To appear, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012.

Geodesic Flow Kernel for Unsupervised Domain Adaptation

Boqing Gong, Yuan Shi, Fei Sha
Dept. of Computer Science
U. of Southern California
{boqinggo, yuanshi, feisha}@usc.edu

Kristen Grauman
Dept. of Computer Science
U. of Texas at Austin
grauman@cs.utexas.edu

Abstract

In real-world applications of visual recognition, many factors—such as pose, illumination, or image quality—can cause a significant mismatch between the source *domain on which classifiers are trained and the* target *domain to which those classifiers are applied. As such, the classifiers often perform poorly on the target domain. Domain adaptation techniques aim to correct the mismatch. Existing approaches have concentrated on learning feature representations that are invariant across domains, and they often do not directly exploit low-dimensional structures that are intrinsic to many vision datasets. In this paper, we propose a new kernel-based method that takes advantage of such structures. Our geodesic flow kernel models domain shift by integrating an infinite number of subspaces that characterize changes in geometric and statistical properties from the source to the target domain. Our approach is computationally advantageous, automatically inferring important algorithmic parameters without requiring extensive cross-validation or labeled data from either domain. We also introduce a metric that reliably measures the adaptability between a pair of source and target domains. For a given target domain and several source domains, the metric can be used to automatically select the optimal source domain to adapt and avoid less desirable ones. Empirical studies on standard datasets demonstrate the advantages of our approach over competing methods.*

1. Introduction

Imagine that we are to deploy an Android application to recognize objects in images captured with mobile phone cameras. Can we train classifiers with Flickr photos, as they have already been collected and annotated, and hope the classifiers still work well on mobile camera images?

Our intuition says no. We suspect that the strong distinction between Flickr and mobile phone images will cripple those classifiers. Indeed, a stream of studies have shown that when image classifiers are evaluated outside of their training datasets, the performance degrades significantly [27, 9, 24]. Beyond image recognition, mismatched training and testing conditions are also abundant: in other computer vision tasks [10, 28, 19, 11], speech and language processing [21, 4, 5], and others.

All these pattern recognition tasks involve two distinct types of datasets, one from a source domain and the other from a target domain. The ***source domain*** contains a large amount of labeled data such that a classifier can be reliably built. The ***target domain*** refers broadly to a dataset that is assumed to have different characteristics from the source domain. The main objective is to adapt classifiers trained on the source domain to the target domain to attain good performance there. Note that we assume the set of possible labels are the same across domains.

Techniques for addressing this challenge have been investigated under the names of domain adaptation, covariate shift, and transfer learning. There are two settings: ***unsupervised domain adaptation*** where the target domain is completely unlabeled, and ***semi-supervised domain adaptation*** where the target domain contains a small amount of labeled data. Often the labeled target data alone is insufficient to construct a good classifier. Thus, how to effectively leverage *unlabeled* target data is key to domain adaptation.

A very fruitful line of work has been focusing on deriving new feature representations to facilitate domain adaptation, where labeled target data is not needed [7, 2, 5, 4, 22, 14]. The objective is to identify a new feature space such that the source domain and the target domain manifest shared characteristics. Intuitively, if they were indistinguishable, a classifier constructed for the source domain would work also for the target domain.

Defining and quantifying shared characteristics entails careful examination of our intuition on what type of representations facilitate adaptation. For example, in the part-of-speech (POS) task of tagging words into different syntactic categories [5], the idea is to extract shared patterns from auxiliary classification tasks that predict "pivot features", frequent words that are themselves indicative of those categories. While sensible for language processing tasks, typi-

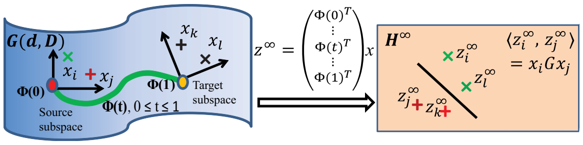

Figure 1. Main idea of our geodesic flow kernel-based approach for domain adaptation (Best viewed in color). We embed source and target datasets in a Grassmann manifold. We then construct a geodesic flow between the two points and integrate an infinite number of subspaces along the flow $\mathbf{\Phi}(t)$. Concretely, raw features are projected into these subspaces to form an infinite-dimensional feature vector $\boldsymbol{z}^\infty \in \mathcal{H}^\infty$. Inner products between these feature vectors define a kernel function that can be computed over the original feature space in closed-form. The kernel encapsulates incremental changes between subspaces that underly the difference and commonness between the two domains. The learning algorithms thus use this kernel to derive low-dimensional representations that are invariant to the domains.

cal histogram based features of low-level visual descriptors do not enjoy having pivot "visual words" — in general, no single feature dimension from a particular histogram bin is discriminative enough to differentiate visual categories.

