Journal of Machine Learning Research 17 (2016) 1-30 Submitted 5/15; Revised 2/16; Published 7/16

Distribution-Matching Embedding for Visual Domain Adaptation

Mahsa Baktashmotlagh M.BAKTASHMOTLAGH@QUT.EDU.AU
Queensland University of Technology
Brisbane, Australia

Mehrtash Harandi MEHRTASH.HARANDI@NICTA.COM.AU
Australian National University & NICTA[*]
Canberra, Australia

Mathieu Salzmann MATHIEU.SALZMANN@EPFL.CH
CVLab, EPFL[*]
Lausanne, Switzerland

Editor: Urun Dogan, Marius Kloft, Francesco Orabona, and Tatiana Tommasi

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 Distribution-Matching Embedding approach: An unsupervised domain adaptation method that overcomes this issue by mapping the data to a latent space where the distance between the empirical distributions of the source and target examples is minimized. In other words, we seek to extract the information that is invariant across the source and target data. In particular, we study two different distances to compare the source and target distributions: the Maximum Mean Discrepancy and the Hellinger distance. Furthermore, we show that our approach allows us to learn either a linear embedding, or a nonlinear one. We demonstrate the benefits of our approach on the tasks of visual object recognition, text categorization, and WiFi localization.

Keywords: Domain Adaptation, Maximum Mean Discrepancy, Hellinger Distance, Distribution Matching, Domain Invariant Representations

1. Introduction

As evidenced by the recent surge of interest in domain adaptation (Saenko et al., 2010; Jain and Learned-Miller, 2011; Gong et al., 2012; Gopalan et al., 2011), domain shift is a fundamental problem for visual recognition. This problem typically occurs when the training and test images are acquired with different cameras, or in very different conditions (e.g., commercial website versus home environment, images taken under different illuminations). As a consequence, the training (source) and test (target) samples follow different distributions. As demonstrated in, e.g., (Saenko et al., 2010; Gopalan et al., 2011; Gong et al., 2012, 2013), failing to model this distribution shift in the hope that the image features will be robust enough often yields poor recognition accuracy.

[*] 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. This research was performed while M. Salzmann was affiliated with the Australian National University and NICTA.

While labeling sufficiently many images from the target domain to train a discriminative classifier specific to this domain could alleviate this problem, it typically is prohibitively time-consuming and impractical in realistic scenarios. Domain adaptation therefore seeks to prevent this by explicitly modeling the domain shift.

Existing domain adaptation methods can be divided into two categories: Semi-supervised approaches that assume that a small number of labeled examples from the target domain are available during training, and unsupervised approaches that do not require any labels from the target domain. In the former category, modifications of existing classifiers have been proposed to exploit the availability of labeled and unlabeled data from the target domain (Daumé III and Marcu, 2006; Duan et al., 2009b; Bergamo and Torresani, 2010; Tommasi and Caputo, 2013). Co-regularization and adaptive regularization of similar classifiers was also introduced to utilize unlabeled target data during training (Daumé III et al., 2010; Rückert and Kloft, 2011). Multi-Model knowledge transfer was proposed to select and weigh prior knowledge coming from different categories (Jie et al., 2011; Tommasi et al., 2014, 2010). For visual recognition, metric learning (Saenko et al., 2010) and transformation learning (Kulis et al., 2011) were shown to be effective at making use of the labeled target examples. Furthermore, semi-supervised methods have also been employed to tackle the case where multiple source domains are available (Duan et al., 2009a; Hoffman et al., 2012). While semi-supervised methods are often effective, in many applications, labeled target examples are not available and cannot easily be acquired.

By contrast, unsupervised domain adaptation approaches rely on purely unsupervised target data (Xing et al., 2007; Bruzzone and Marconcini, 2010; Chen et al., 2011; Kuzborskij and Orabona, 2013). In particular, two types of methods have proven quite successful at the task of visual object recognition: Subspace-based approaches and sample re-weighting techniques. Subspace-based approaches (Blitzer et al., 2011; Gong et al., 2012; Gopalan et al., 2011) typically model each domain with a subspace, and attempt to relate the source and target representations via intermediate subspaces. While these methods have proven effective in practice, they suffer from the fact that they do not explicitly try to match the probability distributions of the source and target data. Therefore, they may easily yield sub-optimal representations for classification. By contrast, sample selection, or re-weighting, approaches (Huang et al., 2006; Gretton et al., 2009; Gong et al., 2013) explicitly attempt to match the source and target distributions by finding the most appropriate source examples for the target data. However, these methods fail to account for the fact that the image features themselves may have been distorted by the domain shift, and that some of these 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 discovering the information that is invariant across the source and target domains. To this end, we introduce a Distribution-Matching Embedding (DME) approach, which aims to learn a 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 samples to perform well on the target domain.

More specifically, here, we study two different distances to compare the source and target distributions in the latent space. First, we make use of the Maximum Mean Discrepancy (MMD) (Gretton et al., 2012a), which compares the means of two empirical distributions in a reproducing kernel Hilbert space. While the MMD is endowed with nice properties (Gretton et al., 2012a), it does not truly consider the geometry of the space of probability distributions. From information geometry,

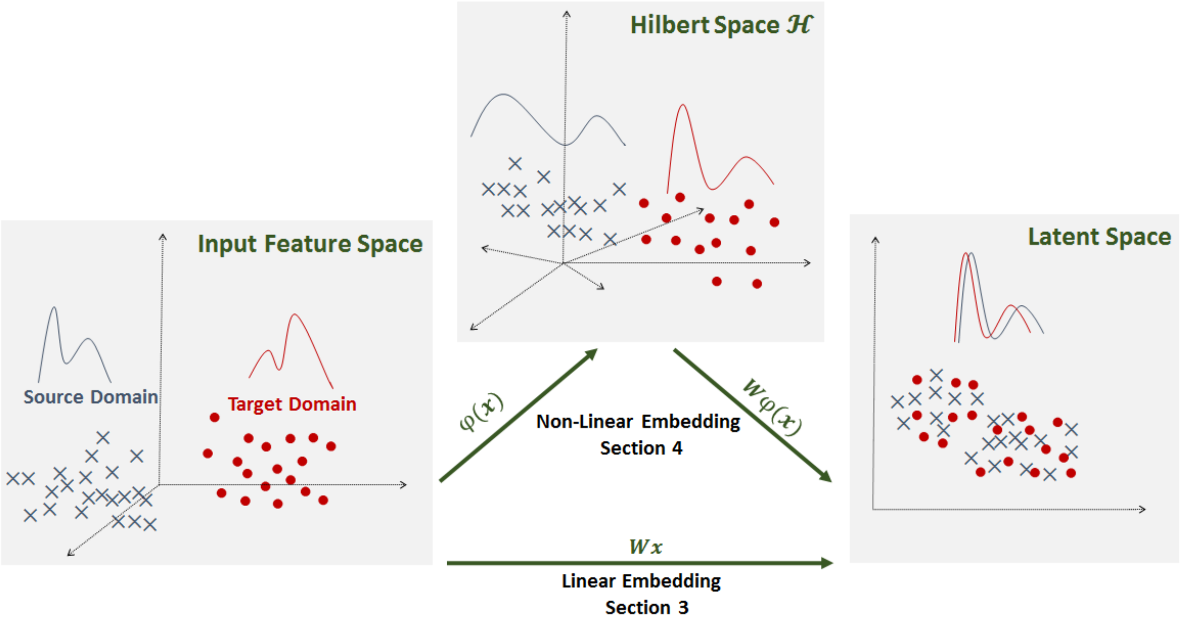


Figure 1: **Illustration of our approach.** Our goal is to learn a latent space, via either a linear mapping or a nonlinear one, such that the source and target distributions in this latent space are as similar as possible.

we know that probability distributions lie on a Riemannian manifold named the statistical manifold. In computer vision, it has been consistently demonstrated that exploiting the Riemannian metric of the manifold to compare manifold-values entities, such as covariance descriptors (Tuzel et al., 2008), linear subspaces (Harandi et al., 2013), or rotation matrices (Hartley et al., 2013), was beneficial for the task at hand. Therefore, we follow a similar intuition here and make use of the Riemannian metric on the statistical manifold as a second distance measure to compare the source and target distributions. Since the true Riemmannian metric, i.e., the Fisher-Rao metric, is difficult to use with general, non-parametric distributions, such as those obtained by kernel density estimation, we propose to rely on the Hellinger distance, which we show to be closely related to the Fisher-Rao Riemannian metric.

Given these two distances, we first introduce algorithms to learn a linear mapping to a low-dimensional latent space where the source and target distributions are similar. By exploiting the Riesz representer theorem (Schölkopf and Smola, 2002), we then show that, for both distances, our approach also allows us to learn a nonlinear embedding to a distribution-matching latent space. In the linear and the nonlinear scenarios, learning our Distribution-Matching Embeddings can then be formulated as an optimization problem on a Grassmann manifold. This lets us utilize Grassmannian geometry to effectively obtain our latent representations. Fig. 1 illustrates our approach, both for linear and nonlinear mappings. In essence, our approach consists of two main components: (i) a mapping to a latent space, which in the nonlinear case is achieved via a mapping to a high-dimensional Hilbert space; and (ii) a distance to compare the source and target distributions in the latent space. While, here, we rely on either the MMD or the Hellinger distance, our approach is general and could potentially be extended to other distance measures.

In short, we introduce the idea of finding a distribution-matching representation of the source and target data, and propose several effective algorithms to learn such a representation. We demonstrate the benefits of our approach on the tasks of visual object recognition, text categorization and Wifi localization using standard domain adaptation data sets. This article is an extended version of our ICCV 2013 (Baktashmotlagh et al., 2013) and CVPR 2014 (Baktashmotlagh et al., 2014) papers. Compared to the conference papers, it contains additional details about the linear formulations, as well as new results. Furthermore, it introduces the nonlinear formulations of our previous methods.

2. Preliminaries

In this section, we provide the background theory and groundwork for the techniques described in the following sections. In particular, we discuss the idea of Maximum Mean Discrepancy and review the derivation of the Hellinger distance on statistical manifolds, as well as study its relationship with the Fisher-Rao Riemannian metric. We then introduce some notions of Grassmannian geometry, as well as discuss the conjugate gradient (CG) algorithm that will be used in our optimization process.

2.1 Maximum Mean Discrepancy (MMD)

In this work, we are interested in measuring the distance 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.

To this end, here, we employ the Maximum Mean Discrepancy (Gretton et al., 2012a). 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

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

where $E_{\boldsymbol{x} \sim s}[\cdot]$ 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 $s = t$ (Gretton et al., 2012a).

Let $\boldsymbol{X_s} = \{\boldsymbol{x}_1^s, \cdots, \boldsymbol{x}_n^s\}$ and $\boldsymbol{X_t} = \{\boldsymbol{x}_1^t, \cdots, \boldsymbol{x}_m^t\}$ be two sets of observations drawn i.i.d. from s and t, respectively. An empirical estimate of the MMD can be computed as

$$\begin{aligned}\hat{D}_M(\tilde{\boldsymbol{X}}_s, \tilde{\boldsymbol{X}}_t) &= \left\| \frac{1}{n} \sum_{i=1}^{n} \phi(\boldsymbol{x}_i^s) - \frac{1}{m} \sum_{j=1}^{m} \phi(\boldsymbol{x}_j^t) \right\|_{\mathcal{H}} \\ &= \left(\sum_{i,j=1}^{n} \frac{k(\boldsymbol{x}_i^s, \boldsymbol{x}_j^s)}{n^2} + \sum_{i,j=1}^{m} \frac{k(\boldsymbol{x}_i^t, \boldsymbol{x}_j^t)}{m^2} - 2 \sum_{i,j=1}^{n,m} \frac{k(\boldsymbol{x}_i^s, \boldsymbol{x}_j^t)}{nm} \right)^{\frac{1}{2}} ,\end{aligned}$$

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.

2.2 Hellinger Distance on Statistical Manifolds

In this section, we review some concepts of Riemannian geometry on statistical manifolds. In particular, we focus on the derivation of the Hellinger distance, which will be used in our algorithms.

Statistical manifolds are Riemannian manifolds whose elements are probability distributions. Loosely speaking, given a non-empty set $\mathcal{X}$ and a family of probability density functions $p(\boldsymbol{x}|\boldsymbol{\theta})$ parametrized by $\boldsymbol{\theta}$ on $\mathcal{X}$, the space $\mathcal{M} = \{p(\boldsymbol{x}|\boldsymbol{\theta})|\boldsymbol{\theta} \in \mathbb{R}^d\}$ forms a Riemannian manifold. The Fisher-Rao Riemannian metric on $\mathcal{M}$ is a function of $\boldsymbol{\theta}$ and induces geodesics, i.e., curves with minimum length on $\mathcal{M}$.

While the Fisher-Rao metric can be computed for specific parametric distributions, such as a Gaussian or a Gaussian mixture (Peter and Rangarajan, 2006), for other parametric forms, it does not even have a closed form solution. More importantly, in general, the parametrization of the PDFs of the data at hand is unknown, and choosing a specific distribution may not reflect the reality. Unfortunately, the Fisher-Rao metric is ill-suited to handle non-parametric distributions, which are of interest for our purpose. Therefore, several studies have opted for approximations of the Fisher-Rao metric. For instance, (Srivastava et al., 2007) proposed to map distributions to the hyper-sphere and use geodesics on this different type of manifold. By contrast, here, we make use of another class of approximations relying on f-divergences, which can be expressed as

$$D_f(s\|t) = \int f(\frac{s(\boldsymbol{x})}{t(\boldsymbol{x})})t(\boldsymbol{x})d\boldsymbol{x} \ .$$

The (squared) Hellinger distance is a special case of f-divergences, obtained by taking $f(y) = (\sqrt{y}-1)^2$. The (squared) Hellinger distance can thus be written as

$$D_H^2(s\|t) = \int \left(\sqrt{s(\boldsymbol{x})} - \sqrt{t(\boldsymbol{x})}\right)^2 d\boldsymbol{x} \ , \tag{1}$$

which is symmetric, satisfies the triangle inequality and is bounded from above by 2.

More importantly, in the following theorem, we show an interesting relationship between the Hellinger distance and the Fisher-Rao Riemannian metric.

Theorem 1 *The length of any curve γ is the same under the Fisher-Rao metric D_{FR} and the Hellinger distance D_H up to a scale of 2.*

Proof We start with the definition of intrinsic metric and curve length. Without any assumption on differentiability, let (M, d) be a metric space. A curve in M is a continuous function $\gamma : [0, 1] \to M$ and joins the starting point $\gamma(0) = p$ to the end point $\gamma(1) = q$. Our proof then relies on two theorems from (Hartley et al., 2013) stated below. To state and exploit these two theorems, we first need the following two definitions coming from (Hartley et al., 2013):

Definition 2 *The length of a curve γ is the supremum of $\ell(\gamma; \alpha_i)$ over all possible partitions α_i such that $0 = \alpha_0 < \alpha_1 < ... < \alpha_n = 1$, where $\ell(\gamma; \alpha_i) = \sum_i d(\gamma(\alpha_i), \gamma(\alpha_{i-1}))$.*

Definition 3 *The intrinsic metric $\hat{\delta}(x, y)$ between two points x and y on a metric space M is defined as the infimum of the lengths of all paths from x to y.*

Theorem 4 *(Hartley et al., 2013) If the intrinsic metrics induced by two metrics d_1 and d_2 are identical to scale ξ, then the length of any given curve is the same under both metrics up to ξ.*

Proof We refer the reader to (Hartley et al., 2013) for the proof of this theorem. ■

Theorem 5 *(Hartley et al., 2013) If $d_1(s,t)$ and $d_2(s,t)$ are two metrics defined on a space M such that*

$$\lim_{d_1(s,t)\to 0} \frac{d_2(s,t)}{d_1(s,t)} = 1 \tag{2}$$

uniformly (with respect to s and t), then their intrinsic metrics are identical.

