2013 IEEE International Conference on Computer Vision

Unsupervised Visual Domain Adaptation Using Subspace Alignment

Basura Fernando[1], Amaury Habrard[2], Marc Sebban[2], and Tinne Tuytelaars[1]

[1]KU Leuven, ESAT-PSI, iMinds, Belgium
[2]Laboratoire Hubert Curien UMR 5516, 18 rue Benoit Lauras, 42000 St-Etienne, France

Abstract

In this paper, we introduce a new domain adaptation (DA) algorithm where the source and target domains are represented by subspaces described by eigenvectors. In this context, our method seeks a domain adaptation solution by learning a mapping function which aligns the source subspace with the target one. We show that the solution of the corresponding optimization problem can be obtained in a simple closed form, leading to an extremely fast algorithm. We use a theoretical result to tune the unique hyperparameter corresponding to the size of the subspaces. We run our method on various datasets and show that, despite its intrinsic simplicity, it outperforms state of the art DA methods.

1. Introduction

In classification, it is typically assumed that the labeled training data comes from the same distribution as that of the test data. However, many real world applications, especially in computer vision, challenge this assumption (see, e.g., the study on dataset bias in [15]). In this context, the learner must take special care during the learning process to infer models that adapt well to the test data they are deployed on. For example, images collected from a web camera are different from those taken with a DSLR camera. A classifier that would be trained on the former would likely fail to classify the latter correctly if applied without adaptation.

We refer to these different but related marginal distributions as domains. In order to build robust classifiers, it is necessary to take into account the shift between these two distributions. This issue is known as *domain adaptation* (DA). DA typically aims at making use of information coming from both source and target domains during the learning process to adapt automatically. We usually differentiate two different scenarios: (1) the *unsupervised* setting where the training data consists of labeled source data and unlabeled target examples (see [11] for a survey); and (2) the *semi-supervised* case where a large number of labels is available for the source domain and only a few labels are provided for the target domain. In this paper, we focus on the most difficult, unsupervised scenario.

As illustrated by recent results [7, 8], *subspace based domain adaptation* seems a promising approach to tackle unsupervised visual DA problems. In [8], Gopalan et al. generate intermediate representations in the form of subspaces along the geodesic path connecting the source subspace and the target subspace on the Grassmann manifold. Then, the source data are projected onto these subspaces and a classifier is learned. In [7], Gong et al. propose a geodesic flow kernel which aims to model incremental changes between the source and target domains. In both papers, a set of intermediate subspaces is used to model the shift between the two distributions.

In this paper, we also make use of subspaces (composed of d eigenvectors induced by a PCA), one for each domain. However, following the theoretical recommendations of [1], we rather suggest to directly reduce the discrepancy between the two domains by moving closer the source and target subspaces. This is achieved by optimizing a mapping function that transforms the source subspace into the target one. From this simple idea, we design a new DA approach based on *subspace alignment*. The advantage of our method is two-fold: (1) by adapting the bases of the subspaces, our approach is *global*. This allows us to induce robust classifiers not subject to local perturbations; and (2) by aligning the source and target subspaces, our method is intrinsically regularized: we do not need to tune regularization parameters in the objective as imposed by a lot of optimization-based DA methods.

Our subspace alignment is achieved by optimizing a mapping function which takes the form of a transformation matrix M. We show that the optimal solution corresponds in fact to the covariance matrix between the source and target eigenvectors. From this transformation matrix, we derive a similarity function $Sim(\mathbf{y_S}, \mathbf{y_T})$ to compare a source data $\mathbf{y_S}$ with a target example $\mathbf{y_T}$. Thanks to a consistency theorem, we prove that $Sim(\mathbf{y_S}, \mathbf{y_T})$, which captures the

1550-5499/13 $31.00 © 2013 IEEE
DOI 10.1109/ICCV.2013.368

idiosyncrasies of the training data, converges uniformly to its true value. We show that we can make use of this theoretical result to tune the hyperparameter d, that tends to make our method parameter-free. The similarity function $Sim(\mathbf{y_S}, \mathbf{y_T})$ can be used directly in a nearest neighbour classifier. Alternatively, we can also learn a global classifier such as support vector machines (SVM) on the source data after mapping them onto the target subspace.

As suggested by Ben-David et al. [1], a reduction of the divergence between the two domains is required to adapt well. In other words, the ability of a DA algorithm to actually reduce that discrepancy is a good indication of its performance. A usual way to estimate the divergence consists in learning a linear classifier h to discriminate between source and target instances, respectively pseudo-labeled with 0 and 1. In this context, the higher the error of h, the smaller the divergence. While such a strategy gives us some insight about the ability for a *global* learning algorithm (e.g. SVM) to be efficient on both domains, it does not seem to be suited to deal with *local* classifiers, such as the k-nearest neighbors. To overcome this limitation, we introduce a new empirical divergence specifically dedicated to local classifiers. We show through our experimental results that our DA method allows us to drastically reduce both empirical divergences.

The rest of the paper is organized as follows. We present the related work in section 2. Section 3 is devoted to the presentation of our DA method and the consistency theorem on the similarity measure deduced from the learned mapping function. In section 4, a comparative study is performed on various datasets. We conclude in section 5.