On the other hand, many visual data *are* assumed to lie in low-dimensional subspaces. Given data from two domains, *how can we exploit the subspaces in these datasets, which can be telltale in revealing the underlying difference and commonness between the domains?*

Moreover, given multiple source domains and a target domain, *how can we select which source domain to pair with the target domain?* This is an especially important problem to address in order to apply domain adaptation to real-world problems. For instance, in the context of object recognition, we can choose from multiple datasets as our source domain: ImageNet, Caltech-101/256, PASCAL VOC, etc. It is much more cost-effective to be able to select one (or a limited few) that are likely to adapt well to the target domain, instead of trying each one of them.

To address the first challenge, we propose a kernel-based method for domain adaptation. The proposed *geodesic flow kernel* is computed over the subspaces of the source and the target domains. It integrates an infinite number of subspaces that lie on the geodesic flow from the source subspace to the target one. The flow represents incremental changes in geometric and statistical properties between the two domains. Being mindful of all these changes, our learning algorithm extracts those subspace directions that are truly domain-invariant. Fig. 1 sketches the main idea.

To address the second challenge, we introduce a metric called Rank of Domain (ROD) that can be used to rank a list of source domains based on how suitable they are to domain adaptation. The metric integrates two pieces of information: how much the subspaces of the source and the target domains overlap, and how similarly the target and source data are distributed in the subspaces. In our experiments, ROD correlates well with adaptation performance.

We demonstrate the effectiveness of the proposed approaches on benchmark tasks of object recognition. The proposed methods outperform significantly state-of-the-art methods for domain adaptation. Additionally, as a novel application of these methods, we investigate the *dataset bias* problem, recently studied in [27]. Through their analysis, the authors identified a few datasets of high "market value", suggesting that they are less biased, and more representative of real-world objects. We re-examine these datasets with a new perspective: *are such high-valued datasets indeed useful in improving a target domain's performance?* Our analysis suggests it would be beneficial to also consider "ease of adaptability" in assessing the value of datasets.

Contributions. To summarize, our main contributions are: i) a kernel-based domain adaptation method that exploits intrinsic low-dimensional structures in the datasets (section 3.3); the method is easy to implement, with no parameters to cross-validate (sections 3.4 and 4.4); ii) a metric that can predict which source domain is better suited for adaptation to a target domain, without using labeled target data (sections 3.5 and 4.5); iii) empirical studies validating the advantages of our approaches over existing approaches on benchmark datasets (section 4.2 and 4.3); iv) a new perspective from re-examining cross-dataset generalization using domain adaptation (section 4.6).

2. Related Work

Domain adaptation has been extensively studied in many areas, including in statistics and machine learning [26, 18, 2, 23], speech and language processing [7, 5, 21], and more recently computer vision [3, 14, 25, 20].

Of particular relevance to our work is the idea of learning new feature representations that are domain-invariant, thus enabling transferring classifiers from the source domain to the target domain [2, 5, 4, 7, 22]. Those approaches are especially appealing to unsupervised domain adaptation as they do not require labeled target data. Other methods for unsupervised domain adaptation have been explored, for example, with transductive SVMs [3] or iteratively relabeling (the target domain) [6]. Note that the latter approach depends very much on tuning several parameters, which requires extensive computation of training many SVMs.

Gopalan *et al*'s work is the closest to ours in spirit [14]. They have also explored the idea of using geodesic flows to derive intermediate subspaces that interpolate between the source and target domains. A crucial difference of that work from ours is that they sample a *finite* number of subspaces and stack these subspaces into a very high-dimensional projection matrix. Our kernel method is both conceptually and computationally simpler and eliminates the need to tune many parameters needed in Gopalan *et al*'s approach. We will return to the comparison after we describe both approaches in section 3.