Proof We refer the reader to (Hartley et al., 2013) for the proof of this theorem. ■

In (Kass and Vos, 2011), it was shown that $\lim_{s\to t} D_H(s\|t) = 0.5 D_{FR}(s\|t)$. The asymptotic behavior of the Hellinger distance and the Fisher-Rao metric can be expressed as $D_H(s,t) = 0.5 * D_{FR}(s,t) + O(D_{FR}(s,t)^3)$ as $s \to t$. This guarantees uniform convergence since the higher order terms are bounded and vanish rapidly independently of the path between s and t. It therefore directly follows from Theorems 5 and 4 that the length of a curve under D_H and D_{FR} is the same up to a scale of 2, which concludes the proof.

■

2.2.1 EMPIRICAL ESTIMATE OF THE HELLINGER DISTANCE

In a practical scenario, our goal is to compute the Hellinger distance between the distributions s and t when discrete observations are provided. In other words, we are interested in estimating Eq. 1 given n samples $\{\boldsymbol{x}_i^s\}$ drawn from s and m samples $\{\boldsymbol{x}_i^t\}$ drawn from t.

To have a symmetric and bounded estimate of the Hellinger distance with respect to a single density, we begin by defining $T(x) = \frac{s(x)}{s(x)+t(x)}$. The Hellinger distance can then be defined in terms of $T(x)$ as

$$\begin{aligned} D_H = & \int (\sqrt{s(x)} - \sqrt{t(x)})^2 dx \\ = & \int \left(\sqrt{\frac{s(x)}{s(x)+t(x)}} - \sqrt{\frac{t(x)}{s(x)+t(x)}} \right)^2 (s(x)+t(x))\, dx \\ = & \int (\sqrt{T(x)} - \sqrt{1-T(x)})^2 s(x) dx + \int (\sqrt{T(x)} - \sqrt{1-T(x)})^2 t(x) dx \,. \end{aligned} \tag{3}$$

Since the two terms in Eq. 3 are expectations, and following the strong law of large numbers, given our two sets of samples $\{\boldsymbol{x}_i^s\}$ and $\{\boldsymbol{x}_i^t\}$, an empirical estimate of the Hellinger distance can be obtained as (Carter, 2009)

$$\hat{D}_H^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\sqrt{\hat{T}(\boldsymbol{x}_i^s)} - \sqrt{1-\hat{T}(\boldsymbol{x}_i^s)}\right)^2 + \frac{1}{m}\sum_{i=1}^{m}\left(\sqrt{\hat{T}(\boldsymbol{x}_i^t)} - \sqrt{1-\hat{T}(\boldsymbol{x}_i^t)}\right)^2 , \tag{4}$$

where $\hat{T}(\boldsymbol{x}) = \hat{s}(\boldsymbol{x}) / \big(\hat{s}(\boldsymbol{x}) + \hat{t}(\boldsymbol{x})\big)$, with $\hat{s}(\boldsymbol{x})$ and $\hat{t}(\boldsymbol{x})$ the empirical estimates of $s(\boldsymbol{x})$ and $t(\boldsymbol{x})$, respectively. Importantly, this numerical approximation respects some of the properties of the true Hellinger distance (Carter, 2009). In particular, it is symmetric and bounded from above by 2.

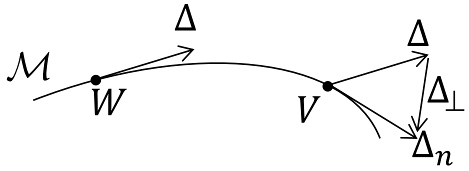


Figure 2: Parallel transport of a tangent vector Δ from a point $\boldsymbol{W}$ to point $\boldsymbol{V}$ on the manifold.

2.3 Grassmann Manifolds

The Grassmann manifold $\mathcal{G}(d, D)$ consists of the set of all linear d-dimensional subspaces of $\mathbb{R}^D$. In particular, for $\boldsymbol{W} \in \mathcal{G}(d, D)$, this lets us handle constraints of the form $\boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}$. As will be shown in Section 3, our mappings to latent space involve nonlinear optimization on the Grassmann manifold. Below, we therefore review some useful notions of differential geometry.

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. As illustrated in Fig. 2, unlike in flat spaces, this cannot be achieved by simple translation, but requires subtracting a normal component at the end point (Edelman et al., 1998).

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 (Edelman et al., 1998). CG techniques are popular nonlinear optimization methods with fast convergence rates. These methods iteratively optimize the objective function in linearly independent directions called conjugate directions (Ruszczynski, 2006). 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

$$\nabla f_{\boldsymbol{W}} = \partial f_{\boldsymbol{W}} - \boldsymbol{W}\boldsymbol{W}^T \partial f_{\boldsymbol{W}} \ , \tag{5}$$

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.

Note that, while we rely on a conjugate gradient method, other optimization strategies have been studied on Grassmann manifolds, such as (i) Stochastic-gradient flow, where a stochastic component is added to the gradient to construct a stochastic gradient process such that the solution converges to a global optimum in the limit; (ii) Acceptance-rejection methods, where the stochastic gradient part provides candidates to update the estimate, that are accepted/rejected according to a probability density function (Metropolis-Hastings type acceptance-rejection step); and (iii) Simulated annealing, where instead of sampling from a probability distribution, an annealing procedure is applied to find the optimal points of the function f (Srivastava and Liu, 2005). A complete study of these Grassmannian optimization strategies goes beyond the scope of this paper.

3. Distribution-Matching Embedding (DME)

In this section, we introduce our approach to unsupervised domain adaptation, which relies on mapping the data to a low-dimensional latent space such that the distance between the source and target distributions is minimized. Intuitively, with such a latent representation, a classifier trained on the source domain should perform equally well on the target domain. In particular, here, we make use of a linear mapping of the form

$$\boldsymbol{y} = \boldsymbol{W}^T \boldsymbol{x} \ , \tag{6}$$

where $\boldsymbol{x} \in \mathbb{R}^D$ is the original data (e.g., image features), $\boldsymbol{y} \in \mathbb{R}^d$ is the resulting low-dimensional representation, and $\boldsymbol{W} \in \mathbb{R}^{D\times d}$ is the parameter matrix that we seek to learn. Furthermore, we enforce orthogonality constraints on $\boldsymbol{W}$, such that

$$\boldsymbol{W}^T \boldsymbol{W} = \boldsymbol{I} \ . \tag{7}$$

These constraints typically avoid degeneracies, such as having all samples collapsing to the origin, and have proven effective in many dimensionality reduction methods, such as Principal Component Analysis (PCA) and Canonical Correlation Analysis (CCA).

In the remainder of this section, we denote by $s(\boldsymbol{x})$ and $t(\boldsymbol{x})$ the probability density functions of the source samples $\boldsymbol{X_s} = [\boldsymbol{x}_1^s, \cdots, \boldsymbol{x}_n^s]$ and target samples $\boldsymbol{X_t} = \left[\boldsymbol{x}_1^t, \cdots, \boldsymbol{x}_m^t\right]$, respectively, where each $\boldsymbol{x}_i^* \in \mathbb{R}^D$. We first derive our MMD-based algorithm (DME-MMD), and then discuss our formulation based on the Hellinger distance (DME-H).

3.1 DME with the MMD (DME-MMD)

To derive our first approach to learning a distribution-matching representation, we make use of the MMD to measure the distance between the source and target distributions. Following the derivations provided in Section 2, and by making use of the linear mapping defined in Eq. 6, the MMD in the latent space can be expressed as

$$\hat{D}_M(\boldsymbol{W}^T\boldsymbol{X_s}, \boldsymbol{W}^T\boldsymbol{X_t}) = \left\| \frac{1}{n}\sum_{i=1}^{n} \phi(\boldsymbol{W}^T\boldsymbol{x}_i^s) - \frac{1}{m}\sum_{j=1}^{m} \phi(\boldsymbol{W}^T\boldsymbol{x}_j^t) \right\|_{\mathcal{H}} , \tag{8}$$

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.

Using the MMD, in conjunction with the constraints described in Eq. 7, learning $\boldsymbol{W}$ can be expressed as the optimization problem

$$\begin{aligned} \boldsymbol{W}^* \quad &= \quad \underset{\boldsymbol{W}}{\text{argmin}} \ D^2(\boldsymbol{W}^T\boldsymbol{X_s}, \boldsymbol{W}^T\boldsymbol{X_t}) \\ &\text{s.t.} \quad \boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I} \ . \end{aligned} \tag{9}$$

As shown in Section 2, the MMD can be expressed in terms of a kernel function $k(\cdot,\cdot)$. Here, we first propose to exploit the Gaussian kernel function, which is known to be universal (Steinwart, 2002). This lets us rewrite our objective function as

$$\begin{aligned}\hat{D}^2_M(\boldsymbol{W}^T\boldsymbol{X_s}, \boldsymbol{W}^T\boldsymbol{X_t}) = &\ \frac{1}{n^2}\sum_{i,j=1}^{n} \exp\left(-\frac{(\boldsymbol{x}_i^s - \boldsymbol{x}_j^s)^T\boldsymbol{W}\boldsymbol{W}^T(\boldsymbol{x}_i^s - \boldsymbol{x}_j^s)}{\sigma}\right) \\ &+\frac{1}{m^2}\sum_{i,j=1}^{m} \exp\left(-\frac{(\boldsymbol{x}_i^t - \boldsymbol{x}_j^t)^T\boldsymbol{W}\boldsymbol{W}^T(\boldsymbol{x}_i^t - \boldsymbol{x}_j^t)}{\sigma}\right) \\ &-\frac{2}{mn}\sum_{i,j=1}^{n,m} \exp\left(-\frac{(\boldsymbol{x}_i^s - \boldsymbol{x}_j^t)^T\boldsymbol{W}\boldsymbol{W}^T(\boldsymbol{x}_i^s - \boldsymbol{x}_j^t)}{\sigma}\right) \ ,\end{aligned} \tag{10}$$

where, in practice, we take σ to be the median squared distance between all the source examples.

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 (Borgwardt et al., 2006). 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 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 (Gretton et al., 2012a) 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

$$\begin{aligned}\hat{D}^2_M(\boldsymbol{W}^T\boldsymbol{X_s}, \boldsymbol{W}^T\boldsymbol{X_t}) = &\ \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}(1 + \boldsymbol{x}_i^{s^T}\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{x}_j^s)^2 \\ &+\frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}(1 + \boldsymbol{x}_i^{t^T}\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{x}_j^t)^2 \\ &-\frac{2}{mn}\sum_{i=1}^{n}\sum_{j=1}^{m}(1 + \boldsymbol{x}_i^{s^T}\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{x}_j^t)^2 .\end{aligned} \tag{11}$$

The two variants of the MMD introduced in Eqs. 10 and 11 can be computed efficiently in matrix form as

$$\hat{D}^2_M(\boldsymbol{W}^T\boldsymbol{X_s}, \boldsymbol{W}^T\boldsymbol{X_t}) = Tr(\boldsymbol{K_W}\boldsymbol{L}) \ , \tag{12}$$

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, $\boldsymbol{K}_{\boldsymbol{W}}$ can be computed efficiently in matrix form (i.e., without looping over its elements). This yields the optimization problem

$$\begin{aligned} \boldsymbol{W}^* &= \underset{\boldsymbol{W}}{\text{argmin}} \;\; Tr\left(\boldsymbol{K}_{\boldsymbol{W}}\boldsymbol{L}\right) \\ &\text{s.t. } \boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}\ , \end{aligned} \tag{13}$$

which is a nonlinear constrained problem. Due to the constraints, this problem can be solved either on the Stiefel manifold, or on the Grassmann manifold. The main difference between these two manifolds lies in the fact that, on the Grassmannian, two subspaces that are identical up to a d-dimensional rotation are identified as the same point on the manifold. In other words, a point on the Grassmann manifold is an equivalence class. It can easily be verified that, with our two kernels, a rotation of $\boldsymbol{W}$ would yield exactly the same objective function value. Therefore, our problem can be solved on the Grassmann manifold. The details of the optimization scheme and the resulting algorithm will be discussed in Section 3.3.

3.2 DME with the Hellinger Distance (DME-H)

While the MMD has nice properties (Gretton et al., 2012a), it does not truly consider the geometry of the space of probability distributions. Furthermore, according to (Gretton et al., 2012b), non-optimal choices of kernel and kernel parameters can lead to poor estimates of the distance between two distributions. This therefore motivates the use of the Hellinger distance instead of the MMD, since, as shown in Section 2.2, the Hellinger distance is related to the geodesic distance on the statistical manifold.

Given the linear mapping in Eq. 6 and the definition of the empirical estimate of the Hellinger distance in Eq. 4, we can express the (squared) distance between the source and target distributions as

$$\begin{aligned} \hat{D}^2_H(\boldsymbol{W}^T\boldsymbol{X_s}, \boldsymbol{W}^T\boldsymbol{X_t}) &= \frac{1}{n}\sum_{i=1}^{n}\left(\sqrt{\hat{T}(\boldsymbol{W}^T\boldsymbol{x}_i^s)} - \sqrt{1-\hat{T}(\boldsymbol{W}^T\boldsymbol{x}_i^s)}\right)^2 \\ &+ \frac{1}{m}\sum_{i=1}^{m}\left(\sqrt{\hat{T}(\boldsymbol{W}^T\boldsymbol{x}_i^t)} - \sqrt{1-\hat{T}(\boldsymbol{W}^T\boldsymbol{x}_i^t)}\right)^2 \ . \end{aligned} \tag{14}$$

This distance depends on the function $\hat{T}(\boldsymbol{W}^T\boldsymbol{x})$, which, as mentioned in Section 2.2.1, is derived from the empirical estimates of the source and target distributions, $\hat{s}(\boldsymbol{W}^T\boldsymbol{x})$ and $\hat{t}(\boldsymbol{W}^T\boldsymbol{x})$, respectively.