2. Related work

DA has been widely studied in the literature and is of great importance in many areas such as natural language processing [4] or computer vision [15]. In this paper, we focus on the unsupervised domain adaptation setting that is well suited to vision problems since it does not require any labeling information from the target domain. This setting makes the problem very challenging and an important issue is to find out a relationship between the two domains. A common approach is to assume the existence of a domain invariant feature space and the objective of a large range of DA work is to approximate this space.

A classical strategy related to our work consists of learning a new domain-invariant feature representation by looking for a new projection space. PCA based DA methods have then been naturally investigated [6, 12, 13] in order to find a common latent space where the difference between the marginal distributions of the two domains is minimized with respect to the Maximum Mean Discrepancy (MMD) divergence. Other strategies have been explored as well such as using metric learning approaches [10, 14] or canonical correlation analysis methods over different views of the data to find a coupled source-target subspace [3] where one assumes the existence of a performing linear classifier on the two domains.

In the structural correspondence learning method [4], Blitzer et al. propose to create a new feature space by identifying correspondences among features from different domains by modeling their correlations with pivot features. Then, they concatenate source and target data using this feature representation and apply PCA to find a relevant common projection. In [5], Chang transforms the source data into an intermediate representation such that each transformed source sample can be linearly reconstructed by the target samples. This is however a local approach that may fail to capture the global structure information of the source domain. Moreover it is sensitive to noise and outliers of the source domain that have no correspondence in the target one.

Our method is also related to manifold alignment [16, 17, 18] whose main objective is to align two datasets from two different manifolds such that they can be projected to a common subspace. Most of these methods [17, 18] need correspondences from the manifolds and all of them exploit the local statistical structure of the data.

Recently, subspace based DA has demonstrated good performance in visual DA [7, 8]. These methods share the same principle: first they compute a domain specific d-dimensional subspace for the source data and another one for the target data, independently created by PCA. Then, they project source and target data into intermediate subspaces along the shortest geodesic path connecting the two d-dimensional subspaces on the Grassmann manifold. They actually model the distribution shift by looking for the best intermediate subspaces. These approaches are the closest to ours but, as mentioned in the introduction, it is more appropriate to align the two subspaces directly, instead of computing a large number of intermediate subspaces which can potentially be a costly tuning procedure. The effectiveness of our idea is supported by our experimental results.

As a summary, our approach has the following differences with existing methods:

We exploit the *global* covariance statistical structure of the two domains during the adaptation process in contrast to the manifold alignment methods that use local statistical structure of the data [16, 17, 18]. We project the source data onto the source subspace and the target data onto the target subspace in contrast to methods that project source data to the target subspace or target data to the source subspace such as [3]. Moreover, we do not project data to a large number of subspaces as in [7, 8]. Our method is totally unsupervised and does not require any target label information like constraints on cross-domain data [10, 14] or correspondences from across datasets [17, 18]. We do not apply PCA on cross-domain data like in [6, 12, 13] as these approaches exploit only shared features in both domains. In contrast, we make use of the correlated features in both domains. Some of these features can be specific to one domain yet correlated to some other features in the other one allowing us to use both shared and domain specific features. As far as we know, this is the first attempt to use a subspace alignment method in the context of domain adaptation.

3. DA using unsupervised subspace alignment

In this section, we introduce our new subspace based DA method. We assume that we have a set S of labeled data (resp. a set T of unlabeled data) both lying in a given D-dimensional space and drawn i.i.d. according to a fixed but unknown source (resp. target) distribution $\mathcal{D}_S$ (resp. $\mathcal{D}_T$). We denote the transpose operation by $'$.

In section 3.1, we explain how to generate the source and target subspaces of size d. Then, we present our DA method in section 3.2 which consists in learning a transformation matrix M that maps the source subspace to the target one. From M, we design a similarity function for which we derive a consistency theorem in section 3.3. This upper bound gives us some insight about how to tune the parameter d.

3.1. Subspace generation

Even though both the source and target data lie in the same D-dimensional space, they have been drawn according to different marginal distributions. Consequently, rather than working on the original data themselves, we suggest to handle more robust representations of the source and target domains and to learn the shift between these two domains. First, we transform every source and target data in the form of a D-dimensional z-normalized vector (i.e. of zero mean and unit standard deviation). Then, using PCA, we select for each domain d eigenvectors corresponding to the d largest eigenvalues. These eigenvectors are used as bases of the source and target subspaces, respectively denoted by X_S and X_T ($X_S, X_T \in \mathbb{R}^{D\times d}$). Note that X'_S and X'_T are orthonormal (thus, $X'_S X_S = I_d$ and $X'_T X_T = I_d$ where I_d is the identity matrix of size d). In the following, X_S and X_T are used to learn the shift between the two domains.

3.2. Domain adaptation with subspace alignment

As presented in section 2, two main strategies are used in subspace based DA methods. The first one consists in projecting both source and target data to a common shared subspace. However, since this only exploits shared features in both domains, it is not always optimal. The second one aims to build a (potentially large) set of intermediate representations. Beyond the fact that such a strategy can be costly, projecting the data to an intermediate common shared subspace may lead to information loss in both source and target domains.