3. Proposed Approach

The main idea behind our approach is to explicitly construct an infinite-dimensional feature space $\mathcal{H}^\infty$ that assembles information on the source domain $\mathcal{D}_\mathcal{S}$, on the target domain $\mathcal{D}_\mathcal{T}$, and on "phantom" domains interpolating between those two — the nature of the interpolation will be made more precise later. Inner products in $\mathcal{H}^\infty$ give rise to a kernel function that can be computed efficiently in closed-form. Thus, this geodesic flow kernel (GFK) can be readily used to construct any kernelized classifiers.

We start by reviewing basic notions of Grassmann manifolds; the subspaces of the data from the source and target domains are represented as two points on one such manifold. We then discuss a previous approach where multiple subspaces are sampled from the manifold to derive new feature representations. Then in section 3.3, we describe our approach in detail and contrast to the previous one.

The dimensionality of the subspaces is an important parameter. In section 3.4, we present a subspace disagreement measure (SDM) for selecting this parameter **automatically** without cross-validation. Finally, in section 3.5, we describe a Rank of Domain (ROD) metric that computes compatibility between two domains for adaptation.

3.1. Background

In statistical modeling, we often assume data can be embedded in a low-dimensional linear subspace. For example, principal component analysis (PCA) identifies the subspace where the variances of the embedded data are maximized. Most of the time, it is both sufficient and convenient to refer to a subspace with its basis $\boldsymbol{P} \in \mathbb{R}^{\mathsf{D}\times\mathsf{d}}$, where D is the dimensionality of the data and d is the dimensionality of the subspace. For PCA, the basis is then the top d eigenvectors of the data's covariance matrix. The collection of all d-dimensional subspaces form the Grassmannian $\mathbb{G}(\mathsf{d},\mathsf{D})$, a smooth Riemannian manifold on which we can define geometric, differential, and probabilistic structures.

As an intuitive example of how Grassmannians can help us to attack the problem of domain adaptation, imagine that we compute the subspaces of the datasets for the $\mathcal{D}_\mathcal{S}$ and $\mathcal{D}_\mathcal{T}$ domains and map them to two points on a Grassmannian. Intuitively, if these two points are close by, then the two domains could be similar to each other, for example, their features may be similarly distributed. Thus, a $\mathcal{D}_\mathcal{S}$-trained classifier is likely to work well on $\mathcal{D}_\mathcal{T}$.

However, *what if these two domains are far apart on the manifold?* We briefly describe an earlier work by Gopalan *et al* [14]. Our method extends and improves upon it.

3.2. Subspaces by sampling geodesic flow (SGF)

Consider two datasets of "Cars" with large differences in poses are placed far apart on the manifold. The key idea is to use intermediate subspaces to learn domain-invariant features to adapt [14]. Specifically, the intermediate subspaces would capture statistics of car images under poses interpolated between the source and the target domain. Being informed of all these different subspaces from the same category, the learning algorithms might be able to extract features that are less sensitive to variations in pose.

Concretely, the approach of sampling geodesic flow (SGF) [14] consists of the following steps: i) construct a geodesic flow curve connecting the source and target domains on the Grassmannian; ii) sample a fixed number of subspaces from this curve; iii) project original feature vectors into these subspaces and concatenate them into feature super-vectors; iv) reduce dimensionality of the super-vectors; v) use the resulting representations as new feature vectors to construct classifiers.

Despite its encouraging results, the SGF approach has several limitations. It is not clear how to choose the best sampling strategy. A few important parameters need to be tuned: the number of subspaces to sample, the dimensionality of the subspaces, and how to cope with the high-dimensionality of the new representations. Critically, cross-validating all these "tweaking knobs" is impractical in typical settings for domain adaptation, where there is little or no labeled target data.

In the following, we show how these limitations can be addressed in a simple kernel-based framework.

3.3. Our approach: geodesic flow kernel (GFK)

Our approach consists of the following steps: i) determine the optimal dimensionality of the subspaces to embed domains; ii) construct the geodesic curve; iii) compute the geodesic flow kernel; iv) use the kernel to construct a classifier with labeled data. We defer describing step i) to the next section and focus on steps ii) and iii).

For step ii), we state only the main computational steps. The detailed derivation can be found in [14] and its references. We also omit step iv) for brevity, as it is the same as constructing any other kernel-based classifier.