In this work, we make use of kernel density estimation (KDE) with a Gaussian kernel to model these distributions. This lets us write

$$\hat{s}(\boldsymbol{W}^T\boldsymbol{x}) = \frac{1}{n}\sum_{j=1}^{n}\frac{1}{\sqrt{|2\pi\boldsymbol{H}_s|}}\exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{x}_j^s)^T\boldsymbol{W}\boldsymbol{H}_s^{-1}\boldsymbol{W}^T(\boldsymbol{x}-\boldsymbol{x}_j^s)}{2}\right), \tag{15}$$

where $\boldsymbol{H}_s$ is a diagonal matrix. In practice, we take $\boldsymbol{H}_s = \sigma_s\boldsymbol{I}$, where σ_s is computed using the maximal smoothing principle (Terrell, 1990) and kept constant. A similar estimate $\hat{t}(\boldsymbol{W}^T\boldsymbol{x})$ can be obtained from the m projected target samples $\{\boldsymbol{W}^T\boldsymbol{x}_j^t\}$. As such, we can write $\hat{T}(\boldsymbol{W}^T\boldsymbol{x})$ as:

$$\hat{T}(\boldsymbol{W}^T\boldsymbol{x}) = \frac{\frac{1}{n}\sum_{j=1}^{n}k(\boldsymbol{W}^T\boldsymbol{x}, \boldsymbol{W}^T\boldsymbol{x}_j^s)}{\frac{1}{n}\sum_{j=1}^{n}k(\boldsymbol{W}^T\boldsymbol{x}, \boldsymbol{W}^T\boldsymbol{x}_j^s) + \frac{1}{m}\sum_{j=1}^{m}k(\boldsymbol{W}^T\boldsymbol{x}, \boldsymbol{W}^T\boldsymbol{x}_j^t)}\ , \tag{16}$$

where $k(\cdot,\cdot)$ is the Gaussian kernel function.

Finding a mapping that minimizes the Hellinger distance between the source and target distributions can then be expressed as

$$\begin{aligned} \boldsymbol{W}^* &= \min_{\boldsymbol{W}} \; \hat{D}_H^2(\boldsymbol{W}^T\boldsymbol{X_s}, \boldsymbol{W}^T\boldsymbol{X_t}) \\ \text{s.t.} &\quad \boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}\ . \end{aligned} \tag{17}$$

As with the MMD, this is a nonlinear, constrained optimization problem, which, because of the form of the constraints, can be modeled as an optimization problem on either the Stiefel manifold or the Grassmannian. As before, it can easily be verified that the objective function will be unaffected by a rotation of $\boldsymbol{W}$. Therefore, this corresponds to a problem on the Grassmann manifold.

3.3 Learning the Mapping

As mentioned in the previous two sections, both DME-MMD and DME-H correspond to optimization problems on the Grassmann manifold. Optimization on Grassmann manifolds has proven effective at avoiding bad local minima (Absil et al., 2008). More precisely, manifold optimization methods often have better convergence behavior than iterative projection methods, which can be crucial with a nonlinear objective function (Absil et al., 2008).

By expressing $\boldsymbol{W}$ as a point on a Grassmann manifold, we can rewrite our constrained optimization problems as unconstrained problems on the manifold $\mathcal{G}(d, D)$. More specifically, we can generally express Problems (13) and (17) as

$$\boldsymbol{W}^* = \underset{\boldsymbol{W}\in\mathcal{G}(d,D)}{\operatorname{argmin}} \quad f(\boldsymbol{W})\ , \tag{18}$$

where $f(\boldsymbol{W})$ represents either the MMD-based objective function, or the Hellinger-based one.

While the optimization problem above has become unconstrained, it remains nonlinear. To effectively address this, we make use of a conjugate gradient method on the manifold. Recall from Section 2.3 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. Our general approach to learning $\boldsymbol{W}$ can then be summarized by Algorithm 1, where we denote by $\tau\left(\Delta, \boldsymbol{W}, \boldsymbol{V}\right)$ the parallel transport of tangent vector Δ from $\boldsymbol{W}$ to $\boldsymbol{V}$. In practice, we initialize $\boldsymbol{W}$ to the truncated identity matrix. We observed that learning $\boldsymbol{W}$ typically converges in only a few iterations.

Note that Eq. 5 shows that the gradient on the manifold depends on the partial derivatives of the objective function w.r.t. $\boldsymbol{W}$, i.e., $\partial f/\partial\boldsymbol{W}$. These derivatives depend on the specific form of the objective function, and are thus different for DME-MMD and DME-H.

For DME-MMD, the general form of $\partial f/\partial\boldsymbol{W}$ can be expressed as

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

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 the MMD in Eq. 10 based on the Gaussian kernel $k_G(\cdot,\cdot)$, the matrix, e.g., $\boldsymbol{G}_{ss}(i,j)$ has the form

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

Algorithm 1 : Learning on a Grassmann Manifold

```
Input:
    $\boldsymbol{X_s}$: the source examples
    $\boldsymbol{X_t}$: the target examples
    $d$: the dimensionality of the subspace
Output:
    $\boldsymbol{W}^* \in \mathbb{R}^{D\times d}$, such that $\boldsymbol{W^*}^T\boldsymbol{W}^* = \boldsymbol{I}$

$\boldsymbol{W}_{prev} \leftarrow \boldsymbol{I}_{D\times d}$ (i.e., truncated identity matrix)
$\boldsymbol{W}_{cur} \leftarrow \boldsymbol{W}_{prev}$
$\boldsymbol{H}_{prev} \leftarrow \boldsymbol{0}$
repeat
    $\boldsymbol{D}_{cur} \leftarrow \nabla f_{\boldsymbol{W}_i}$
    $\boldsymbol{H}_{cur} \leftarrow -\boldsymbol{D}_{cur} + \eta\tau(\boldsymbol{H}_{prev}, \boldsymbol{W}_{prev}, \boldsymbol{W}_{cur})$
    Line search to find $\boldsymbol{W}^*$ that minimizes $f(\boldsymbol{W})$ along the geodesic at $\boldsymbol{W}_{cur}$ in the direction
    $\boldsymbol{H}_{cur}$
    $\boldsymbol{H}_{prev} \leftarrow \boldsymbol{H}_{cur}$
    $\boldsymbol{W}_{prev} \leftarrow \boldsymbol{W}_{cur}$
    $\boldsymbol{W}_{cur} \leftarrow \boldsymbol{W}^*$
until convergence
```

and similarly for $\boldsymbol{G}_{tt}(\cdot,\cdot)$ and $\boldsymbol{G}_{st}(\cdot,\cdot)$. With the MMD of Eq. 11 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 $\boldsymbol{G}_{st}(\cdot,\cdot)$. Similarly to the objective function itself, these derivatives can be efficiently computed in matrix form.

For DME-H, the derivatives can be written as

$$\begin{aligned} \frac{\partial f}{\partial \boldsymbol{W}} &= \tfrac{1}{n}\textstyle\sum_i^n \left\{ \sqrt{\frac{2T\left(\boldsymbol{W}^T\boldsymbol{x}_i^s\right)-1}{T\left(\boldsymbol{W}^T\boldsymbol{x}_i^s\right)\left(1-T\left(\boldsymbol{W}^T\boldsymbol{x}_i^s\right)\right)}}\, \frac{\partial T\left(\boldsymbol{W}^T\boldsymbol{x}_i^s\right)}{\partial \boldsymbol{W}} \right\} \\ &+ \tfrac{1}{m}\textstyle\sum_i^m \left\{ \sqrt{\frac{2T\left(\boldsymbol{W}^T\boldsymbol{x}_i^t\right)-1}{T\left(\boldsymbol{W}^T\boldsymbol{x}_i^t\right)\left(1-T\left(\boldsymbol{W}^T\boldsymbol{x}_i^t\right)\right)}}\, \frac{\partial T\left(\boldsymbol{W}^T\boldsymbol{x}_i^t\right)}{\partial \boldsymbol{W}} \right\} , \end{aligned} \tag{19}$$

where

$$\begin{aligned} \frac{\partial T\left(\boldsymbol{W}^T\boldsymbol{x}\right)}{\partial \boldsymbol{W}} &= \frac{\partial}{\partial \boldsymbol{W}} \frac{s(\boldsymbol{W}^T\boldsymbol{x})}{s(\boldsymbol{W}^T\boldsymbol{x}) + t(\boldsymbol{W}^T\boldsymbol{x})} \\ &= \frac{1}{\left(s(\boldsymbol{W}^T\boldsymbol{x}) + t(\boldsymbol{W}^T\boldsymbol{x})\right)^2} \left(t(\boldsymbol{W}^T\boldsymbol{x}) \frac{\partial s(\boldsymbol{W}^T\boldsymbol{x})}{\partial \boldsymbol{W}} - s(\boldsymbol{W}^T\boldsymbol{x}) \frac{\partial t(\boldsymbol{W}^T\boldsymbol{x})}{\partial \boldsymbol{W}} \right) . \end{aligned} \tag{20}$$

This lets us learn a linear mapping to a low-dimensional subspace that minimizes either the MMD or the Hellinger distance between the source and target data in a completely unsupervised manner. A classifier can then be learned from the labeled source samples projected to this latent space, and directly applied to the projected target samples.

4. Nonlinear DME (NL-DME)

The Distribution-Matching Embedding methods introduced in the previous section make use of a linear mapping. As such, they have limited power to represent complex transformations between the source and target domains. To overcome this limitation, we introduce a nonlinear version of DME, which, as is often the case with nonlinear embedding techniques, boils down to applying a linear method after mapping the data to a high-dimensional Reproducing Kernel Hilbert Space.

More specifically, let $\varphi : \mathbb{R}^D \rightarrow \tilde{\mathcal{H}}$ be the function mapping an input vector $\boldsymbol{x}$ to a high-dimensional RKHS. Our goal is to learn an embedding of the form

$$\boldsymbol{y} = \boldsymbol{W}^T\varphi(\boldsymbol{x}) \; . \tag{21}$$

Since, in theory, $\tilde{\mathcal{H}}$ can be infinite-dimensional, learning the matrix $\boldsymbol{W}$ directly is not practical. Therefore, here, we make use of the Riesz representer theorem (Schölkopf and Smola, 2002; Canu and Smola, 2006) to express $\boldsymbol{W}$ as a linear combination of the examples in $\tilde{\mathcal{H}}$. In other words, we write

$$\boldsymbol{W} = \varphi(\boldsymbol{X_{s+t}})\boldsymbol{\alpha} \; , \tag{22}$$

where $\varphi(\boldsymbol{X_{s+t}})$ is the matrix containing all samples (i.e., source and target) mapped to $\tilde{\mathcal{H}}$, and $\boldsymbol{\alpha} \in \mathbb{R}^{(n+m)\times d}$ corresponds to the new parameters of the mapping.

4.1 NL-DME with the MMD (NL-DME-MMD)

Given the definition of the mapping above, we can write the Gaussian-kernel-based MMD as

$$\begin{aligned}
\hat{D}^2_M(\boldsymbol{W}^T\varphi(\boldsymbol{X_s}), \boldsymbol{W}^T\varphi(\boldsymbol{X_t})) &= \left\| \frac{1}{n}\sum_{i=1}^{n} \phi(\boldsymbol{\alpha}^T\varphi(\boldsymbol{X_{s+t}})^T\varphi(\boldsymbol{x}_i^s)) - \frac{1}{m}\sum_{j=1}^{m} \phi(\boldsymbol{\alpha}^T\varphi(\boldsymbol{X_{s+t}})^T\varphi(\boldsymbol{x}_j^t)) \right\|^2_{\mathcal{H}} \qquad (23)\\
= \frac{1}{n^2}\sum_{i,j=1}^{n} \exp&\left(-\frac{(\tilde{k}(\boldsymbol{X_{s+t}},\boldsymbol{x}_i^s) - \tilde{k}(\boldsymbol{X_{s+t}},\boldsymbol{x}_j^s))^T\boldsymbol{\alpha}\boldsymbol{\alpha}^T(\tilde{k}(\boldsymbol{X_{s+t}},\boldsymbol{x}_i^s) - \tilde{k}(\boldsymbol{X_{s+t}},\boldsymbol{x}_j^s))}{\sigma}\right)\\
+\frac{1}{m^2}\sum_{i,j=1}^{m} \exp&\left(-\frac{(\tilde{k}(\boldsymbol{X_{s+t}},\boldsymbol{x}_i^t) - \tilde{k}(\boldsymbol{X_{s+t}},\boldsymbol{x}_j^t))^T\boldsymbol{\alpha}\boldsymbol{\alpha}^T(\tilde{k}(\boldsymbol{X_{s+t}},\boldsymbol{x}_i^t) - \tilde{k}(\boldsymbol{X_{s+t}},\boldsymbol{x}_j^t))}{\sigma}\right)\\
-\frac{2}{mn}\sum_{i,j=1}^{n,m} \exp&\left(-\frac{(\tilde{k}(\boldsymbol{X_{s+t}},\boldsymbol{x}_i^s) - \tilde{k}(\boldsymbol{X_{s+t}},\boldsymbol{x}_j^t))^T\boldsymbol{\alpha}\boldsymbol{\alpha}^T(\tilde{k}(\boldsymbol{X_{s+t}},\boldsymbol{x}_i^s) - \tilde{k}(\boldsymbol{X_{s+t}},\boldsymbol{x}_j^t))}{\sigma}\right) \; ,
\end{aligned}$$

where $\tilde{k}(\cdot,\cdot)$ is the kernel function corresponding to the mapping to $\tilde{\mathcal{H}}$. Note that the MMD then only depends on kernel values and not on the high-dimensional representation $\varphi(\boldsymbol{x})$. It can easily be verified that this remains true when expressing the MMD with the degree 2 polynomial kernel, as in Eq. 11.

Learning a nonlinear distribution-matching embedding with the MMD can then be expressed as the optimization problem

$$\begin{aligned} \boldsymbol{\alpha}^* &= \underset{\boldsymbol{\alpha}}{\operatorname{argmin}}\ \hat{D}_M^2(\boldsymbol{\alpha}^T\varphi(\boldsymbol{X_{s+t}})^T\varphi(\boldsymbol{X_s}), \boldsymbol{\alpha}^T\varphi(\boldsymbol{X_{s+t}})^T\varphi(\boldsymbol{X_t})) \\ &\text{s.t.} \quad \boldsymbol{\alpha}^T\boldsymbol{K}\boldsymbol{\alpha} = \boldsymbol{I}\ , \end{aligned} \tag{24}$$

As in the linear case, the objective function can be computed efficiently in matrix form, thus yielding the optimization problem

$$\begin{aligned} \boldsymbol{\beta}^* &= \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\ Tr\left(\boldsymbol{K'}_{\boldsymbol{\beta}}\boldsymbol{L}\right) \\ &\text{s.t.}\ \boldsymbol{\beta}^T\boldsymbol{\beta} = \boldsymbol{I}\ , \end{aligned} \tag{25}$$

where $\boldsymbol{L}$ is defined as in (12), $\boldsymbol{K'}_{\boldsymbol{\beta}}$ can be written as

$$\boldsymbol{K'}_{\boldsymbol{\beta}} = \begin{bmatrix} \boldsymbol{K}(\tilde{\boldsymbol{K}}_{s,s+t}, \tilde{\boldsymbol{K}}_{s,s+t}) & \boldsymbol{K}(\tilde{\boldsymbol{K}}_{s,s+t}, \tilde{\boldsymbol{K}}_{t,s+t}) \\ \boldsymbol{K}(\tilde{\boldsymbol{K}}_{t,s+t}, \tilde{\boldsymbol{K}}_{s,s+t}) & \boldsymbol{K}(\tilde{\boldsymbol{K}}_{t,s+t}, \tilde{\boldsymbol{K}}_{t,s+t}) \end{bmatrix}\ ,$$

and we defined a new variable $\boldsymbol{\beta} = \boldsymbol{K}^{1/2}\boldsymbol{\alpha}$. This new variable $\boldsymbol{\beta}$ can be represented as a point on a Grassmann manifold. Therefore, it allows us to make use of the same conjugate gradient method on the manifold as before. The original variable $\boldsymbol{\alpha}$ can then be obtained as $\boldsymbol{\alpha} = \boldsymbol{K}^{-1/2}\boldsymbol{\beta}$.