In our method, we suggest to project each source ($\mathbf{y_S}$) and target ($\mathbf{y_T}$) data (where $\mathbf{y_S}, \mathbf{y_T} \in \mathbb{R}^{1\times D}$) to its respective subspace X_S and X_T by the operations $\mathbf{y_S} X_S$ and $\mathbf{y_T} X_T$, respectively. Then, we learn a linear transformation function that align the source subspace coordinate system to the target one. This step allows us to directly compare source and target samples in their respective subspaces without unnecessary data projections. To achieve this task, we use a *subspace alignment* approach. We align basis vectors by using a transformation matrix M from X_S to X_T. M is learned by minimizing the following Bregman matrix divergence:

$$F(M) = ||X_S M - X_T||_F^2 \tag{1}$$

$$M^* = argmin_M(F(M)) \tag{2}$$

where $||.||_F^2$ is the Frobenius norm. Since X_S and X_T are generated from the first d eigenvectors, it turns out that they tend to be intrinsically regularized. Therefore, we do not add a regularization term in the equation 1. It is thus possible to obtain a simple solution of equation 2 in closed form. Because the Frobenius norm is invariant to orthonormal operations, we can re-write equation 1 as follows:

$$F(M) = ||X'_S X_S M - X'_S X_T||_F^2 = ||M - X'_S X_T||_F^2. \tag{3}$$

From this result, we can conclude that the optimal M^* is obtained as $M^* = X'_S X_T$. This implies that the new coordinate system is equivalent to $X_a = X_S X'_S X_T$. We call X_a the *target aligned source coordinate system.* It is worth noting that if the source and target domains are the same, then $X_S = X_T$ and M^* is the identity matrix.

Matrix M transforms the source subspace coordinate system into the target subspace coordinate system by aligning the source basis vectors with the target ones. If a source basis vector is orthogonal to all target basis vectors, it is ignored. On the other hand, a high weight is given to a source basis vector that is well aligned with the target basis vectors.

In order to compare a source data $\mathbf{y_S}$ with a target data $\mathbf{y_T}$, one needs a similarity function $Sim(\mathbf{y_S}, \mathbf{y_T})$. Projecting $\mathbf{y_S}$ and $\mathbf{y_T}$ in their respective subspace X_S and X_T and applying the optimal transformation matrix M^*, we can define $Sim(\mathbf{y_S}, \mathbf{y_T})$ as follows:

$$\begin{aligned} Sim(\mathbf{y_S}, \mathbf{y_T}) &= (\mathbf{y_S} X_S M^*)(\mathbf{y_T} X_T)' = \mathbf{y_S} X_S M^* X'_T \mathbf{y_T}' \\ &= \mathbf{y_S} A \mathbf{y_T}', \end{aligned} \tag{4}$$

where $A = X_S X'_S X_T X'_T$. Note that Eq. 4 looks like a generalized dot product (even though A is not necessarily

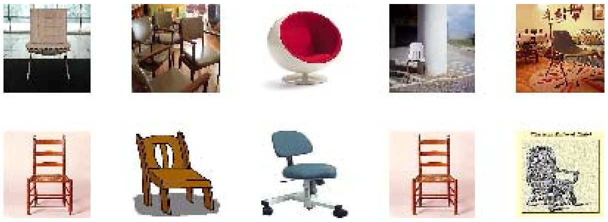

Figure 1. Classifying ImageNet images using Caltech-256 images as the source domain. In the first row, we show an ImageNet query image. In the second row, the nearest neighbour image selected by our method is shown.

positive semidefinite) where A encodes the relative contributions of the different components of the vectors in their original space.

We use $Sim(\mathbf{y_S}, \mathbf{y_T})$ directly to perform a k-nearest neighbor classification task. On the other hand, since $Sim(\mathbf{y_S}, \mathbf{y_T})$ is not PSD we can not make use of it to learn a SVM directly. As we will see in the experimental section, an alternative solution will consist in (i) projecting the source data via X_a into the target aligned source subspace and the target data into the target subspace (using X_T), (ii) learn a SVM from this d-dimensional space. The pseudo-code of our algorithm is presented in Algorithm 1.

```
Data: Source data $S$, Target data $T$, Source labels $L_S$,
      Subspace dimension $d$
Result: Predicted target labels $L_T$
$X_S \leftarrow PCA(S, d)$ ;
$X_T \leftarrow PCA(T, d)$ ;
$X_a \leftarrow X_S X_S' X_T$ ;
$S_a = S X_a$ ;
$T_T = T X_T$ ;
$L_T \leftarrow Classifier(S_a, T_T, L_S)$ ;
```

Algorithm 1: Subspace alignment DA algorithm

3.3. Consistency theorem on $Sim(\mathbf{y_S}, \mathbf{y_T})$

The unique hyperparameter of our algorithm is the number d of eigenvectors. In this section, inspired by concentration inequalities on eigenvectors [19], we derive an upper bound on the similarity function $Sim(\mathbf{y_S}, \mathbf{y_T})$. Then, we show that we can make use of this theoretical result to efficiently tune d.

Let $\tilde{D}_n$ be the covariance matrix of a sample D of size n drawn i.i.d. from a given distribution and $\tilde{D}$ its expected value over that distribution.