Construct geodesic flow Let $\boldsymbol{P}_\mathcal{S}, \boldsymbol{P}_\mathcal{T} \in \mathbb{R}^{\mathsf{D}\times\mathsf{d}}$ denote the two sets of basis of the subspaces for the source and target domains. Let $\boldsymbol{R}_\mathcal{S} \in \mathbb{R}^{\mathsf{D}\times(\mathsf{D}-\mathsf{d})}$ denote the orthogonal complement to $\boldsymbol{P}_\mathcal{S}$, namely $\boldsymbol{R}_\mathcal{S}^{\mathrm{T}}\boldsymbol{P}_\mathcal{S} = \boldsymbol{0}$. Using the canonical Euclidean metric for the Riemannian manifold, the geodesic flow is parameterized as $\boldsymbol{\Phi} : t \in [0,1] \rightarrow \boldsymbol{\Phi}(t) \in G(\mathsf{d},\mathsf{D})$ under the constraints $\boldsymbol{\Phi}(0) = \boldsymbol{P}_\mathcal{S}$ and $\boldsymbol{\Phi}(1) = \boldsymbol{P}_\mathcal{T}$. For other t,

$$\boldsymbol{\Phi}(t) = \boldsymbol{P}_\mathcal{S}\boldsymbol{U}_1\boldsymbol{\Gamma}(t) - \boldsymbol{R}_\mathcal{S}\boldsymbol{U}_2\boldsymbol{\Sigma}(t), \qquad (1)$$

where $\boldsymbol{U}_1 \in \mathbb{R}^{\mathsf{d}\times\mathsf{d}}$ and $\boldsymbol{U}_2 \in \mathbb{R}^{(\mathsf{D}-\mathsf{d})\times\mathsf{d}}$ are orthonormal matrices. They are given by the following pair of SVDs,

$$\boldsymbol{P}_{\mathcal{S}}^{\mathrm{T}}\boldsymbol{P}_{\mathcal{T}} = \boldsymbol{U}_1\boldsymbol{\Gamma}\boldsymbol{V}^{\mathrm{T}}, \quad \boldsymbol{R}_{\mathcal{S}}^{\mathrm{T}}\boldsymbol{P}_{\mathcal{T}} = -\boldsymbol{U}_2\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}} . \tag{2}$$

$\boldsymbol{\Gamma}$ and $\boldsymbol{\Sigma}$ are $\mathsf{d}\times\mathsf{d}$ diagonal matrices. The diagonal elements are $\cos\theta_i$ and $\sin\theta_i$ for $i = 1, 2, \ldots, \mathsf{d}$. Particularly, θ_i are called the principal angles between $\boldsymbol{P}_{\mathcal{S}}$ and $\boldsymbol{P}_{\mathcal{T}}$:

$$0 \leq \theta_1 \leq \theta_2 \leq \cdots \leq \theta_{\mathsf{d}} \leq \pi/2 \tag{3}$$

They measure the degree that subspaces "overlap". Moreover, $\boldsymbol{\Gamma}(t)$ and $\boldsymbol{\Sigma}(t)$ are diagonal matrices whose elements are $\cos(t\theta_i)$ and $\sin(t\theta_i)$ respectively.

Compute geodesic flow kernel (GFK) The geodesic flow parameterizes how the source domain smoothly changes to the target domain. Consider the subspace $\boldsymbol{\Phi}(t)$ for a $t \in (0, 1)$ and compute $\boldsymbol{\Phi}(t)^{\mathrm{T}}\boldsymbol{x}$, ie, the projection of a feature vector $\boldsymbol{x}$ into this subspace. If $\boldsymbol{x}$ is from the source domain and t is close to 1, then the projection will appear more likely coming from the target domain and conversely for t close to 0. Thus, using the projection to build a classifier would result in a model using a set of features that are characteristic of both domains. Hence, this classifier would likely perform well on the target domain.

Which (or which set of) t should we use then? Our answer is surprising at the first glance: *all of them*! Intuitively, by expanding the original features with projections into **all** subspaces, we force a measurement of similarity (as we will be using inner products to construct classifiers) that is robust to any variation that leans either toward the source or towards the target or in between. In other words, the net effect is a representation that is insensitive to idiosyncrasies in either domain. Computationally, however, we cannot use this representation explicitly. Nevertheless, we next show that there is no need to actually compute, store and manipulate infinitely many projections.