4.2 NL-DME with the Hellinger Distance (NL-DME-H)

Similarly, from the definition of our nonlinear mapping, the Hellinger distance can be written as

$$\begin{aligned} &\hat{D}_H^2(\boldsymbol{W}^T\varphi(\boldsymbol{X_s}), \boldsymbol{W}^T\varphi(\boldsymbol{X_t})) = \\ &\quad \frac{1}{n}\sum_{i=1}^{n}\left(\sqrt{\hat{T}(\boldsymbol{\alpha}^T\varphi(\boldsymbol{X_{s+t}})^T\varphi(x_i^s))} - \sqrt{1-\hat{T}(\boldsymbol{\alpha}^T\varphi(\boldsymbol{X_{s+t}})^T\varphi(x_i^s))}\right)^2 \\ &\quad + \frac{1}{m}\sum_{i=1}^{m}\left(\sqrt{\hat{T}(\boldsymbol{\alpha}^T\varphi(\boldsymbol{X_{s+t}})^T\varphi(x_i^t))} - \sqrt{1-\hat{T}(\boldsymbol{\alpha}^T\varphi(\boldsymbol{X_{s+t}})^T\varphi(x_i^t))}\right)^2\ . \end{aligned} \tag{26}$$

Using KDE, the distribution of the source data in the latent space can then be expressed as

$$\hat{s}(\boldsymbol{W}^T\varphi(\boldsymbol{x})) = \frac{1}{n}\sum_{j=1}^{n}\frac{1}{\sqrt{|2\pi\boldsymbol{H}|}}\exp\left(-\frac{(\tilde{k}(\boldsymbol{X_{s+t}},\boldsymbol{x}) - \tilde{k}(\boldsymbol{X_{s+t}},\boldsymbol{x}_j^s))^T\boldsymbol{\alpha}\boldsymbol{H}^{-1}\boldsymbol{\alpha}^T(\tilde{k}(\boldsymbol{X_{s+t}},x) - \tilde{k}(\boldsymbol{X_{s+t}},\boldsymbol{x}_j^s))}{2}\right)\ , \tag{27}$$

and similarly for the target distribution. Note that, here again, this only depends on kernel values.

This lets us write NL-DME-H as the optimization problem

$$\begin{aligned} \boldsymbol{\alpha}^* &= \min_{\boldsymbol{\alpha}}\ \hat{D}_H^2(\boldsymbol{\alpha}^T\varphi(\boldsymbol{X_{s+t}})^T\varphi(\boldsymbol{X_s}), \boldsymbol{\alpha}^T\varphi(\boldsymbol{X_{s+t}})^T\varphi(\boldsymbol{X_t})) \\ &\text{s.t.} \quad \boldsymbol{\alpha}^T\boldsymbol{K}\boldsymbol{\alpha} = \boldsymbol{I}\ . \end{aligned} \tag{28}$$

As with the MMD, this problem can be re-written in terms of a new variable $\boldsymbol{\beta} = \boldsymbol{K}^{1/2}\boldsymbol{\alpha}$, and solved using a conjugate gradient method on the Grassmann manifold.

5. Related Work

We now discuss the domain adaptation methods that are most related to our approach. In particular, we focus on sample selection, or re-weighting, techniques and subspace-based methods.

Similarly to our approach, sample selection methods focus on comparing the distributions of the source and target data. In particular, in (Huang et al., 2006; Gretton et al., 2009), 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 the MMD was introduced (Gong et al., 2013). 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. More precisely, these techniques place a weight outside the mapping to Hilbert space $\phi(\cdot)$ performed in the MMD. Unfortunately, the original feature space may not be well-suited to compare the distributions, since the features may have been distorted by the domain shift, and since some of the features may only be relevant to one specific domain. By contrast, in this work, we compare the source and target distributions in a low-dimensional latent space where these effects are removed, or reduced.

Several techniques have also proposed to rely on subspaces to address the problem of domain adaptation. A popular approach in this class of methods differs significantly from our work in that, instead of learning a projection of the data, it seeks to directly represent the data with multiple subspaces (Blitzer et al., 2011; Gopalan et al., 2011; Gong et al., 2012). In particular, in (Blitzer et al., 2011), 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 (Gopalan et al., 2011), 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 (Gong et al., 2012), 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 data.

By contrast, and similarly to our goal, a few methods have proposed to learn linear transformations of the data by considering the distributions of the different domains (Pan et al., 2011; Muandet et al., 2013). In particular, Transfer Component Analysis (TCA) (Pan et al., 2011) makes use of an MMD-based criterion to learn a subspace. However, in TCA, the linear transformation is applied outside the mapping to Hilbert space $\phi(\cdot)$ performed by the MMD. In other words, the distance between the sample means is measured in a lower-dimensional space rather than in RKHS, which somewhat contradicts the intuition behind the use of kernels. Domain-Invariant Component Analysis (DICA) (Muandet et al., 2013) is closely related to TCA in the sense that it is a kernel-based optimization algorithm that learns an invariant transformation of the data by minimizing the dissimilarity across domains. Moreover, it preserves the functional relationship between input and output variables based on the assumption that the functional relationship is stable or varies smoothly across domains.

Importantly, while the above-mentioned approaches have indeed also followed the idea of comparing distributions, they are all confined to using the MMD. In our experiments, we will show that in many cases the Hellinger distance yields better performance. Furthermore, to the best of our knowledge, no existing method has proposed to learn a nonlinear transformation of the data to account for the domain shift. As evidenced by our results, this again can yield to significant gains in accuracy.

6. Experiments

We evaluated our approach on the tasks of visual object recognition, cross-domain text categorization, and cross-domain WiFi localization, and compare its performance against the state-of-the art methods in each task.

In all our experiments, we used the subspace disagreement measure of (Gong et al., 2012) to automatically determine the dimensionality of the projection matrix $\boldsymbol{W}$. This method can be summarized as follows: We extracted PCA subspaces for the source data, the target data, and their combination. Intuitively, the similarity of the source and target domains should be directly proportional to the distance between these three subspaces on the Grassmann manifold. Therefore, the dimensionality d is taken as the one minimizing the sum of the minimum correlation distances (Hamm and Lee, 2008) between the source and combination subspaces and the target and combination subspaces, respectively. The same dimensionality was used for all dimensionality-reduction-based methods, which makes the comparison fair, since the dimensionality was not tuned for our approach either.

For recognition, we employed either a kernel SVM classifier with a degree 2 polynomial kernel, or a linear SVM classifier. These classifiers were trained on the projected source samples. The same types of classifiers were trained for all the baselines that do not inherently include a classifier (i.e., all the baselines except SVMA and DAM). The only parameter of such classifiers is the regularizer weight C. For each method, we tested with $C \in \{10^{-5}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$, and report the best result. Note that the parameters for SVMA, DAM and KMM were chosen using the code provided by the authors of these methods.

As mentioned earlier, the two hyperparameters of our approach were set as follows: The bandwidth of the Gaussian RBF kernel used in MMD was taken as the median distance computed over all pairwise data points; the value σ_s, such that $\boldsymbol{H}_s = \sigma_s \boldsymbol{I}$ in Eq. 15, was computed using the maximal smoothing principle (Terrell, 1990) .

6.1 Visual Object Recognition

We first evaluated our approach on the task of visual object recognition using the benchmark domain adaptation data set introduced in (Saenko et al., 2010). This data set 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 were 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 (Griffin et al., 2007), consists of images of 256 object classes downloaded from Google images. Following (Gong

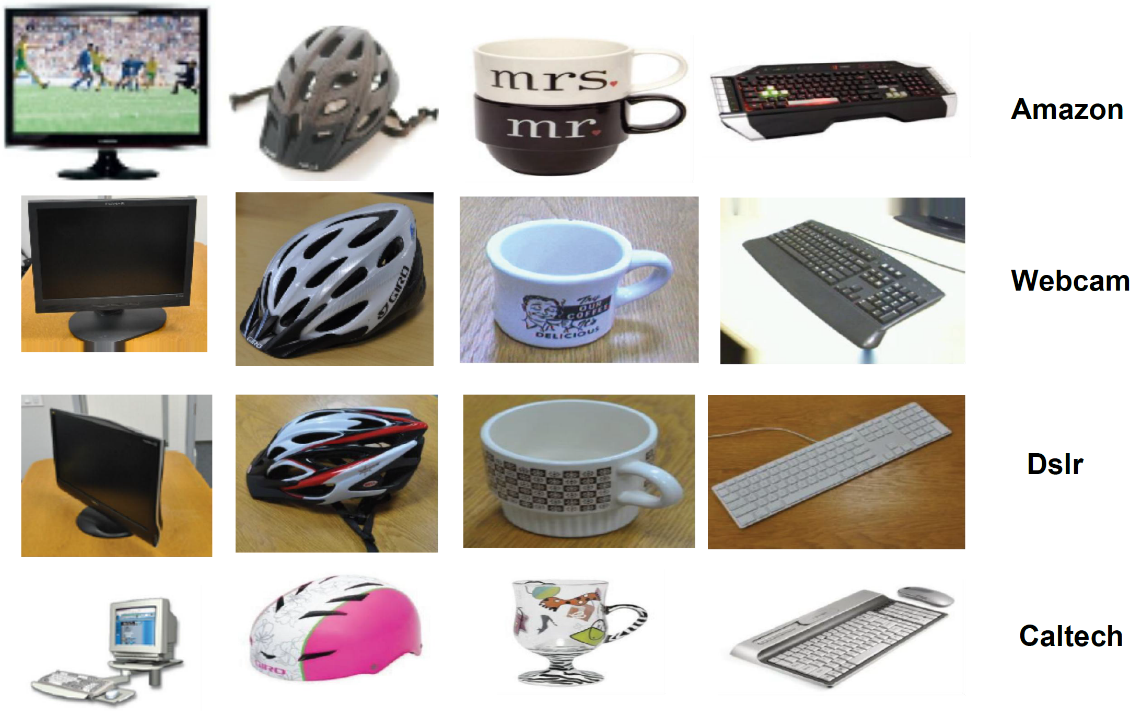


Figure 3: Sample images from the object categories ***monitor***, ***helmet***, ***mug*** and ***keyboard*** *in the four domains* ***Amazon***, ***Webcam***, ***DSLR***, *and* ***Caltech***.

et al., 2012), we used the 10 object classes common to all four data sets. This yields 2533 images in total, with 8 to 151 images per category and per domain. Fig. 3 depicts sample images from the four domains.

For our evaluation, we used two different types of image features. First, we employed the features provided by (Gong et al., 2012), which were obtained using the protocol described in (Saenko et al., 2010). 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 (Bay et al., 2006), 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 data set 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. As a second feature type, we used the deep learning features of (Donahue et al., 2014) which have shown promising results for object recognition. Specifically, as visual features, we used the outputs derived from the activation of the 6th, 7th, and 8th layers of a deep convolutional network (CNN) with weights trained on the ImageNet data set (Deng et al., 2009), leading to 4096-dimensional $DeCAF_6$ and $DeCAF_7$ features, as well as 1000-dimensional $DeCAF_8$ features (Tommasi and Tuytelaars, 2014).

We used the conventional evaluation protocol introduced in (Saenko et al., 2010), 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[1]. With this protocol, we evaluated all possible combinations of source and target domains. For all the methods based on dimensionality reduction, we used the dimensionalities provided in the GFK code (i.e.,

[1] `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-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
SVMA (Duan et al., 2012)	34.77 ± 1.43	34.14 ± 3.70	32.47 ± 2.85	39.13 ± 2.02	34.52 ± 3.54	32.88 ± 2.27
DAM (Duan et al., 2012)	34.92 ± 1.46	34.27 ± 3.58	32.54 ± 2.72	39.20 ± 2.07	34.65 ± 3.53	33.05 ± 2.27
GFK (Gong et al., 2012)	37.2 ± 1.9	37.7 ± 3.3	38.9 ± 3.1	46.6 ± 2.7	36.8 ± 1.8	37.2 ± 4.0
TCA (Pan et al., 2011)	40 ± 1.3	39.1 ± 1.5	40.1 ± 1.2	46.7 ± 1.1	41.4 ± 1.2	36.2 ± 1.0
SA (Fernando et al., 2013)	41 ± 1.8	41.2 ± 5.0	42 ± 3.5	48.2 ± 3.1	50.3 ± 4.2	46.5 ± 4.9
KMM (Huang et al., 2006)	40.7 ± 2.1	39.8 ± 1.9	39.0 ± 3.7	48.6 ± 2.8	46.6 ± 3.3	42.2 ± 4.3
DME-MMD	43.3 ± 1.4	42.8 ± 2.5	46.7 ± 2.7	50 ± 3.2	49 ± 2.9	47.6 ± 3.5
DME-MMD (Poly)	43.1 ± 1.3	41.3 ± 2.7	45.6 ± 2.4	50.6 ± 2.9	47.8 ± 3.1	46.1 ± 3.1
DME-H	44.5 ± 1.7	43.2 ± 0.9	48.6 ± 2.3	51.9 ± 1.4	52.5 ± 2.9	47.3 ± 4.6
NL-DME-MMD	43.1 ± 1.9	44.4 ± 3.8	45.4 ± 4.0	50.4 ± 2.3	50.9 ± 3.1	49.6 ± 3.3
NL-DME-H	44.3 ± 1.3	47.2 ± 2.1	47.7 ± 2.8	52.1 ± 3.2	52.6 ± 2.8	48.8 ± 3.0

Table 1: Recognition accuracies on 6 pairs of source/target domains using the evaluation protocol of (Saenko et al., 2010) and using SURF features with Kernel SVM. C**: Caltech,** A**: Amazon,** W**: Webcam,** D**: DSLR**. The remaining pairs and the average accuracy over all pairs are shown in Table 2.