Theorem 1. *We start by using a theorem from [19]. Let B be s.t. for any vector $\mathbf{x}$, $\|\mathbf{x}\| \leq B$, let $X^d_{\tilde{D}}$ and $X^d_{\tilde{D}_n}$ be the orthogonal projectors of the subspaces spanned by the first d eigenvectors of $\tilde{D}$ and $\tilde{D}_n$. Let $\lambda_1 > \lambda_2 > ... > \lambda_d > \lambda_{d+1} \geq 0$ be the first $d+1$ eigenvalues of $\tilde{D}$, then for any $n \geq \left(\frac{4B}{(\lambda_d - \lambda_{d+1})}\left(1 + \sqrt{\frac{\ln(1/\delta)}{2}}\right)\right)^2$ with probability at least $1 - \delta$ we have:*

$$\|X^d_{\tilde{D}} - X^d_{\tilde{D}_n}\| \leq \frac{4B}{\sqrt{n}(\lambda_d - \lambda_{d+1})}\left(1 + \sqrt{\frac{\ln(1/\delta)}{2}}\right).$$

From the previous theorem, we can derive the following lemma for the deviation between $X^d_{\tilde{D}} {X^d_{\tilde{D}}}'$ and $X^d_{\tilde{D}_n} {X^d_{\tilde{D}_n}}'$. For the sake of simplification, we will use in the following the same notation D (resp. D_n) for defining either the sample D (resp. D_n) or its covariance matrix $\tilde{D}$ (resp. $\tilde{D}_n$).

Lemma 1. *Let B s.t. for any $\mathbf{x}$, $\|\mathbf{x}\| \leq B$, let X^d_D and $X^d_{D_n}$ the orthogonal projectors of the subspaces spanned by the first d eigenvectors of D and D_n. Let $\lambda_1 > \lambda_2 > ... > \lambda_d > \lambda_{d+1} \geq 0$ be the first $d+1$ eigenvalues of D, then for any $n \geq \left(\frac{4B}{(\lambda_d - \lambda_{d+1})}\left(1 + \sqrt{\frac{\ln(1/\delta)}{2}}\right)\right)^2$ with probability at least $1 - \delta$ we have:*

$$\|X^d_D {X^d_D}' - X^d_{D_n} {X^d_{D_n}}'\| \leq \frac{8\sqrt{d}}{\sqrt{n}} \frac{B}{(\lambda_d - \lambda_{d+1})}\left(1 + \sqrt{\frac{\ln(1/\delta)}{2}}\right)$$

Proof.

$$\begin{aligned}
&\|X^d_D {X^d_D}' - X^d_{D_n} {X^d_{D_n}}'\| \\
=&\|X^d_D {X^d_D}' - X^d_D {X^d_{D_n}}' + X^d_D {X^d_{D_n}}' - X^d_{D_n} {X^d_{D_n}}'\| \\
\leq&\|X^d_D\|\|{X^d_D}' - {X^d_{D_n}}'\| + \|X^d_D - X^d_{D_n}\|\|{X^d_{D_n}}'\| \\
\leq&\frac{2\sqrt{d}}{\sqrt{n}} \frac{4B}{(\lambda_d - \lambda_{d+1})}\left(1 + \sqrt{\frac{\ln(1/\delta)}{2}}\right)
\end{aligned}$$

The last inequality is obtained by the fact that the eigenvectors are normalized and thus $\|X_D\| \leq \sqrt{d}$ and application of Theorem 1 twice. □

We now give a theorem for the projector of our DA method.

Theorem 2. *Let $X^d_{S_n}$ (resp. $X^d_{T_n}$) be the d-dimensional projection operator built from the source (resp. target) sample of size n_S (resp. n_T) and X^d_S (resp. X^d_T) its expected value with the associated first $d+1$ eigenvalues $\lambda^S_1 > ... > \lambda^S_d > \lambda^S_{d+1}$ (resp. $\lambda^T_1 > ... > \lambda^T_d > \lambda^T_{d+1}$), then we have with probability at least $1 - \delta$*

$$\|X^d_S M {X^d_T}' - X^d_{S_n} M_n {X^d_{T_n}}'\| \leq 8d^{3/2}B\left(1 + \sqrt{\frac{\ln(2/\delta)}{2}}\right)$$
$$\times\left(\frac{1}{\sqrt{n_S}(\lambda^S_d - \lambda^S_{d+1})} + \frac{1}{\sqrt{n_T}(\lambda^T_d - \lambda^T_{d+1})}\right)$$

where M_n is the solution of the optimization problem of Eq 2 using source and target samples of sizes n_S and n_T respectively, and M is its expected value.

Proof.

$$\begin{aligned}
&\|X_S^d M X_T^{d\prime} - X_{S_n}^d M_n X_{T_n}^{d}{}'\| = \\
&\|X_S^d X_S^{d\prime} X_T^d X_T^{d\prime} - X_{S_n}^d X_{S_n}^{d}{}' X_{T_n}^d X_{T_n}^{d}{}'\| \\
=&\|X_S^d X_S^{d\prime} X_T^d X_T^{d\prime} - X_S^d X_S^{d\prime} X_{T_n}^d X_{T_n}^{d}{}' + \\
&X_S^d X_S^{d\prime} X_{T_n}^d X_{T_n}^{d}{}' - X_{S_n}^d X_{S_n}^{d}{}' X_{T_n}^d X_{T_n}^{d}{}'\| \\
\leq&\|X_S^d X_S^{d\prime} X_T^d X_T^{d\prime} - X_S^d X_S^{d\prime} X_{T_n}^d X_{T_n}^{d}{}'\| + \\
&\|X_S^d X_S^{d\prime} X_{T_n}^d X_{T_n}^{d}{}' - X_{S_n}^d X_{S_n}^{d}{}' X_{T_n}^d X_{T_n}^{d}{}'\| \\
\leq&\|X_S^d\|\|X_S^{d\prime}\|\|X_T^d X_T^{d\prime} - X_{T_n}^d X_{T_n}^{d}{}'\| + \\
&\|X_S^d X_S^{d\prime} - X_{S_n}^d X_{S_n}^{d}{}'\|\|X_{T_n}^d\|\|X_{T_n}^{d}{}'\| \\
\leq&8d^{3/2}B\left(1+\sqrt{\frac{\ln(2/\delta)}{2}}\right)\times \\
&\left(\frac{1}{\sqrt{n_S}(\lambda_d^S-\lambda_{d+1}^S)}+\frac{1}{\sqrt{n_T}(\lambda_d^T-\lambda_{d+1}^T)}\right).
\end{aligned}$$

