52.7 ± 2.2	42.8 ± 1.1	**90.1 ± 1.3**	50 ± 2.1	40.1 ± 1.1	90.1 ± 1.5
SS-DIP-CC	**56.9 ± 1.6**	**44.2 ± 1.3**	89.1 ± 1.6	**53.4 ± 1.9**	**43.6 ± 1.2**	**92.6 ± 1.4**

Table 5. Recognition accuracies on the remaining 6 pairs of source/target domains using the semi-supervised evaluation protocol of [26]. C**: Caltech,** A**: Amazon,** W**: Webcam,** D**: DSLR**.

References

[1] P. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.

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

[3] A. Bergamo and L. Torresani. Exploiting weakly-labeled web images to improve object classification: a domain adaptation approach. In *NIPS*, 2010.

[4] J. Blitzer, D. Foster, and S. Kakade. Domain adaptation with coupled subspaces. *JMLR*, 2011.

[5] J. Blitzer, R. McDonald, and F. Pereira. Domain adaptation with structural correspondence learning. In *Conf. Empirical Methods in Natural. Lang. Proc.*, 2006.

[6] K. Borgwardt, A. Gretton, M. Rasch, H. Kriegel, B. Schoelkopf, and A. Smola. Integrating structured biological data by kernel maximum mean discrepancy. *J. Bioinformatics*, 2006.

[7] L. Bruzzone and M. Marconcini. Domain adaptation problems: A dasvm classification technique and a circular validation strategy. *TPAMI*, 2010.

[8] M. Chen, K. Weinberger, and J. Blitzer. Co-training for domain adaptation. In *NIPS*, 2011.

[9] H. Daumé III, A. Kumar, and A. Saha. Co-regularization based semi-supervised domain adaptation. In *NIPS*, 2010.

[10] H. Daumé III and D. Marcu. Domain adaptation for statistical classifiers. *JAIR*, 2006.

[11] L. Duan, I. Tsang, D. Xu, and T. Chua. Domain adaptation from multiple sources via auxiliary classifiers. In *ICML*, 2009.

[12] L. Duan, I. Tsang, D. Xu, and S. Maybank. Domain transfer svm for video concept detection. In *CVPR*, 2009.

[13] A. Edelman, T. Arias, and S. Smith. The geometry of algorithms with orthogonality constraints. *SIAM*, 1998.

[14] B. Gong, K. Grauman, and F. Sha. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *ICML*, 2013.

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

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

[17] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *JMLR*, 2012.

[18] A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Schölkopf. Covariate shift by kernel mean matching. *J. Royal. Statistical Society*, 2009.

[19] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical report, Calif. Inst. of Tech., 2007.

[20] J. Hoffman, B. Kulis, T. Darrell, and K. Saenko. Discovering latent domains for multisource domain adaptation. In *ECCV*, 2012.

[21] J. Huang, A. J. Smola, A. Gretton, K. Borgwardt, and B. Scholkopf. Correcting sample selection bias by unlabeled data. In *NIPS*, 2007.

[22] V. Jain and E. Learned-Miller. Online domain adaptation of a pre-trained cascade of classifiers. In *CVPR*, 2011.

[23] B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *CVPR*, 2011.

[24] S. Pan, I. Tsang, J. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. *TNN*, 2011.

[25] A. Ruszczynski. *Nonlinear optimization*. Princeton University press, 2006.

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

[27] I. Steinwart. On the influence of the kernel on the consistency of support vector machines. *JMLR*, 2002.

[28] D. Xing, W. Dai, G. Xue, and Y. Yu. Bridged refinement for transfer learning. In *ECML*, 2007.

[29] Q. Yang, J. Pan, and V. Zheng. Estimating location using wi-fi. *IEEE Intelligent Systems*, 2008.