Method	$D \to A$	$D \to C$	$D \to W$	$W \to A$	$W \to C$	$W \to D$	Avg.
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	44
SVMA (Duan et al., 2012)	33.43 ± 1.24	31.40 ± 0.87	74.44 ± 2.21	36.63 ± 1.08	33.52 ± 0.77	74.97 ± 2.65	41.1
DAM (Duan et al., 2012)	33.50 ± 1.29	31.52 ± 0.88	74.68 ± 2.14	34.73 ± 1.14	31.18 ± 1.25	68.34 ± 3.16	40.2
GFK (Gong et al., 2012)	37.7 ± 1.8	33.3 ± 1.3	79.9 ± 2.8	41.5 ± 1.8	34.5 ± 0.9	76.7 ± 1.4	44.8
TCA (Pan et al., 2011)	39.6 ± 1.2	34 ± 1.1	80.4 ± 2.6	40.2 ± 1.1	33.7 ± 1.1	77.5 ± 2.5	42.8
SA (Fernando et al., 2013)	41.1 ± 1.6	35.4 ± 1.8	84.4 ± 2.4	38.2 ± 1.4	33.3 ± 1.2	83.3 ± 1.6	48.7
KMM (Huang et al., 2006)	38 ± 1.8	34.3 ± 1.2	82.0 ± 1.7	39.0 ± 1.2	35.3 ± 1.0	86.8 ± 2.0	47.7
DME-MMD	40.5 ± 1	39 ± 0.5	86.7 ± 1.2	42.5 ± 1.5	37 ± 0.9	86.4 ± 1.8	50.9
DME-MMD (Poly)	40.8 ± 0.9	39.1 ± 0.6	87.1 ± 1.0	41.3 ± 1.3	36.8 ± 0.9	85.8 ± 2.2	50.4
DME-H	39.1 ± 0.6	38.9 ± 0.4	88.6 ± 1.0	44.1 ± 0.8	39.9 ± 0.7	89.3 ± 0.5	52.3
NL-DME-MMD	40.1 ± 2.2	37.6 ± 0.6	87.3 ± 1.0	41.02 ± 1.3	36.7 ± 2.4	86.7 ± 2.2	51.1
NL-DME-H	41.6 ± 1.3	36.4 ± 0.4	87.5 ± 1.1	44.3 ± 1.1	38 ± 0.7	88.8 ± 1.6	$\mathbf{52.5}$

Table 2: Recognition accuracies on the remaining 6 pairs of source/target domains using the evaluation protocol of (Saenko et al., 2010) and using SURF features with Kernel SVM. C**: Caltech,** A**: Amazon,** W**: Webcam,** D**: DSLR**. As shown by the average accuracy over all pairs in the last columns, our approach clearly outperforms the baselines, with best performance for NL-DME-H.

W-D: 10, D-A: 20, W-A: 10, C-W: 20, C-D: 10, C-A: 20. Note that the dimensionality for X-Y is the same as for Y-X).

For the kernel SVM classifier, we report the results of our algorithms and several baselines on all source and target pairs in Tables 1 and 2 for the SURF features, in Tables 3 and 4 for the DeCAF6 features, in Tables 5 and 6 for the DeCAF7 features, and in Tables 7 and 8 for the DeCAF8 features. Similar results using a linear SVM classifier are provided in Tables 9 to 16. In the last column of every other table, we report the average accuracy over all the source-target pairs. Note that, with SURF features, all our algorithms clearly outperform the baselines, with the best average accuracy

Method	$A \to C$	$A \to D$	$A \to W$	$C \to A$	$C \to D$	$C \to W$
NO ADAPT-SVM	81.6 ± 1.4	82.6 ± 3.4	74.6 ± 3.3	89.8 ± 1.5	84.3 ± 2.6	77.8 ± 1.7
SVMA (Duan et al., 2012)	83.54	81.72	74.58	91.00	83.89	76.61
DAM (Duan et al., 2012)	84.73	82.48	78.14	91.8	84.59	79.39
GFK (Gong et al., 2012)	84.8 ± 1.0	89.3 ± 2.4	84.6 ± 2.1	90.9 ± 1.0	87.1 ± 1.5	85.2 ± 1.5
TCA(Pan et al., 2011)	82.9 ± 0.9	89 ± 2.0	77.8 ± 3.9	89.9 ± 1.7	87.8 ± 2.1	82.9 ± 2.0
SA (Fernando et al., 2013)	86.1 ± 0.9	80.1 ± 1.0	75.3 ± 0.8	91.5 ± 0.7	85.6 ± 4.1	85.1 ± 0.9
KMM (Huang et al., 2006)	83.7 ± 1.4	86.7 ± 2.8	75.4 ± 4.6	90.4 ± 1.3	85.03 ± 3.1	78 ± 3.3
DME-MMD	84.3 ± 1.0	89.2 ± 3.7	81.5 ± 4.3	90.3 ± 0.6	87.3 ± 2.3	84.7 ± 3.4
DME-H	85.1 ± 1.4	89.5 ± 2.5	80.9 ± 3.4	90.9 ± 1.1	88.1 ± 2.1	83.2 ± 2.7
NL-DME-MMD	85.3 ± 1.0	89.3 ± 3.0	79.8 ± 3.1	91.4 ± 0.7	87.9 ± 2.3	82.2 ± 2.7
NL-DME-H	85.3 ± 1.1	91.1 ± 2.1	83.8 ± 3.6	91.6 ± 0.9	86.8 ± 3.3	85.2 ± 2.9

Table 3: Recognition accuracies on 6 pairs of source/target domains using the evaluation protocol of (Saenko et al., 2010) and using Decaf6 features with Kernel SVM. C**: Caltech,** A**: Amazon,** W**: Webcam,** D**: DSLR**. The remaining pairs and the average accuracy over all pairs are shown in Table 4.

Method	$D \to A$	$D \to C$	$D \to W$	$W \to A$	$W \to C$	$W \to D$	Avg.
NO ADAPT-SVM	79.2 ± 2.3	73.4 ± 2.0	95.6 ± 1.1	75.3 ± 1.5	69.5 ± 1.1	99.4 ± 0.6	81.9
SVMA (Duan et al., 2012)	85.37	78.14	96.71	74.36	70.58	96.6	82.7
DAM (Duan et al., 2012)	87.88	81.27	96.31	76.6	74.32	93.8	84.2
GFK (Gong et al., 2012)	84.2 ± 2.3	77.5 ± 2.0	96.4 ± 1.1	85.4 ± 1.7	77.1 ± 0.5	99.5 ± 0.3	86.8
TCA (Pan et al., 2011)	84.1 ± 1.6	77.7 ± 1.9	95.9 ± 0.8	83.8 ± 1.0	76.5 ± 0.9	98.6 ± 0.9	85.6
SA (Fernando et al., 2013)	90.1 ± 0.9	83.9 ± 1.6	96.8 ± 1.6	85.0 ± 3.3	78.7 ± 2.8	99.3 ± 0.7	86.5
KMM (Huang et al., 2006)	84.3 ± 2.4	77.4 ± 1.1	96.2 ± 1.8	75.5 ± 3.2	72.8 ± 1.9	97.9 ± 0.9	83.6
DME-MMD	82.9 ± 2.9	77.5 ± 2.7	96.4 ± 1.2	82.1 ± 1.9	78.6 ± 1.4	98.8 ± 0.3	86.2
DME-H	84.5 ± 2.5	79.6 ± 1.8	97 ± 0.9	83.9 ± 1.1	77.9 ± 1.4	99.7 ± 0.4	86.7
NL-DME-MMD	86.4 ± 2.2	76.01 ± 2.9	97.7 ± 1.3	84.3 ± 1.4	77.3 ± 1.5	98.6 ± 0.7	86.4
NL-DME-H	86.3 ± 2.6	82.2 ± 2.6	98.1 ± 1.4	86.1 ± 1.6	78.1 ± 1.6	99.3 ± 1.0	**87.9**

Table 4: Recognition accuracies on the remaining 6 pairs of source/target domains using the evaluation protocol of (Saenko et al., 2010) and using Decaf6 features with Kernel SVM. C**: Caltech,** A**: Amazon,** W**: Webcam,** D**: DSLR**. As shown by the average accuracy over all pairs in the last columns, NL-DME-H still yields the best accuracy.

achieved by NL-DME-H. To the best of our knowledge, this represents the state-of-the-art result on this data. With deep learning features, our algorithms yield accuracies that are on par with the state-of-the-art results. Note that, among our algorithms, the best results are achieved using Decaf7 features with NL-DME-H. In Fig. 4, we illustrate the behavior of our algorithms on the *mug* class. As shown in the bottom-right panel, even humans would have a hard time to correctly label some of the misclassified examples.

We then further compare the results of our approach and the previous baselines with two recent deep learning DA methods, Deep Domain Confusion (DDC) (Tzeng et al., 2014) and Reverse Gradient (RG) (Ganin and Lempitsky, 2015). Table 17 was computed from the Office-Caltech data set with 10 classes, as in the previous experiments. In this setting, only the results of DDC for 6 pairs

Method	$A \rightarrow C$	$A \rightarrow D$	$A \rightarrow W$	$C \rightarrow A$	$C \rightarrow D$	$C \rightarrow W$
NO ADAPT-SVM	84.2 ± 0.8	87.9 ± 1.8	77.5 ± 1.5	89.9 ± 1.2	84.9 ± 3.5	78.9 ± 2.6
SVMA (Duan et al., 2012)	84.5 ± 1.3	84.9 ± 3.2	74.8 ± 4.41	91.8 ± 0.70	83.5 ± 2.3	78.5 ± 4.0
DAM (Duan et al., 2012)	85.5 ± 1.2	84.0 ± 5.0	77.4 ± 4.55	92.2 ± 0.6	83.5 ± 2.7	81.1 ± 3.9
GFK (Gong et al., 2012)	85.8 ± 0.5	91 ± 2.2	83.7 ± 2.4	91.3 ± 0.9	88.3 ± 1.7	85.7 ± 2.4
SA (Fernando et al., 2013)	86.4 ± 0.8	91.4 ± 3.2	87.8 ± 2.5	92.2 ± 0.7	89.0 ± 3.2	88.9 ± 2.1
TCA(Pan et al., 2011)	84.6 ± 0.8	89.6 ± 2.8	82.3 ± 4.2	89.8 ± 1.2	87.3 ± 3.5	83.7 ± 3.6
KMM(Huang et al., 2006)	85.7 ± 0.7	86.8 ± 1.4	76.5 ± 1.6	91.3 ± 0.6	85.3 ± 2.8	79.8 ± 4.1
DME-MMD	85.4 ± 0.7	91.1 ± 1.3	81.5 ± 1.6	91.4 ± 0.4	88.03 ± 1.9	82.6 ± 2.3
DME-H	85.6 ± 0.9	89.3 ± 2.2	80 ± 2.2	91.8 ± 0.6	86.4 ± 2.7	83.6 ± 3.01
NL-DME-MMD	85.7 ± 0.6	89.2 ± 2.5	83.6 ± 2.3	91.6 ± 0.3	88.9 ± 2.7	85.6 ± 2.3
NL-DME-H	86.7 ± 0.9	89.4 ± 2.0	79.7 ± 2.5	91.8 ± 0.5	89.6 ± 2.3	83.8 ± 2.04

Table 5: Recognition accuracies on 6 pairs of source/target domains using the evaluation protocol of (Saenko et al., 2010) and using Decaf7 features with Kernel SVM. C**: Caltech,** A**: Amazon,** W**: Webcam,** D**: DSLR**. The remaining pairs and the average accuracy over all pairs are shown in Table 6.

Method	$D \rightarrow A$	$D \rightarrow C$	$D \rightarrow W$	$W \rightarrow A$	$W \rightarrow C$	$W \rightarrow D$	Avg.
NO ADAPT-SVM	84.0 ± 1.9	77.9 ± 1.1	96.6 ± 1.6	82.9 ± 1.1	74.4 ± 0.7	99.1 ± 0.6	84.8
SVMA (Duan et al., 2012)	87.6 ± 1.1	80.8 ± 0.9	96.0 ± 1.5	81.0 ± 1.6	77.0 ± 0.7	98.4 ± 1.0	84.9
DAM (Duan et al., 2012)	90.1 ± 1.1	83.1 ± 1.2	95.3 ± 1.3	81.7 ± 3.4	77.8 ± 2.4	95.6 ± 2.2	85.6
GFK (Gong et al., 2012)	85.9 ± 1.9	80.7 ± 1.3	96.6 ± 1.3	89.2 ± 1.0	78.8 ± 0.6	99.6 ± 0.5	88.1
SA (Fernando et al., 2013)	89.8 ± 0.7	83.7 ± 1.8	97.1 ± 0.8	88.3 ± 2.2	83.4 ± 0.7	99.6 ± 0.3	$\mathbf{89.8}$
TCA(Pan et al., 2011)	86.0 ± 1.6	80.5 ± 1.1	95.6 ± 1.8	90.1 ± 0.8	78.7 ± 1.0	98.3 ± 0.7	87.2
KMM(Huang et al., 2006)	87.9 ± 1.2	81.5 ± 1.1	96.7 ± 1.2	83.9 ± 1.9	77.8 ± 1.2	98.4 ± 0.8	86.0
DME-MMD	85.6 ± 1.8	80.8 ± 2.8	96.5 ± 0.7	83.6 ± 2.2	77.3 ± 2.5	99.4 ± 0.4	86.9
DME-H	88.3 ± 1.6	78.9 ± 2.1	97 ± 1.7	85.1 ± 2.3	78.9 ± 1.6	99.3 ± 0.4	87.01
NL-DME-MMD	86.2 ± 1.1	82.5 ± 0.9	96.7 ± 1.3	89 ± 0.8	78.3 ± 0.6	99.6 ± 0.3	88.1
NL-DME-H	89.5 ± 0.8	82.8 ± 2.3	97.3 ± 1	90.1 ± 3.1	79.8 ± 2.0	99.2 ± 0.7	88.3

Table 6: Recognition accuracies on the remaining 6 pairs of source/target domains using the evaluation protocol of (Saenko et al., 2010) and using Decaf7 features with Kernel SVM. C**: Caltech,** A**: Amazon,** W**: Webcam,** D**: DSLR**. As shown by the average accuracy over all pairs in the last columns, NL-DME-H still yields competitive accuracy.

were available. Note that our approach yields slightly better accuracies than DDC. In Table 18, we compare our results with both deep learning baselines using the 31 classes of the original Office data set, and for the 3 pairs that were reported in (Ganin and Lempitsky, 2015). Note that, here, while our approach is among the top performer non-deep-learning methods, the two works that jointly learn the features and perform domain adaptation tend to perform better. This suggests an interesting avenue for future research by incorporating our Hellinger-based metric within a deep learning framework.

Method	$A \to C$	$A \to D$	$A \to W$	$C \to A$	$C \to D$	$C \to W$
NO ADAPT-SVM	75.3 ± 1.4	83.5 ± 2.2	75.6 ± 2.1	77.6 ± 1.3	81.3 ± 1.3	73.1 ± 1.4
SVMA (Duan et al., 2012)	76.3 ± 0.8	85.1 ± 1.1	75.8 ± 2.3	79.5 ± 0.6	82.1 ± 1.3	73.3 ± 3.1
DAM (Duan et al., 2012)	77.1 ± 1.1	85.4 ± 1.2	77.8 ± 2.5	79.9 ± 0.6	84.2 ± 1.0	75.4 ± 2.7
GFK (Gong et al., 2012)	76.1 ± 2.0	86.4 ± 1.9	78.9 ± 2.4	79.4 ± 0.9	82.4 ± 1.1	76.3 ± 2.2
SA (Fernando et al., 2013)	77.2 ± 1.4	84.6 ± 1.2	78.8 ± 1.5	78.9 ± 1.2	85.1 ± 1.4	73.7 ± 2.2
TCA(Pan et al., 2011)	75.9 ± 1.6	86.7 ± 1.7	75.7 ± 1.5	79.7 ± 0.8	82.5 ± 1.0	75.6 ± 2.3
KMM(Huang et al., 2006)	77.3 ± 1.1	83.1 ± 1.6	75.8 ± 2.2	79.9 ± 1.0	81.4 ± 2.1	77.3 ± 1.7
DME-MMD	75.6 ± 0.8	86.2 ± 2.1	75.4 ± 1.9	78.9 ± 1.0	81.8 ± 1.3	73.5 ±1.5
DME-H	77.5 ± 1.0	87.6 ± 1.9	75.8 ± 2.8	79.4 ± 0.8	83.5 ± 1.0	74.5 ± 3.1
NL-DME-MMD	76.5 ± 1.2	86.9 ± 2.6	78.2 ± 2.5	80.3 ± 1.2	81.9 ± 1.7	77.9 ± 1.5
NL-DME-H	78.9 ± 1.2	87.4 ± 1.7	79.3 ± 2.4	81 ± 0.9	82.4 ± 0.8	77.3 ± 1.6

Table 7: Recognition accuracies on 6 pairs of source/target domains using the evaluation protocol of (Saenko et al., 2010) and using Decaf8 features with Kernel SVM. C**: Caltech,** A**: Amazon,** W**: Webcam,** D**: DSLR**. The remaining pairs and the average accuracy over all pairs are shown in Table 8.