□

The first equality is obtained by replacing M and M_n by their corresponding optimal solutions $X_S^d X_T^{d\prime}$ and $X_{S_n}^d X_{T_n}^{d}{}'$ from Eq 3. The last inequality is obtained by applying twice Lemma 1 and bounding the projection operators.

From Theorem 2, we can deduce a bound on the deviation between two successive eigenvalues. We can make use of this bound as a cutting rule for automatically determining the size of the subspaces. Let $n_{min} = \min(n_S, n_T)$ and $(\lambda_d^{min} - \lambda_{d+1}^{min}) = \min((\lambda_d^T - \lambda_{d+1}^T), (\lambda_d^S - \lambda_{d+1}^S))$ and let $\gamma > 0$ be a given allowed deviation such that:

$$\gamma \geq \left(1+\sqrt{\frac{\ln 2/\delta}{2}}\right)\left(\frac{16d^{3/2}B}{\sqrt{n_{min}}(\lambda_d^{min}-\lambda_{d+1}^{min})}\right).$$

Given a confidence $\delta > 0$ and a fixed deviation $\gamma > 0$, we can select the maximum dimension d_{max} such that:

$$(\lambda_{d_{max}}^{min} - \lambda_{d_{max}+1}^{min}) \geq \left(1+\sqrt{\frac{\ln 2/\delta}{2}}\right)\left(\frac{16d^{3/2}B}{\gamma\sqrt{n_{min}}}\right). \tag{5}$$

For each $d \in \{d|1\ldots d_{max}\}$, we then have the guarantee that $\|X_S^d M X_T^{d\prime} - X_{S_n}^d M_n X_{T_n}^{d}{}'\| \leq \gamma$. In other words, as long as we select a subspace dimension d such that $d \leq d_{max}$, the solution M^* is stable and not over-fitting.

3.4. Divergence between source and target domains

The pioneer work of Ben-David et al. [1] provides a generalization bound on the target error which depends on the source error and a measure of divergence, called the $H\Delta H$ divergence, between the source and target distributions $\mathcal{D}_S$ and $\mathcal{D}_T$.

$$\epsilon_T(h) = \epsilon_S(h) + d_{H\Delta H}(\mathcal{D}_S, \mathcal{D}_T) + \lambda, \tag{6}$$

where h is a learned hypothesis, $\epsilon_T(h)$ the generalization target error, $\epsilon_S(h)$ the generalization source error, and λ the error of the ideal joint hypothesis on S and T, which is supposed to be a negligible term if the adaptation is possible. Eq. 6 tells us that to adapt well, one has to learn a hypothesis which works well on S while reducing the divergence between $\mathcal{D}_S$ and $\mathcal{D}_T$. To estimate $d_{H\Delta H}(\mathcal{D}_S, \mathcal{D}_T)$, a usual way consists in learning a linear classifier h to discriminate between source and target instances, respectively pseudo-labeled with 0 and 1. In this context, the higher the error of h, the smaller the divergence. While such a strategy gives us some insight about the ability for a *global* learning algorithm (e.g. SVM) to be efficient on both domains, it does not seem to be suited to deal with *local* classifiers, such as the k-nearest neighbors. To overcome this limitation, we introduce a new empirical divergence specifically dedicated to local classifiers. Based on the recommendations of [2], we propose a discrepancy measure to estimate the local density of a target point w.r.t. a given source point. This discrepancy, called *Target density around source* **TDAS** counts how many target points can be found on average within a ϵ neighborhood of a source point. More formally:

$$TDAS = \frac{1}{n_S}\sum_{\forall \mathbf{y_S}} |\{\mathbf{y_T}|Sim(\mathbf{y_S}, \mathbf{y_T}) \geq \epsilon\}|. \tag{7}$$

Note that **TDAS** is associated with similarity measure $Sim(\mathbf{y_S}, \mathbf{y_T}) = \mathbf{y_S} A \mathbf{y_T}'$ where A is the learned metric. As we will see in the next section, **TDAS** can be used to evaluate the effectiveness of a DA method under the covariate shift assumption and probabilistic Lipschitzness assumption [2]. The larger the TDAS, the better the DA method.

4. Experiments

We evaluate our method in the context of object recognition using a standard dataset and protocol for evaluating visual domain adaptation methods as in [5, 7, 8, 10, 14]. In addition, we also evaluate our method using various other image classification datasets.