For two original D-dimensional feature vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, we compute their projections into $\boldsymbol{\Phi}(t)$ for a continuous t from 0 to 1 and concatenate all the projections into infinite-dimensional feature vectors $\boldsymbol{z}_i^{\infty}$ and $\boldsymbol{z}_j^{\infty}$. The inner product between them defines our geodesic-flow kernel,

$$\langle \boldsymbol{z}_i^{\infty}, \boldsymbol{z}_j^{\infty} \rangle = \int_0^1 (\boldsymbol{\Phi}(t)^{\mathrm{T}}\boldsymbol{x}_i)^{\mathrm{T}}(\boldsymbol{\Phi}(t)^{\mathrm{T}}\boldsymbol{x}_j)\, dt = \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{G}\boldsymbol{x}_j \tag{4}$$

where $\boldsymbol{G} \in \mathbb{R}^{\mathsf{D}\times\mathsf{D}}$ is a positive semidefinite matrix. This is precisely the "kernel trick", where a kernel function induces inner products between infinite-dimensional features.

The matrix $\boldsymbol{G}$ can be computed in a closed-form from previously defined matrices:

$$\boldsymbol{G} = [\boldsymbol{P}_{\mathcal{S}}\boldsymbol{U}_1 \;\; \boldsymbol{R}_{\mathcal{S}}\boldsymbol{U}_2] \begin{bmatrix} \boldsymbol{\Lambda}_1 & \boldsymbol{\Lambda}_2 \\ \boldsymbol{\Lambda}_2 & \boldsymbol{\Lambda}_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_1^{\mathrm{T}}\boldsymbol{P}_{\mathcal{S}}^{\mathrm{T}} \\ \boldsymbol{U}_2^{\mathrm{T}}\boldsymbol{R}_{\mathcal{S}}^{\mathrm{T}} \end{bmatrix} \tag{5}$$

where $\boldsymbol{\Lambda}_1$ to $\boldsymbol{\Lambda}_3$ are diagonal matrices, whose diagonal elements are

$$\lambda_{1i} = 1+\frac{\sin(2\theta_i)}{2\theta_i}, \lambda_{2i} = \frac{\cos(2\theta_i)-1}{2\theta_i}, \lambda_{3i} = 1-\frac{\sin(2\theta_i)}{2\theta_i}. \tag{6}$$

Detailed derivations are given in the Supplementary.

Our approach is both conceptually and computationally simpler when compared to the previous SGF approach. In particular, we do not need to tune any parameters — the only free parameter is the dimensionality of the subspaces d, which we show below how to automatically infer.

3.4. Subspace disagreement measure (SDM)

For *unsupervised* domain adaptation, we must be able to select the optimal d automatically, with unlabeled data only. We address this challenge by proposing a subspace disagreement measure (SDM).

To compute SDM, we first compute the PCA subspaces of the two datasets, $\mathbf{PCA}_{\mathcal{S}}$ and $\mathbf{PCA}_{\mathcal{T}}$. We also combine the datasets into one dataset and compute its subspace $\mathbf{PCA}_{\mathcal{S}+\mathcal{T}}$. Intuitively, if the two datasets are similar, then all three subspaces should not be too far away from each other on the Grassmannian. The SDM captures this notion and is defined in terms of the principal angles (cf. eq. (3)),

$$\mathcal{D}(\mathsf{d}) = 0.5\,[\sin\alpha_{\mathsf{d}} + \sin\beta_{\mathsf{d}}] \tag{7}$$

where α_{d} denotes the d-th principal angle between the $\mathbf{PCA}_{\mathcal{S}}$ and $\mathbf{PCA}_{\mathcal{S}+\mathcal{T}}$ and β_{d} between $\mathbf{PCA}_{\mathcal{T}}$ and $\mathbf{PCA}_{\mathcal{S}+\mathcal{T}}$. $\sin\alpha_{\mathsf{d}}$ or $\sin\beta_{\mathsf{d}}$ is called the minimum correlation distance [16].

Note that $\mathcal{D}(\mathsf{d})$ is at most 1. A small value indicates that both α_{d} and β_{d} are small, thus $\mathbf{PCA}_{\mathcal{S}}$ and $\mathbf{PCA}_{\mathcal{T}}$ are aligned (at the d-th dimension). At its maximum value of 1, the two subspaces have orthogonal directions (i.e., $\alpha_d = \beta_d = \pi/2$). In this case, domain adaptation will become difficult as variances captured in one subspace would not be able to transfer to the other subspace.