Method	$D \to A$	$D \to C$	$D \to W$	$W \to A$	$W \to C$	$W \to D$	Avg.
NO ADAPT-SVM	80.3 ± 0.9	73.6 ± 0.8	81.4 ± 1.6	84.1 ± 0.5	78.3 ± 1.2	93.3 ± 0.8	79.7
SVMA (Duan et al., 2012)	81.8 ± 2.1	75.0 ± 1.1	83.1 ± 1.0	83.5 ± 0.4	77.4 ± 0.6	93.5 ± 1.0	80.5
DAM (Duan et al., 2012)	82.1 ± 2.1	75.7 ± 1.1	83.5 ± 0.8	82.0 ± 1.7	78.8 ± 1.8	90.2 ± 1.5	81.1
GFK (Gong et al., 2012)	81.7 ± 0.6	74.9 ± 1.4	82.3 ± 1.5	83.1 ± 0.4	80.2 ± 1.6	93.6 ± 1.1	81.3
SA (Fernando et al., 2013)	82.4 ± 0.8	77.9 ± 1.7	83.4 ± 1.3	82.3 ± 0.5	81.0 ± 1.6	90.7 ± 2.1	81.3
TCA(Pan et al., 2011)	79.7 ± 2.4	74.4 ± 1.7	81.1 ± 1.9	82.9 ± 0.6	78.9 ± 1.0	93.6 ± 1.4	80.6
KMM(Huang et al., 2006)	79.5 ± 2.0	74.5 ± 1.8	82.3 ± 1.9	84.3 ± 0.6	78.8 ± 1.1	93.4 ± 0.9	80.6
DME-MMD	82.6 ± 0.5	74.5 ± 1.7	80.2 ± 1.5	83.3 ± 0.7	78.4 ± 1.0	95.2 ± 1.1	80.5
DME-H	80.7 ± 1.5	75.4 ± 1.1	82.8 ± 1.3	83.6 ± 1.2	79.6 ± 1.2	95.3 ± 0.5	81.3
NL-DME-MMD	83.3 ± 0.9	75.6 ± 1.5	81.4 ± 1.0	82.8 ± 0.5	78.3 ± 1.3	93.7 ±1.5	81.4
NL-DME-H	82.9 ± 0.9	76.7 ± 1.7	84.9 ± 1.6	83.2 ± 0.9	79.9 ± 1.3	94.6 ± 0.8	**82.4**

Table 8: Recognition accuracies on the remaining 6 pairs of source/target domains using the evaluation protocol of (Saenko et al., 2010) and using Decaf8 features with Kernel SVM. C**: Caltech,** A**: Amazon,** W**: Webcam,** D**: DSLR**. As shown by the average accuracy over all pairs in the last columns, NL-DME-H still yields the best accuracy.

6.2 Cross-domain Text Categorization

As a second type of experiment, we made use of the 20 Newsgroups data set, which has become popular in the machine learning community to evaluate methods tackling problems such as text classification and text clustering. The 20 Newsgroups data set is a collection of 18,774 newsgroup documents organized in a hierarchical structure of six main categories and 20 subcategories, each corresponding to a different topic. Some of the newsgroups (from the same category) are very closely related to each other (e.g., comp.sys.ibm.pc.hardware / comp.sys.mac.hardware), while others are highly unrelated (e.g., misc.forsale / soc.religion.christian), making this data set well-suited to evaluate cross-domain learning algorithms.

For this experiment, we used the protocol of (Duan et al., 2012): The four largest main categories (comp, rec, sci, and talk) were chosen for evaluation. Specifically, for each main category, the largest

Method	$A \to C$	$A \to D$	$A \to W$	$C \to A$	$C \to D$	$C \to W$
NO ADAPT-SVM	38.3 ± 1.8	35.9 ± 2.6	38.1 ± 2.4	44.7 ± 2.1	40.7 ± 3.3	37.4 ± 1.9
SVMA (Duan et al., 2012)	34.77 ± 1.43	34.14 ± 3.70	32.47 ± 2.85	39.13 ± 2.02	34.52 ± 3.54	32.88 ± 2.27
DAM (Duan et al., 2012)	34.92 ± 1.46	34.27 ± 3.58	32.54 ± 2.72	39.20 ± 2.07	34.65 ± 3.53	33.05 ± 2.27
GFK (Gong et al., 2012)	41.1 ± 1.2	40.5 ± 4.3	42.5 ± 4.7	47.9 ± 3.5	47.8 ± 2.0	44.4 ± 3.8
TCA (Pan et al., 2011)	37 ± 2.5	36.6 ± 3.2	33.2 ± 4.4	42.9 ± 2.7	39.5 ± 3.2	36.03 ± 4.3
SA (Fernando et al., 2013)	40.6 ± 1.7	39.2 ± 2.6	37.6 ± 4.5	49.4 ± 3.1	48.7 ± 3.6	43.4 ± 2.9
KMM (Huang et al., 2006)	38.3 ± 1.8	36 ± 2.6	38.1 ± 2.4	44.7 ± 2.1	40.7 ± 3.3	37.4 ± 1.9
DME-MMD	43.1 ± 1.9	39.9 ± 4.0	41.7 ± 3.2	45.9 ± 3.1	43.7 ± 3.8	45.7 ± 3.6
DME-H	42.7 ± 1.9	40.3 ± 3.8	42 ± 2.7	46.7 ± 2.4	44.1 ± 2.6	45.2 ± 3.1
NL-DME-MMD	42.1 ± 1.3	41.5 ± 3.5	40.8 ± 3.8	48.5 ± 2.3	45 ± 4.5	47.3 ± 4.5
NL-DME-H	41.5 ± 1.6	41.7 ± 3.4	40.5 ± 3.0	49.4 ± 3.4	46.1 ± 3.8	47.1 ± 3.8

Table 9: Recognition accuracies on 6 pairs of source/target domains using the evaluation protocol of (Saenko et al., 2010) and using SURF features with Linear SVM. C**: Caltech,** A**: Amazon,** W**: Webcam,** D**: DSLR**. The remaining pairs and the average accuracy over all pairs are shown in Table 10.

Method	$D \to A$	$D \to C$	$D \to W$	$W \to A$	$W \to C$	$W \to D$	Avg.
NO ADAPT-SVM	33.8 ± 2.0	31.4 ± 1.3	75.5 ± 2.2	36.6 ± 1.1	32 ± 1.1	80.5 ± 1.4	43.7
SVMA (Duan et al., 2012)	33.4 ± 1.2	31.4 ± 0.9	74.4 ± 2.2	36.6 ± 1.1	33.5 ± 0.8	74.9 ± 2.7	41.1
DAM (Duan et al., 2012)	33.5 ± 1.3	31.5 ± 0.9	74.6 ± 2.1	34.7 ± 1.1	31.1 ± 1.3	68.3 ± 3.2	40.2
GFK (Gong et al., 2012)	36.9 ± 2.9	34.3 ± 1.7	77.3 ± 2.2	41 ± 1.7	34.3 ± 1.1	75.8 ± 2.8	47
TCA (Pan et al., 2011)	34.4 ± 1.9	33.2 ± 1.7	76.4 ± 2.3	38.3 ± 2.2	33.8 ± 1.6	57.8 ± 5.2	41.6
SA (Fernando et al., 2013)	37.7 ± 2.2	33.9 ± 1.3	74.8 ± 4.5	39.4 ± 1.2	35.3 ± 1.6	76.6 ± 3.9	46.4
KMM (Huang et al., 2006)	33.8 ± 2.0	31.4 ± 1.3	75.5 ± 2.2	36.6 ± 1.1	32.1 ± 1.1	80.5 ± 1.4	43.8
DME-MMD	38.1 ± 2.7	32.8 ± 1.1	74.9 ± 1.9	42.3 ± 2.8	35.2 ± 1.2	74.5 ± 2.1	46.4
DME-H	37.4 ± 1.9	34.6 ± 1.7	74.3 ± 1.4	41.3 ± 0.9	35 ± 1.4	73.9 ± 2.2	46.5
NL-DME-MMD	36.9 ± 2.2	32.5 ± 1.3	76.8 ± 3.1	40.4 ± 1.9	35.7 ± 0.7	79.5 ± 3.2	47.2
NL-DME-H	38.5 ± 1.6	34.8 ± 1.7	73.6 ± 2.0	42.1 ± 1.1	35.8 ± 0.8	78.4 ± 3.0	**47.5**

Table 10: Recognition accuracies on the remaining 6 pairs of source/target domains using the evaluation protocol of (Saenko et al., 2010) and using SURF features with Linear SVM. C**: Caltech,** A**: Amazon,** W**: Webcam,** D**: DSLR**. As shown by the average accuracy over all pairs in the last columns, our approach outperforms the baselines, with best performance for NL-DME-H.

subcategory was selected as the target domain. We considered the largest category ”comp” as the positive class and one of the three other categories as the negative class for each setting. Table 19 provides detailed information about the three settings. We used word-frequency features to represent each document. To construct the training set, we used 1000 randomly selected samples (evenly distributed positive and negative samples) from the source domain, and repeated this procedure 5 times. For each such partition, our mappings were then learned using this training data and the unlabeled samples from the target domain.

In Table 20, we report the mean recognition accuracies of our algorithms and of state-of-the-art baselines. For all the baselines, we employed a kernel SVM classifier with a degree 2 polynomial kernel, we tested with the regularizer weight $C \in \{10^{-5}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$, and report the best result. For all the baselines based on dimensionality reduction, we set the dimen-

Method	$A \to C$	$A \to D$	$A \to W$	$C \to A$	$C \to D$	$C \to W$
NO ADAPT-SVM	83.4 ± 0.8	85.4 ± 3.0	77.1 ± 2.5	90.5 ± 1.2	84.2 ± 2.9	77.4 ± 2.8
SVMA (Duan et al., 2012)	83.54	81.72	74.58	91.00	83.89	76.61
DAM (Duan et al., 2012)	84.73	82.48	78.14	91.8	84.59	79.39
GFK (Gong et al., 2012)	85.2 ± 0.9	86.8 ± 1.0	80.8 ± 1.8	91.3 ± 1.2	84.7 ± 1.7	82.8 ± 2.4
TCA(Pan et al., 2011)	85.5 ± 1.2	87.4 ± 3.6	80.6 ± 1.8	91.2 ± 1.2	84.9 ± 2.1	81.0 ± 1.9
SA (Fernando et al., 2013)	84.9 ± 0.7	87.0 ± 3.9	79.6 ± 5.7	91.6 ± 1.0	86.6 ± 2.7	84 ± 1.8
KMM (Huang et al., 2006)	83.6 ± 0.7	85.2 ± 3.0	75.7 ± 2.9	90.5 ± 1.2	84.5 ± 3.1	77.0 ± 2.2
DME-MMD	84.3 ± 0.8	83.6 ± 3.7	76.4 ± 1.5	88 ± 1.9	84.4 ± 3.4	77.4 ± 6.4
DME-H	84.7 ± 0.8	86.1 ± 1.1	77.6 ± 2.5	91.3 ± 1.7	85.2 ± 3.3	77.8 ± 3.3
NL-DME-MMD	84.8 ± 0.9	86.2 ± 0.8	78 ± 5.2	91.5 ± 1.0	85.9 ± 2.3	78.6 ± 3.9
NL-DME-H	83.6 ± 1.2	83.4 ± 4.4	77.03 ± 2.7	90.4 ± 1.6	85 ± 2.5	81.3 ± 3.6

Table 11: Recognition accuracies on 6 pairs of source/target domains using the evaluation protocol of (Saenko et al., 2010) and using Decaf6 features with Linear SVM. C**: Caltech,** A**: Amazon,** W**: Webcam,** D**: DSLR**. The remaining pairs and the average accuracy over all pairs are shown in Table 12.

Method	$D \to A$	$D \to C$	$D \to W$	$W \to A$	$W \to C$	$W \to D$	Avg.
NO ADAPT-SVM	85.5 ± 2.4	77.1 ± 2.0	94.1 ± 1.7	76.6 ± 2.8	70.5 ± 1.2	98.9 ± 0.8	83.3
SVMA (Duan et al., 2012)	85.37	78.14	96.71	74.36	70.58	96.6	82.7
DAM (Duan et al., 2012)	87.88	81.27	96.31	76.6	74.32	93.8	84.2
GFK (Gong et al., 2012)	88.2 ± 1.2	80.7 ± 1.9	97.4 ± 1.5	88 ± 0.9	79.5 ± 1.4	98.9 ± 0.8	87
TCA (Pan et al., 2011)	89.9 ± 0.9	82.7 ± 2.1	95.9 ± 1.5	86.7 ± 1.5	77.8 ± 2.8	99.2 ± 0.4	86.9
SA (Fernando et al., 2013)	91.2 ± 0.4	84.2 ± 1.2	97.4 ± 1.2	89.6 ± 0.6	79.9 ± 1.1	99.5 ± 0.4	**87.9**
KMM (Huang et al., 2006)	85.6 ± 1.5	77.4 ± 2.3	97.1 ± 1.7	76.0 ± 2.5	72.2 ± 1.4	98.4 ± 1.0	83.6
DME-MMD	88.2 ± 1.1	85.2 ± 2.1	95.5 ± 0.9	87.6 ± 1.8	74.9 ± 1.4	98.8 ± 0.8	85.4
DME-H	86.4 ± 1.7	85.6 ± 0.6	95.1 ± 1.3	87.3 ± 1.3	73.5 ± 0.9	98 ± 1.0	85.7
NL-DME-MMD	86.8 ± 1.5	84.2 ± 1.1	97.2 ± 1.1	86.8 ± 2.4	77.2 ± 1.5	99.4 ± 0.7	86.3
NL-DME-H	91.7 ± 0.8	85 ± 1.3	96.1 ± 1.1	90.6 ± 2.2	82.5 ± 1.7	99.0 ± 0.9	87.1

Table 12: Recognition accuracies on the remaining 6 pairs of source/target domains using the evaluation protocol of (Saenko et al., 2010) and using Decaf6 features with Linear SVM. C**: Caltech,** A**: Amazon,** W**: Webcam,** D**: DSLR**. As shown by the average accuracy over all pairs in the last columns, NL-DME-H still yields competitive accuracy.

sionalities based on the Subspace Disagreement Measure (SDM) (Gong et al., 2012)[2] (i.e., comp vs. rec: 10, comp vs. sci: 27, comp vs. talk: 47). Similarly to the visual recognition task, our algorithms achieve state-of-the-art results. Again, NL-DME-H yields the best average accuracy of all methods.