4.1. DA datasets and data preparation

We provide three series of experiments on different datasets. In the first series, we use the Office dataset [14] and Caltech10 [7] dataset that contain four domains altogether to evaluate all DA methods. The Office dataset consists of images from web-cam (denoted by **W**), DSLR images (denoted by **D**) and Amazon images (denoted by **A**). The Caltech10 images are denoted by **C**. We follow the same setup as in [7]. We use each source of images as a domain, consequently we get four domains (**A**, **C**, **D** and **W**) leading to 12 DA problems. We denote a DA problem by the notation $S \rightarrow T$. We use the image representations provided by [7] for Office and Caltech10 datasets (SURF features encoded with a visual dictionary of 800 words). We follow the standard protocol of [7, 8, 10, 14] for generating the source and target samples[1].

In a second series, we evaluate the effectiveness of our DA method using other datasets, namely ImageNet (**I**), LabelMe (**L**) and Caltech-256 (**C**). In this setting we consider each dataset as a domain. We select five common objects (bird, car, chair, dog and person) for all three datasets leading to a total of 7719 images. We extract dense SIFT features and create a bag-of-words dictionary of 256 words using kmeans. Afterwards, we use LLC encoding and a spatial pyramid (2×2 quadrants + 3×1 horizontal + 1 full image) to obtain a 2048 dimensional image representation (similar data preparation as in [9]).

In the last series, we evaluate the effectiveness of our DA method using larger datasets, namely PASCAL-VOC-2007 and ImageNet. We select all the classes of PASCAL-VOC-2007. The objective here is to classify PASCAL-VOC-2007 test images using classifiers that are built from the ImageNet dataset. To prepare the data, we extract dense SIFT features and create a bag-of-words dictionary of 256 using only ImageNet images. Afterwards, we use LLC encoding and spatial pyramids ($2{\times}2 + 3{\times}1 + 1$) to obtain a 2048 dimensional image representation.

4.2. Experimental setup

We compare our subspace DA approach with two other DA methods and three baselines. Each of these methods defines a new representation space and our goal is to compare the performance of a 1-Nearest-Neighbor (NN) classifier and a SVM classifier on DA problems in the subspace found.

We consider the DA methods Geodesic Flow Kernel (**GFK** [7]) and Geodesic Flow Sampling (**GFS** [8]). They have indeed demonstrated state of the art performances achieving better results than metric learning methods [14] and better than those reported by Chang's method in [5]. Moreover, these methods are the closest to our approach. We also report results obtained by the following three baselines: **Baseline 1:** where we use the projection defined by the PCA subspace X_S built from the source domain to project both source and target data and work in the resulting representation. **Baseline 2:** where we use similarly the projection defined by the PCA subspace X_T built from the target domain. No adaptation **NA:** where no projection is made, we use the original input space without learning a new representation.

For each method, we compare the performance of a 1-Nearest-Neighbor (NN) classifier and of a SVM classifier (with C parameter set to the mean similarity value obtained from the training set) in the subspace defined by each method. For each source-target DA problem in the first two series of experiments, we evaluate the accuracy of each method on the target domain over 20 random trials. For each trial, we consider an unsupervised DA setting where we randomly sample labeled data in the source domain as training data and unlabeled data in the target domain as testing examples. In the last series involving the PASCAL-VOC dataset, we rather evaluate the approaches by measuring the mean average precision over target data using SVM.

We have also compared the behavior of the approaches in a semi-supervised scenario by adding 3 labelled target examples to the training set for Office+Caltech10 series and 50 for the PASCAL-VOC series. This can be found in the supplementary material.

4.3. Selecting the optimal dimensionality

In this section, we present our procedure for selecting the space dimensionality d in the context of our method. The same dimensionality is used for Baseline1 and Baseline2. For GFK and GFS we follow the published procedures to obtain optimal results as presented in [7]. First, we perform a PCA on the two domains and compute the deviation $\lambda_d^{min} - \lambda_{d+1}^{min}$ for all possible d values. Then, using the theoretical bound of Eq: 5, we can estimate a $d_{max} << D$ that provides a stable solution with fixed deviation $\gamma > 0$ for a given confidence $\delta > 0$. Afterwards, we consider the subspaces of dimensionality from $d = 1$ to d_{max} and select the best d^* that minimizes the classification error using a 2 fold cross-validation over the labelled source data. This procedure is founded by the theoretical result of Ben-David et al. of Eq 6 where the idea is to try to move the domain distribution closer while maintaining a good accuracy on the source domain. As an illustration, the best dimensions for the Office dataset vary between $10 - 50$. For example, for the DA problem $\mathbf{W} \rightarrow \mathbf{C}$, taking $\gamma = 10^5$ and $\delta = 0.1$, we obtain $d_{max} = 22$ (see Figure 2) and by cross validation we found that the optimal dimension is $d^* = 20$.

4.4. Evaluating DA with divergence measures

Here, we propose to evaluate the capability of our method to move the domain distributions closer according to the measures presented in Section 3.4: the TDAS adapted to NN classification where a high value indicates a better distribution closeness and the $H\Delta H$ using a SVM where a value close to 50 indicates close distributions. We compute these discrepancy measures for the 12 DA problems coming from the Office and Caltech datasets and report the mean values over the 12 problems for each method in Table 1. We can remark that our approach reduces significantly the discrepancy between the source and target domains com-

[1] See supplementary material section 1.1 for the experimental details and additional results.