To identify the optimal d, we adopt a greedy strategy:

$$\mathsf{d}^* = \min\{\mathsf{d}|\mathcal{D}(\mathsf{d}) = 1\}. \tag{8}$$

Intuitively, the optimal d^* should be as high as possible (to preserve variances in the source domain for the purpose of building good classifiers) but should not be so high that the two subspaces start to have orthogonal directions.

3.5. Rank of domain (ROD)

Imagine we need to build a classifier for a target domain for object recognition. We have several datasets, Caltech-101, PASCAL VOC, and ImageNet to choose from as the source domain. Without actually running our domain adaptation algorithms and building classifiers, is it possible to determine which dataset(s) would give us the best performance on the target domain?

To answer this question, we introduce a Rank of Domain (ROD) metric that integrates two sets of information: geometrically, the alignment between subspaces, and statistically, KL divergences between data distributions once they are projected into the subspaces.

We sketch the main idea in the following; the detailed derivation is described in the Supplementary. Given a pair of domains, computing ROD involves 3 steps: i) determine the optimal dimensionality d^* for the subspaces (as in section 3.4); ii) at each dimension $i \leq \mathsf{d}^*$, approximate the data distributions of the two domains with two one-dimensional Gaussians and then compute the symmetrized KL divergences between them; iii) compute the KL-divergence weighted average of principal angles, namely,

$$\mathcal{R}(\mathcal{S}, \mathcal{T}) = \frac{1}{\mathsf{d}^*} \sum_{i}^{\mathsf{d}^*} \theta_i \left[KL(\mathcal{S}_i \| \mathcal{T}_i) + KL(\mathcal{T}_i \| \mathcal{S}_i) \right]. \quad (9)$$

$\mathcal{S}_i$ and $\mathcal{T}_i$ are the two above-mentioned Gaussian distributions; they are estimated from data projected onto the principal vectors (associated with the i-th principal angle).

A pair of domains with smaller values of $\mathcal{R}(\mathcal{S}, \mathcal{T})$ are more likely to adapt well: the two domains are both geometrically well-aligned (small principal angles) and similarly distributed (small KL divergences). Empirically, when we use the metric to rank various datasets as source domains, we find the ranking correlates well with their relative performance improvements on the target domain.

4. Experiments

We evaluate our methods in the context of object recognition. We first compare our geodesic-flow kernel method to baselines and other domain adaptation methods [25, 14]. We then report results that validate our automatic procedure of selecting the optimal dimensionality of subspaces (section 3.4). Next we report results to demonstrate our Rank of Domain (ROD) metric predicts well which source domain is more suitable for domain adaptation. At last, we re-examine the dataset bias problem, recently studied in [27], from the perspective of "ease of adaptability".

4.1. Setup

Our experiments use the three datasets which were studied in [25]: Amazon (images downloaded from online merchants), Webcam (low-resolution images by a web camera), and DSLR (high-resolution images by a digital SLR camera). Additionally, to validate the proposed methods on a wide range of datasets, we added Caltech-256 [15] as a fourth dataset. We regard each dataset as a domain.

We extracted 10 classes common to all four datasets: BACKPACK, TOURING-BIKE, CALCULATOR, HEADPHONES, COMPUTER-KEYBOARD, LAPTOP-101, COMPUTER-MONITOR, COMPUTER-MOUSE, COFFEE-MUG, AND VIDEO-PROJECTOR. There are 8 to 151 samples per category per domain, and 2533 images in total. Fig. 2 highlights the differences among these domains with example images from the category of MONITOR.

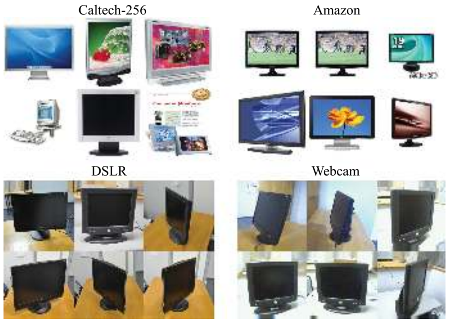

Figure 2. Example images from the MONITOR category in Caltech-256, Amazon, DSLR, and Webcam. Caltech and Amazon images are mostly from online merchants, while DSLR and Webcam images are from offices. (Best viewed in color.)