6.3 Cross-domain WiFi Localization

To evaluate our approach on a different domain adaptation task, we used the WiFi data set published in the 2007 IEEE ICDM Contest for domain adaptation (Yang et al., 2008). The goal here is to estimate the location of mobile devices based on the received signal strength (RSS) values from

[2] The code can be downloaded from: `http://users.cecs.anu.edu.au/~basura/DA_SA/getGFKDim.m`

Method	$A \to C$	$A \to D$	$A \to W$	$C \to A$	$C \to D$	$C \to W$
NO ADAPT-SVM	85.4 ± 0.8	87.3 ± 1.6	76.8 ± 3.2	90.9 ± 0.9	86.1 ± 3.2	77.5 ± 4.3
SVMA (Duan et al., 2012)	84.5 ± 1.3	84.9 ± 3.2	74.8 ± 4.41	91.8 ± 0.70	83.5 ± 2.3	78.5 ± 4.0
DAM (Duan et al., 2012)	85.5 ± 1.2	84.0 ± 5.0	77.4 ± 4.55	92.2 ± 0.6	83.5 ± 2.7	81.1 ± 3.9
GFK (Gong et al., 2012)	86.2 ± 0.8	89.6 ± 2.1	81.1 ± 2.5	91.2 ± 0.8	89.5 ± 2.3	84.6 ± 1.8
SA (Fernando et al., 2013)	86.3 ± 0.7	90.2 ± 3.1	87.9 ± 2.2	91.4 ± 0.6	88.4 ± 4.3	85.8 ± 3.1
TCA(Pan et al., 2011)	86.5 ± 0.8	90.1 ± 3.0	80.5 ± 3.0	91.7 ± 0.5	86.3 ± 3.8	84.6 ± 2.9
KMM(Huang et al., 2006)	85.5 ± 0.7	87.1 ± 2.4	76.5 ± 4.1	91.4 ± 0.7	86.3 ± 2.7	78.2 ± 4.9
DME-MMD	85.6 ± 0.8	89.2 ± 2.9	78.1 ± 3.2	91.1 ± 0.8	88.2 ± 2.3	82.5 ± 2.9
DME-H	86.1 ± 1.1	88.5 ± 2.0	78.9 ± 4.3	92.2 ± 0.5	88.6 ± 2.9	82 ± 3.4
NL-DME-MMD	86.2 ± 0.9	88.3 ± 3.2	79.9 ± 5.3	91.7 ± 0.53	87.5 ± 2.8	82.5 ± 2.1
NL-DME-H	85.3 ± 1.0	89.4 ± 3.8	83.3 ± 3.5	91.3 ± 0.8	88.4 ± 2.9	83 ± 0.4

Table 13: Recognition accuracies on 6 pairs of source/target domains using the evaluation protocol of (Saenko et al., 2010) and using Decaf7 features with Linear SVM. C: **Caltech,** A: **Amazon,** W: **Webcam,** D: **DSLR**. The remaining pairs and the average accuracy over all pairs are shown in Table 14.

Method	$D \to A$	$D \to C$	$D \to W$	$W \to A$	$W \to C$	$W \to D$	Avg.
NO ADAPT-SVM	88.3 ± 1.2	80.4 ± 1.0	97.3 ± 1.2	82.1 ± 1.1	76.1 ± 0.6	95.2 ± 0.9	85.2
SVMA (Duan et al., 2012)	87.6 ± 1.1	80.8 ± 0.9	96.0 ± 1.5	81.0 ± 1.6	77.0 ± 0.7	98.4 ± 1.0	84.9
DAM (Duan et al., 2012)	90.1 ± 1.1	83.1 ± 1.2	95.3 ± 1.3	81.7 ± 3.4	77.8 ± 2.4	95.6 ± 2.2	85.6
GFK (Gong et al., 2012)	89.4 ± 0.9	81.6 ± 1.1	97.3 ± 1.0	89 ± 0.9	83.6 ± 1.1	99.0 ± 0.5	88.5
SA (Fernando et al., 2013)	90 ± 0.4	82.9 ± 1.2	97.0 ± 0.9	88.2 ± 1.7	82.7 ± 0.8	98.7 ± 1.6	$\mathbf{89.1}$
TCA(Pan et al., 2011)	91.7 ± 0.4	84 ± 0.9	97 ± 0.9	90.6 ± 1.6	81.1 ± 0.8	98.6 ± 0.9	88.5
KMM(Huang et al., 2006)	87.9 ± 1.4	80.3 ± 0.8	97 ± 1.4	82.4 ± 1.9	77.5 ± 1.4	98.5 ± 0.9	85.7
DME-MMD	88.5 ± 1.1	83.1 ± 2.6	96.6 ± 1.6	86.1 ± 3.1	79.3 ± 0.9	98.3 ± 0.9	87.2
DME-H	89 ± 1.1	82.4 ± 2.4	96.8 ± 1.0	89.7 ± 3.1	79.5 ± 2.0	98.1 ± 1.0	87.7
NL-DME-MMD	88.5 ± 1.3	84.8 ± 0.9	97.4 ± 0.8	89.5 ± 0.7	78.7 ± 1.0	98.9 ± 0.4	87.8
NL-DME-H	91.3 ± 1.0	85.8 ± 1.0	96.8 ± 1.5	90.9 ± 1.3	79.6 ± 1.7	98.2 ± 1.0	88.6

Table 14: Recognition accuracies on the remaining 6 pairs of source/target domains using the evaluation protocol of (Saenko et al., 2010) and using Decaf7 features with Linear SVM. C: **Caltech,** A: **Amazon,** W: **Webcam,** D: **DSLR**. As shown by the average accuracy over all pairs in the last columns, NL-DME-H still yields competitive accuracy.

different access points. The different domains represent two different time periods during which the collected RSS values may have different distributions. The data set contains 621 labeled examples collected during time period A (i.e., the source) and 3128 unlabeled examples collected during time period B (i.e., the target). We followed the transductive setting of Pan et al. (2011), which uses all the samples from the source and 400 random samples from the target.

In this case, we report the mean Average Error Distance (AED) over 10 random selections of target samples. The AED is computed as $AED = \frac{\sum_i l(\boldsymbol{x}_i)-y_i}{N}$, where $\boldsymbol{x}_i$ is a vector of RSS values, $l(\boldsymbol{x}_i)$ is the predicted location and y_i the corresponding ground-truth location. Note that, here, all results were obtained with a nearest-neighbor classifier to follow the procedure of (Pan et al., 2011). Fig. 5 depicts the accuracy as a function of the dimensionality of the learned subspace for several

Method	$A \to C$	$A \to D$	$A \to W$	$C \to A$	$C \to D$	$C \to W$
NO ADAPT-SVM	73 ± 1.4	83.9 ± 1.9	71.5 ± 2.6	77 ± 0.5	80.7 ± 1.8	71.8 ± 1.3
SVMA (Duan et al., 2012)	76.3 ± 0.8	85.1 ± 1.1	75.8 ± 2.3	79.5 ± 0.6	82.1 ± 1.3	73.3 ± 3.1
DAM (Duan et al., 2012)	77.1 ± 1.1	85.4 ± 1.2	77.8 ± 2.5	79.9 ± 0.6	84.2 ± 1.0	75.4 ± 2.7
GFK (Gong et al., 2012)	73.9 ± 1.3	85.2 ± 2.1	70.1 ± 1.7	79.2 ± 0.7	80.2 ± 1.9	70.5 ± 2.0
SA (Fernando et al., 2013)	75.6 ± 1	83.7 ± 1.1	72.8 ± 2.3	78.6 ± 1.0	78.5 ± 1.4	69.2 ± 2.3
TCA(Pan et al., 2011)	72.7 ± 1.5	85.3 ± 1.2	71.7 ± 1.4	79.1 ± 1.0	80.7 ± 1.6	71.3 ± 1.7
KMM(Huang et al., 2006)	75.1 ± 1.8	83.1 ± 1.5	74.3 ± 2.2	78.6 ± 1.7	81 ± 1.8	74.2 ± 2.7
DME-MMD	76.4 ± 1.6	85.4 ± 2.8	76.6 ± 3.4	78.5 ± 0.7	79.9 ± 2.1	71.9 ± 2.4
DME-H	75.3 ± 1.3	83.8 ± 1.9	74.4 ± 1.2	78.9 ± 1.2	78.3 ± 1.9	73.6 ± 1.9
NL-DME-MMD	74.9 ± 1.3	86 ± 2.1	78 ± 1.7	79.4 ± 1.4	81.2 ± 2.0	73.4 ± 1.9
NL-DME-H	74.4 ± 1.2	84.4 ± 1.6	75.2 ± 3.0	79.6 ± 1.3	78.9 ± 4.3	74 ± 2.5

Table 15: Recognition accuracies on 6 pairs of source/target domains using the evaluation protocol of (Saenko et al., 2010) and using Decaf8 features with Linear SVM. C**: Caltech,** A**: Amazon,** W**: Webcam,** D**: DSLR**. The remaining pairs and the average accuracy over all pairs are shown in Table 16.

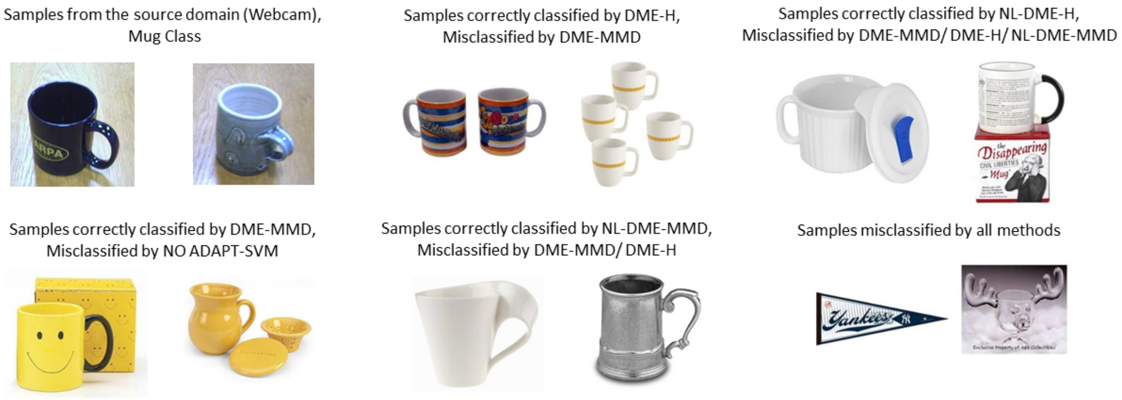
Method	$D \to A$	$D \to C$	$D \to W$	$W \to A$	$W \to C$	$W \to D$	Avg.
NO ADAPT-SVM	80.6 ± 0.9	72.7 ± 1.1	71.4 ± 1.7	81.4 ± 1.3	76.1 ± 0.6	91.8 ± 0.9	77.6
SVMA (Duan et al., 2012)	81.8 ± 2.1	75.0 ± 1.1	83.1 ± 1.0	83.5 ± 0.4	77.4 ± 0.6	93.5 ± 1.0	80.5
DAM (Duan et al., 2012)	82.1 ± 2.1	75.7 ± 1.1	83.5 ± 0.8	82.0 ± 1.7	78.8 ± 1.8	90.2 ± 1.5	**81.1**
GFK (Gong et al., 2012)	82.5 ± 0.7	73.7 ± 1.7	78.1 ± 1.8	81.9 ± 1.3	80.6 ± 1.1	91.2 ± 1.5	78.9
SA (Fernando et al., 2013)	82.4 ± 1.1	74.9 ± 1.7	76.3 ± 2.3	81.2 ± 1.3	82.7 ± 0.8	86.9 ± 3.3	78.5
TCA(Pan et al., 2011)	80.2 ± 0.9	69.4 ± 1.2	72.8 ± 1.0	81.2 ± 1.1	83.1 ± 0.8	89.9 ± 1.9	78.1
KMM(Huang et al., 2006)	80.8 ± 1.6	75.5 ± 2.5	81.5 ± 2.8	83.5 ± 1.6	77.5 ± 1.4	89.5 ± 1.7	79.5
DME-MMD	82.7 ± 0.9	72.6 ± 2.5	78.2 ± 1.4	81.8 ± 1.3	76.8 ± 1.5	88.2 ± 1.5	79
DME-H	82.2 ± 1.4	74.1 ± 1.2	79 ± 1.1	81.8 ± 0.9	78 ± 1.1	87.8 ± 2.6	78.9
NL-DME-MMD	81.9 ± 0.7	73.7 ± 1.3	78.8 ± 2.2	82.8 ± 1.3	77.2 ± 1.0	94.1 ± 1.7	80.1
NL-DME-H	82.6 ± 1.1	74.8 ± 1.5	79.6 ± 1.9	81.3 ± 3.4	77.5 ± 1.0	90.1 ± 2.4	79.3

Table 16: Recognition accuracies on the remaining 6 pairs of source/target domains using the evaluation protocol of (Saenko et al., 2010) and using Decaf8 features with Linear SVM. C**: Caltech,** A**: Amazon,** W**: Webcam,** D**: DSLR**.

subspace-based methods. As before, NL-DME-H yields the best average accuracy of all methods. Note that all our algorithms are quite robust to the choice of subspace dimension.

7. 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. In particular, we have studied two different metrics to compare the distributions and have introduced linear and nonlinear techniques to map the data to latent spaces. Our experiments have evidenced the importance of exploiting distribution invariance for domain adaptation by revealing that our algorithms yield state-of-the-art results on several problems, with best



Samples from the source domain (Webcam), Mug Class

Samples correctly classified by DME-H, Misclassified by DME-MMD

Samples correctly classified by NL-DME-H, Misclassified by DME-MMD/ DME-H/ NL-DME-MMD

Samples correctly classified by DME-MMD, Misclassified by NO ADAPT-SVM

Samples correctly classified by NL-DME-MMD, Misclassified by DME-MMD/ DME-H

Samples misclassified by all methods

Figure 4: Misclassification examples for our algorithms when using Decaf7 features: Webcam as source, Amazon as target.