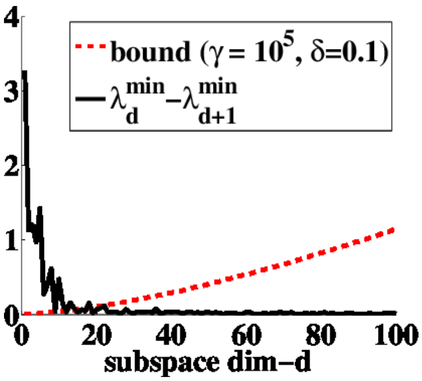

Figure 2. Finding a stable solution and a subspace dimensionality using the consistency theorem.

Method	NA	Baseline 1	Baseline 2	GFK	OUR
TDAS	1.25	3.34	2.74	2.84	**4.26**
H∆H	98.1	99.0	99.0	74.3	**53.2**

Table 1. Several distribution discrepancy measures averaged over 12 DA problems using Office dataset.

pared to the other baselines (highest TDAS value and lowest $H\Delta H$ measure). Both GFK and our method have lower $H\Delta H$ values meaning that these methods are more likely to perform well[2].

4.5. Classification Results

Visual domain adaptation performance with Office/Caltech10 datasets: In this experiment we evaluate the different methods using Office [14]/Caltech10 [8] datasets which consist of four domains (**A**, **C**, **D** and **W**). The results for the 12 DA problems in the unsupervised setting using a NN classifier are shown in Table 2. In 9 out of the 12 DA problems our method outperforms the other ones. The results obtained in the semi-supervised DA setting (see supplementary material) confirm this behavior. Here our method outperforms the others in 10 DA problems.

The results obtained with a SVM classifier in the unsupervised DA case are shown in Table 3. Our method outperforms all the other methods in 11 DA problems. These results indicate that our method works better than other DA methods not only for NN-like local classifiers but also with more global SVM classifiers.

Domain adaptation on ImageNet, LabelMe and Caltech-256 datasets : Results obtained for unsupervised DA using NN classifiers are shown in Table 4. First, we can remark that all the other DA methods achieve poor accuracy when LabelMe images are used as the source domain, while our method seems to adapt the source to the target reasonably well. On average, our method significantly outperforms all other DA methods.

A visual example where we classify ImageNet images using Caltech-256 images is shown in Figure 1. The nearest neighbor coming from Caltech-256 corresponds to the same class, even though the appearance of images are very different from the two datasets.

Method	C→A	D→A	W→A	A→C	D→C	W→C
NA	21.5	26.9	20.8	22.8	24.8	16.4
Baseline 1	38.0	29.8	35.5	30.9	29.6	31.3
Baseline 2	**40.5**	33.0	**38.0**	33.3	31.2	31.9
GFS [8]	36.9	32	27.5	35.3	29.4	21.7
GFK [7]	36.9	32.5	31.1	**35.6**	29.8	27.2
OUR	39.0	**38.0**	37.4	35.3	**32.4**	**32.3**
Method	A→D	C→D	W→D	A→W	C→W	D→W
NA	22.4	21.7	40.5	23.3	20.0	53.0
Baseline 1	34.6	37.4	71.8	35.1	33.5	74.0
Baseline 2	34.7	36.4	72.9	36.8	34.4	78.4
GFS [8]	30.7	32.6	54.3	31.0	30.6	66.0
GFK [7]	35.2	35.2	70.6	34.4	33.7	74.9
OUR	**37.6**	**39.6**	**80.3**	**38.6**	**36.8**	**83.6**

Table 2. Recognition accuracy with unsupervised DA using a NN classifier (Office dataset + Caltech10).

Method	C→A	D→A	W→A	A→C	D→C	W→C
Baseline 1	44.3	36.8	32.9	36.8	29.6	24.9
Baseline 2	44.5	38.6	34.2	37.3	31.6	28.4
GFK	44.8	37.9	37.1	38.3	31.4	29.1
OUR	**46.1**	**42.0**	**39.3**	**39.9**	**35.0**	**31.8**
Method	A→D	C→D	W→D	A→W	C→W	D→W
Baseline 1	36.1	38.9	73.6	**42.5**	34.6	75.4
Baseline 2	32.5	35.3	73.6	37.3	34.2	80.5
GFK	37.9	36.1	74.6	39.8	34.9	79.1
OUR	**38.8**	**39.4**	**77.9**	39.6	**38.9**	**82.3**

Table 3. Recognition accuracy with unsupervised DA using a SVM classifier(Office dataset + Caltech10).

Method	L→C	L→I	C→L	C→I	I→L	I→C	AVG
NA	46.0	38.4	29.5	31.3	36.9	45.5	37.9
Baseline1	24.2	27.2	46.9	41.8	35.7	33.8	34.9
Baseline2	24.6	27.4	**47.0**	**42.0**	35.6	33.8	35.0
GFK	24.2	26.8	44.9	40.7	35.1	33.8	34.3
OUR	**49.1**	**41.2**	**47.0**	39.1	**39.4**	**54.5**	**45.0**

Table 4. Recognition accuracy with unsupervised DA with NN classifier (ImageNet (I), LabelMe (L) and Caltech-256 (C)).