We report in the main text our results on the 10 common classes. Moreover, we report in the Supplementary our results on 31 categories common to Amazon, Webcam and DSLR, to compare directly to published results [25, 20, 14]. Our results on either the 10 or 31 common classes demonstrate the same trend that the proposed methods significantly outperform existing approaches.

We follow similar feature extraction and experiment protocols used in previous work. Briefly, we use SURF features [1] and encode the images with 800-bin histograms with the codebook trained from a subset of Amazon images. The histograms are normalized first and then z-scored to have zero mean and unit standard deviation in each dimension. For each pair of source and target domains, we conduct experiments in 20 random trials. In each trial, we randomly sample labeled data in the source domain as training examples, and unlabeled data in the target domain as testing examples. In semi-supervised domain adaptation, we also sample a small number of images from the target domain to augment the training set. More details on how data are split are given in the Supplementary. We report averaged accuracies on target domains as well as standard errors.

1-nearest neighbor is used as our classifier as it does not require cross-validating parameters. For our algorithms, the dimensionality of subspaces are selected according to the criterion in section 3.4. For methods we compare to, we use what is recommended in the published work.

4.2. Results on unsupervised adaptation

Our baseline is **OrigFeat**, where we use original features, ie., without learning a new representation for adaptation. Other types of baselines are reported in the Suppl.

For our methods, we use two types of subspaces for the

Table 1. Recognition accuracies on target domains with *unsupervised* adaptation (C: Caltech, A: Amazon, W: Webcam, and D: DSLR)

Method	C → A	C → D	A → C	A → W	W → C	W → A	D → A	D → W
OrigFeat	20.8±0.4	22.0±0.6	22.6±0.3	23.5±0.6	16.1±0.4	20.7±0.6	27.7±0.4	53.1±0.6
SGF[14]	*36.8*±0.5	32.6±0.7	*35.3*±0.5	31.0±0.7	21.7±0.4	27.5±0.5	32.0±0.4	66.0±0.5
GFK(PCA, PCA)	*36.9*±0.4	*35.2*±1.0	*35.6*±0.4	**34.4**±0.9	*27.2*±0.5	*31.1*±0.7	*32.5*±0.5	*74.9*±0.6
GFK(PLS, PCA)	**40.4**±0.7	**41.1**±1.3	**37.9**±0.4	**35.7**±0.9	**29.3**±0.4	**35.5**±0.7	**36.1**±0.4	**79.1**±0.7

Table 2. Recognition accuracies on target domains with *semi-supervised* adaptation (C: Caltech, A: Amazon, W: Webcam, and D: DSLR)

Method	C → A	C → D	A → C	A → W	W → C	W → A	D → A	D → W
OrigFeat	23.1±0.4	26.5±0.7	24.0±0.3	31.6±0.6	20.8±0.5	30.8±0.6	31.3±0.7	55.5±0.7
Metric[25]	33.7±0.8	35.0±1.1	27.3±0.7	36.0±1.0	21.7±0.5	32.3±0.8	30.3±0.8	55.6±0.7
SGF[14]	40.2±0.7	36.6±0.8	*37.7*±0.5	37.9±0.7	29.2±0.7	38.2±0.6	39.2±0.7	69.5±0.9
GFK(PCA, PCA)	*42.0*±0.5	*49.5*±0.8	*37.8*±0.4	*53.7*±0.8	**32.8**±0.7	**42.8**±0.7	*45.0*±0.7	*78.7*±0.5
GFK(PLS, PCA)	**46.1**±0.6	**55.0**±0.9	**39.6**±0.4	**56.9**±1.0	**32.1**±0.7	**46.2**±0.7	**46.2**±0.6	**80.2**±0.4
GFK(PLS, PLS)	*38.7*±0.6	38.6±1.4	*36.6*±0.4	36.3±0.9	28.6±0.6	36.3±0.5	35.0±0.4	74.6±0.5

source data: **PCA** which is the PCA subspace and **PLS** which is the Partial Least Squares (PLS) subspace. PLS is similar to PCA except it takes label information into consideration, and thus can be seen as a form of supervised dimension reduction [17]. For the target domains, we use only **PCA** as there is no label. Thus, there are two variants of our kernel-based method: **GFK(PCA, PCA)** and **GFK(PLS, PCA)**.

We also implement the method described in section 3.2 [14]. We refer to it as **SGF**. As the authors of this method suggest, we use the PCA subspaces for both domains. We also use the parameter settings reported in [14].