Method	$A \rightarrow C$	$C \rightarrow A$	$C \rightarrow D$	$C \rightarrow W$	$D \rightarrow C$	$W \rightarrow D$	Avg.
NO ADAPT-SVM	85.4 ± 0.8	90.9 ± 0.9	86.1 ± 3.2	77.5 ± 4.3	80.4 ± 1.0	76.1 ± 0.6	82.7
SVMA (Duan et al., 2012)	84.5 ± 1.3	91.8 ± 0.70	83.5 ± 2.3	78.5 ± 4.0	80.8 ± 0.9	77.4 ± 0.6	82.8
DAM (Duan et al., 2012)	85.5 ± 1.2	92.2 ± 0.6	83.5 ± 2.7	81.1 ± 3.9	83.1 ± 1.2	78.8 ± 1.8	84.0
GFK (Gong et al., 2012)	86.2 ± 0.8	91.2 ± 0.8	89.5 ± 2.3	84.6 ± 1.8	81.6 ± 1.1	80.6 ± 1.1	85.6
SA (Fernando et al., 2013)	86.3 ± 0.7	91.4 ± 0.6	88.4 ± 4.3	85.8 ± 3.1	82.9 ± 1.2	82.7 ± 0.8	86.3
TCA(Pan et al., 2011)	86.5 ± 0.8	91.7 ± 0.5	86.3 ± 3.8	84.6 ± 2.9	84 ± 0.9	83.1 ± 0.8	86.0
KMM(Huang et al., 2006)	85.5 ± 0.7	91.4 ± 0.7	86.3 ± 2.7	78.2 ± 4.9	80.3 ± 0.8	77.5 ± 1.4	83.2
DDC(Tzeng et al., 2014)	84.3 ± 0.5	91.3 ± 0.3	89.1 ± 0.3	85.5 ± 0.3	80.5 ± 0.2	76.9 ± 0.4	84.6
DME-MMD	85.6 ± 0.8	91.1 ± 0.8	88.2 ± 2.3	82.5 ± 2.9	83.1 ± 2.6	76.8 ± 1.5	84.6
DME-H	86.1 ± 1.1	92.2 ± 0.5	88.6 ± 2.9	82 ± 3.4	82.4 ± 2.4	78 ± 1.1	84.8
NL-DME-MMD	86.2 ± 0.9	91.7 ± 0.53	87.5 ± 2.8	82.5 ± 2.1	84.8 ± 0.9	77.2 ± 1.0	84.9
NL-DME-H	85.3 ± 1.0	91.3 ± 0.8	88.4 ± 2.9	83 ± 0.4	85.8 ± 1.0	77.5 ± 1.0	85.2

Table 17: Recognition accuracies on 6 pairs of source/target domains using the evaluation protocol of (Saenko et al., 2010) and using Decaf7 features (Comparison with Domain Deep Confusion). C**: Caltech,** A**: Amazon,** W**: Webcam,** D**: DSLR**.

performance achieved by NL-DME-H. A current limitation of our approach is the non-convexity of the optimization problems. 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. Finally, we also plan to study the performance of other metrics, such as the KL-divergence and the Bhattacharyya distance, to compare the source and target distributions.

References

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

Method	$A \to D$	$D \to W$	$W \to D$
NO ADAPT-SVM	50.5 ± 2.5	87.1 ± 1.9	93.7 ± 0.8
SVMA (Duan et al., 2012)	56.7 ± 2.6	88.3 ± 1.5	95.2 ± 0.8
DAM (Duan et al., 2012)	54.9 ± 3.2	86.9 ± 1.4	84.2 ± 1.4
GFK (Gong et al., 2012)	51.2 ± 3.1	85.2 ± 2.3	93.2 ± 1.2
TCA (Pan et al., 2011)	50.1 ± 3.1	82.7 ± 1.9	91.6 ± 2.4
KMM (Huang et al., 2006)	43.7 ± 2.8	83.9 ± 2.3	93.3 ± 1.2
SA (Fernando et al., 2013)	51.8 ± 3.1	88.4 ± 1.8	96.5 ± 0.8
RG(Ganin and Lempitsky, 2015)	67.3 ± 1.7	94.0 ± 0.8	93.7 ± 1.0
DDC(Tzeng et al., 2014)	59.4 ± 0.8	92.5 ± 0.3	91.7 ± 0.8
DME-MMD	53.2 ± 2.3	86.3 ± 1.8	93.7 ± 1.5
DME-H	51.6 ± 2.0	87.4 ± 1.3	92.9 ± 1.3
NL-DME-MMD	53.5 ± 3.1	87.8 ± 1.2	95.6 ± 2.4
NL-DME-H	52.2 ± 2.4	88.3 ± 1.9	94.7 ± 2.1

Table 18: Recognition accuracies on the 3 pairs of source/target domains on Office31 data set using the evaluation protocol of (Saenko et al., 2010) and using Decaf7 features with Linear SVM. A**: Amazon,** W**: Webcam,** D**: DSLR**.

Setting	**Source Domain**	**Target Domain**
comp vs. rec	comp.windows.x & rec.sport.hockey	comp.sys.ibm.pc.hardware & rec.motorcycles
comp vs. sci	comp.windows.x & sci.crypt	comp.sys.ibm.pc.hardware & sci.med
comp vs. talk	comp.windows.x & talk.politics.mideast	comp.sys.ibm.pc.hardware & talk.politics.guns

Table 19: Experimental setting for the 20 Newsgroups data set.

Method	comp vs. rec	comp vs. sci	comp vs. talk	Avg.
NO ADAPT-SVM	79.9 ± 2.4	70.2 ± 1.6	80.6 ± 1.1	76.9
DTMKL (Duan et al., 2012)	87.8 ± 2.1	74.5 ± 1.1	$\mathbf{92.4 \pm 0.9}$	84.9
TCA (Pan et al., 2011)	85.5 ± 1.7	75.9 ± 1.0	91.9 ± 0.7	84.4
SA (Fernando et al., 2013)	80.4 ± 2.8	71.2 ± 2.1	90.1 ± 1.1	80.6
DME-MMD	87.2 ± 1.7	77 ± 1.0	91.7 ± 0.7	85.3
DME-H	87.1 ± 1.4	74.2 ± 1.9	91.6 ± 0.7	84.3
NL-DME-MMD	88.3 ± 1.8	77.3 ± 1.0	89.8 ± 0.9	85.2
NL-DME-H	$\mathbf{88.7} \pm 1.6$	$\mathbf{79.0} \pm \mathbf{2.7}$	89.3 ± 0.8	$\mathbf{85.7}$

Table 20: Recognition accuracies for the 3 source/target pairs on the 20 Newsgroups data set.

M. Baktashmotlagh, M. Harandi, B. Lovell, and M. Salzmann. Unsupervised domain adaptation by domain invariant projection. In *Int'l Conf. on Computer Vision (ICCV)*, pages 769–776. IEEE, 2013.

M. Baktashmotlagh, M. Harandi, B. Lovell, and M. Salzmann. Domain adaptation on the statistical manifold. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2481–2488. IEEE, 2014.

H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *Euro. Conf. on Computer Vision (ECCV)*, pages 404–417. Springer, 2006.

A. Bergamo and L. Torresani. Exploiting weakly-labeled web images to improve object classification: a domain adaptation approach. In *Advances in Neural Information Processing Systems (NIPS)*, pages 181–189, 2010.

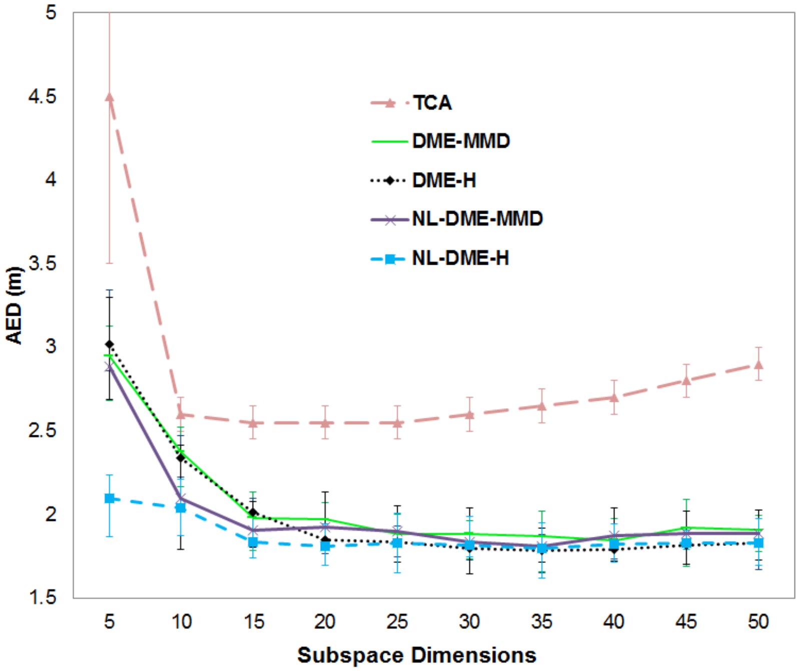


Figure 5: Comparison of the our algorithms with TCA on the task of WiFi localization. Note that NL-DME-H yields the best results, and that, in general, our algorithms are robust to the choice of subspace dimension.

J. Blitzer, D. Foster, and S. Kakade. Domain adaptation with coupled subspaces. *Journal of Machine Learning Research (JMLR)*, pages 173–181, 2011.

K. Borgwardt, A. Gretton, M. Rasch, H. Kriegel, B. Schoelkopf, and A. Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Journal of Bioinformatics (BIO)*, 22(14):49–57, 2006.

L. Bruzzone and M. Marconcini. Domain adaptation problems: A dasvm classification technique and a circular validation strategy. *The IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(5):770–787, 2010.

S. Canu and A. Smola. Kernel methods and the exponential family. *Neurocomputing*, 69(7):714–720, 2006.

K. Carter. *Dimensionality reduction on statistical manifolds*. ProQuest, 2009.

M. Chen, K. Weinberger, and J. Blitzer. Co-training for domain adaptation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2456–2464, 2011.

H. Daumé III and D. Marcu. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research (JAIR)*, 26:101–126, 2006.

H. Daumé III, A. Kumar, and A. Saha. Co-regularization based semi-supervised domain adaptation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 478–486, 2010.

J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. IEEE, 2009.

J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *Int'l Conf. on Machine Learning (ICML)*, pages 647–655, 2014.

L. Duan, I. Tsang, D. Xu, and T. Chua. Domain adaptation from multiple sources via auxiliary classifiers. In *Int'l Conf. on Machine Learning (ICML)*, pages 289–296. ACM, 2009a.

L. Duan, I. Tsang, D. Xu, and S. Maybank. Domain transfer svm for video concept detection. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1375–1381. IEEE, 2009b.

L. Duan, I. W Tsang, and D. Xu. Domain transfer multiple kernel learning. *The IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(3):465–479, 2012.

A. Edelman, T. Arias, and S. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Mathematical Analysis (SIAM)*, 20(2):303–353, 1998.

B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *Int'l Conf. on Computer Vision (ICCV)*, pages 2960–2967, 2013.

Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *Int'l Conf. on Machine Learning (ICML)*, pages 1180–1189, 2015.

B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2066–2073. IEEE, 2012.

B. Gong, K. Grauman, and F. Sha. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *Int'l Conf. on Machine Learning (ICML)*, pages 222–230, 2013.

R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *Int'l Conf. on Computer Vision (ICCV)*, pages 999–1006. IEEE, 2011.

A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Schölkopf. Covariate shift by kernel mean matching. *Journal of Royal. Statistical Society (JRSS)*, 3(4):5, 2009.

A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *Journal of Machine Learning Research (JMLR)*, 13(1):723–773, 2012a.

A. Gretton, D. Sejdinovic, H. Strathmann, S. Balakrishnan, M. Pontil, K. Fukumizu, and B. Sriperumbudur. Optimal kernel choice for large-scale two-sample tests. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1205–1213, 2012b.

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

J. Hamm and D. Lee. Grassmann discriminant analysis: a unifying view on subspace-based learning. In *Int'l Conf. on Machine Learning (ICML)*, pages 376–383. ACM, 2008.

M. Harandi, C. Sanderson, S. Shirazi, and B. Lovell. Kernel analysis on grassmann manifolds for action recognition. *Pattern Recognition Letters (PRL)*, 34(15):1906–1915, 2013.

R. Hartley, J. Trumpf, Y. Dai, and H. Li. Rotation averaging. *Int'l Journal of Computer Vision (IJCV)*, 103(3):267–305, 2013.

J. Hoffman, B. Kulis, T. Darrell, and K. Saenko. Discovering latent domains for multisource domain adaptation. In *Euro. Conf. on Computer Vision (ECCV)*, pages 702–715. Springer, 2012.

J. Huang, A. J Smola, A. Gretton, K. Borgwardt, and B. Scholkopf. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems (NIPS)*, pages 601–608, 2006.

V. Jain and E. Learned-Miller. Online domain adaptation of a pre-trained cascade of classifiers. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 577–584. IEEE, 2011.

L. Jie, T. Tommasi, and B. Caputo. Multiclass transfer learning from unconstrained priors. In *Int'l Conf. on Computer Vision (ICCV)*, pages 1863–1870. IEEE, 2011.

Robert E Kass and Paul W Vos. *Geometrical foundations of asymptotic inference*. Wiley. com, 2011.

B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1785–1792. IEEE, 2011.

I. Kuzborskij and F. Orabona. Stability and hypothesis transfer learning. In *Int'l Conf. on Machine Learning (ICML)*, pages 942–950, 2013.

K. Muandet, D. Balduzzi, and B. Schölkopf. Domain generalization via invariant feature representation. In *Int'l Conf. on Machine Learning (ICML)*, pages 10–18, 2013.

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

A. Peter and A. Rangarajan. Shape analysis using the fisher-rao riemannian metric: Unifying shape representation and deformation. In *Biomedical Imaging (ISBI)*, pages 1164–1167. IEEE, 2006.

U. Rückert and M. Kloft. Transfer learning with adaptive regularizers. In *Euro. Conf. on Machine Learning (ECML)*, pages 65–80. 2011.

A. Ruszczynski. *Nonlinear optimization*. Princeton University press, 2006.

K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *Euro. Conf. on Computer Vision (ECCV)*, pages 213–226. Springer, 2010.

B. Schölkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT press, 2002.

A. Srivastava and X. Liu. Tools for application-driven linear dimension reduction. *Neurocomputing*, 67:136–160, 2005.

A. Srivastava, I. Jermyn, and S. Joshi. Riemannian analysis of probability density functions with applications in vision. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2007.

I. Steinwart. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research (JMLR)*, 2:67–93, 2002.

G. Terrell. The maximal smoothing principle in density estimation. *J. of the American Statistical Association*, 85(410): 470–477, 1990.

T. Tommasi and B. Caputo. Frustratingly easy nbnn domain adaptation. In *Int'l Conf. on Computer Vision (ICCV)*, pages 897–904, 2013.

T. Tommasi and T. Tuytelaars. A testbed for cross-dataset analysis. In *Euro. Conf. on Computer Vision (ECCV Workshops)*, pages 18–31, 2014.

T. Tommasi, F. Orabona, and B. Caputo. Safety in numbers: Learning categories from few examples with multi model knowledge transfer. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3081–3088. IEEE, 2010.

T. Tommasi, F. Orabona, and B. Caputo. Learning categories from few examples with multi model knowledge transfer. *The IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(5):928–941, 2014.

O. Tuzel, F. Porikli, and P. Meer. Pedestrian detection via classification on riemannian manifolds. *The IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(10):1713–1727, 2008.

E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Deep domain confusion: Maximizing for domain invariance. In *CoRR, abs/1412.3474*, 2014.

D. Xing, W. Dai, G. Xue, and Y. Yu. Bridged refinement for transfer learning. In *Euro. Conf. on Machine Learning (ECML)*, pages 324–335. Springer, 2007.

Q. Yang, J. Pan, and V. Zheng. Estimating location using wi-fi. *IEEE Intelligent Systems*, 23(1):8–13, 2008.