In Table 5 we report results using a SVM classifier for the unsupervised DA setting. In this case our method outperforms all other DA methods, confirming the good behavior of our approach.

Classifying PASCAL-VOC-2007 images using classifiers built on ImageNet : In this experiment, we compare the average precision obtained on PASCAL-VOC-2007 by a SVM classifier in both unsupervised and semi-supervised DA settings. We use ImageNet as the source domain and PASCAL-VOC-2007 as the target domain. The results are shown in Figure 3 for the unsupervised case and in the sup-

[2] See section 1.4 of supplementary material for more details.

Method	L→C	L→I	C→L	C→I	I→L	I→C	AVG
NA	49.6	40.8	36.0	45.6	41.3	58.9	45.4
Baseline1	50.5	42.0	39.1	48.3	44.0	59.7	47.3
Baseline2	48.7	41.9	39.2	48.4	43.6	58.0	46.6
GFK	52.3	43.5	39.6	49.0	45.3	61.8	48.6
OUR	**52.9**	**43.9**	**43.8**	**50.9**	**46.3**	**62.8**	**50.1**

Table 5. Recognition accuracy with unsupervised DA with SVM classifier (ImageNet (I), LabelMe (L) and Caltech-256 (C)).

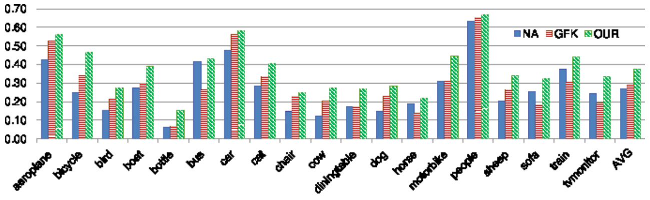

Figure 3. Train on ImageNet and classify PASCAL-VOC-2007 images using unsupervised DA with SVM.

plementary material for the semi-supervised one.

Our method achieves the best results for all the categories in both settings and outperforms all the methods on average. The semi-supervised DA seems to improve unsupervised DA by 10% (relative) in mAP. In the unsupervised DA setting, GFK improves by 7% in mAP over no adaptation while our method improves by 27% in mAP over GFK. In the semi-supervised setting our method improves by 13% in mAP over GFK and by 46% over no adaptation.

5. Conclusion

We present a new visual domain adaptation method using subspace alignment. In this method, we create subspaces for both source and target domains and learn a linear mapping that aligns the source subspace with the target subspace. This allows us to compare the source domain data directly with the target domain data and to build classifiers on source data and apply them on the target domain. We demonstrate excellent performance on several image classification datasets such as Office dataset, Caltech, ImageNet, LabelMe and Pascal-VOC. We show that our method outperforms state of the art domain adaptation methods using both SVM and nearest neighbour classifiers. We experimentally show that our method can be used on tasks such as labelling PASCAL-VOC images using ImageNet dataset for training. Due to its simplicity and theoretically founded stability, we believe that our method has the potential to be applied on large datasets consisting of millions of images. As future work we plan to extend our domain adaptation method to large scale image retrieval and on the fly learning of classifiers.

Acknowledgements : The authors acknowledge the support of the FP7 ERC Starting Grant 240530 COGNIMUND, ANR LAMPADA 09-EMER-007-02 project and PASCAL 2 network of Excellence.

References

[1] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. In *NIPS*. 2007.

[2] S. Ben-David, S. Shalev-Shwartz, and R. Urner. Domain adaptation–can quantity compensate for quality? In *International Symposium on Artificial Intelligence and Mathematics*, 2012.

[3] J. Blitzer, D. Foster, and S. Kakade. Domain adaptation with coupled subspaces. In *Conference on Artificial Intelligence and Statistics*, 2011.

[4] J. Blitzer, R. McDonald, and F. Pereira. Domain adaptation with structural correspondence learning. In *Conference on Empirical Methods in Natural Language Processing*, 2006.

[5] S.-F. Chang. Robust visual domain adaptation with low-rank reconstruction. In *CVPR*, 2012.

[6] B. Chen, W. Lam, I. Tsang, and T.-L. Wong. Extracting discriminative concepts for domain adaptation in text mining. In *ACM SIGKDD*, 2009.

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

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

[9] A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba. Undoing the damage of dataset bias. In *ECCV*, 2012.

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

[11] A. Margolis. A literature review of domain adaptation with unlabeled data. Technical report, University of Washington, 2011.

[12] S. J. Pan, J. T. Kwok, and Q. Yang. Transfer learning via dimensionality reduction. In *AAAI*, 2008.

[13] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. In *IJCAI*, 2009.

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

[15] A. Torralba and A. Efros. Unbiased look at dataset bias. In *CVPR*, 2011.

[16] C. Wang and S. Mahadevan. Manifold alignment without correspondence. In *IJCAI*, 2009.

[17] C. Wang and S. Mahadevan. Heterogeneous domain adaptation using manifold alignment. In *IJCAI*, 2011.

[18] D. Zhai, B. Li, H. Chang, S. Shan, X. Chen, and W. Gao. Manifold alignment via corresponding projections. In *BMVC*, 2010.

[19] L. Zwald and G. Blanchard. On the convergence of eigenspaces in kernel principal components analysis. In *NIPS*, 2005.