Table 1 summarizes the classification accuracies as well as standard errors of all the above methods for different pairings of the source and target domains. We report 8 pairings; the rest are reported in the Supplementary. The best group (differences up to one standard error) in each column are in bold font and the second best group (differences up to one standard error) are in italics and underlined.

All domain adaptation methods improve accuracy over the baseline **OrigFeat**. Further, our **GFK** based methods in general outperform **SGF**. Moreover, **GFK(PLS, PCA)** performs the best. Two key factors may contribute to the superiority of our method: i) the kernel integrates all the subspaces along the flow, and is hence able to model better the domain shift between the source and the target; ii) this method uses a discriminative subspace (by PLS) in the source domain to incorporate the label information. This has the benefit of avoiding projection directions that contain noise and very little useful discriminative information, albeit making source and target domains look similar. PCA, on the other hand, does not always yield subspaces that contain discriminative information. Consequently all the improvements by our **GFK(PLS, PCA)** over **SGF** are statistically significant, with margins more than one standard error.

For a given target domain, there is a preferred source domain which leads to the best performance, either using **OrigFeat** or any of the domain adaptation methods. For example, for the domain Webcam, the source domain DSLR is better than the domain Amazon. This might be attributed to the similarity in DSLR and Webcam, illustrated in Fig. 2. We analyze this in detail in section 4.5.

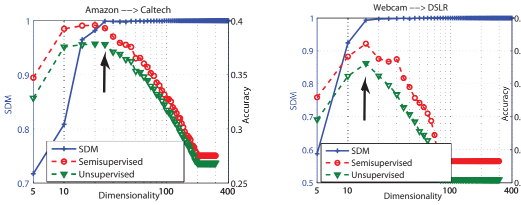

Figure 3. Selecting the optimal dimensionality d^* with SDM (sec. 3.4); selected d^* (where the arrows point to) leads to the best adaptation performance. (Best viewed in color)

4.3. Results on semi-supervised adaptation

In semi-supervised adaptation, our algorithms have access to a small labeled set of target data. Therefore, we also compare to **GFK(PLS, PLS)**, and the metric learning based method **Metric** [25] which uses the correspondence between source and target labeled data to learn a Mahalanobis metric to map data into a new feature space.

Table 2 shows the results of all methods. Our **GFK(PLS, PCA)** is still the best, followed by **GFK(PCA, PCA)**. Note that though **GFK(PLS, PLS)** incorporates discriminative information from both domains, it does not perform as well as **GFK(PLS, PCA)**. This is probably due to the lack of enough labeled data in the target domains to give a reliable estimate of PLS subspaces. The **Metric** method does not perform well either, probably due to the same reason.

As in Table 1, for a given target domain, there is a "pal" source domain that improves the performance the most. Moreover, this pal is the same as the one in the setting of unsupervised domain adaptation. Thus, we believe that this "pal" relationship is intrinsic to datasets; in section 4.5, we will analyze them with our ROD metric.

4.4. Selecting the optimal dimensionality

Being able to choose the optimal dimensionality for the subspaces is an important property of our methods. Fig. 3

Table 3. Cross-dataset generalization with and without domain adaptation among domains with high and low "market values" [27]

%	No domain adaptation					Using domain adaptation					
→	P	I	C101	Mean Targets	Drop_1	P	I	C101	Mean Targets	Drop_2	Improvement
PASCAL	**37.9**	38.5	34.3	36.4	**4%**	–	43.6	39.8	41.7	**-10%**	**14%**
ImageNet	38.0	**47.9**	40.0	39.0	**19%**	42.9	–	49.1	46.0	**4%**	**18%**
Caltech101	31.9	38.6	**66.6**	35.3	**47%**	34.1	37.4	–	35.8	**46%**	**1%**

Table 4. ROD values between 4 domains. Lower values signify stronger adaptability of the corresponding source domain.

→	Caltech	Amazon	DSLR	Webcam
Caltech	0	**0.003**	0.21	0.09
Amazon	**0.003**	0	0.26	0.05
DSLR	0.21	0.26	0	**0.03**
Webcam	0.09	0.05	**0.03**	0

shows that the subspace disagreement measure (SDM) described in section 3.4 correlates well with recognition accuracies on the target domains. In the plots, the horizontal axis is the proposed dimensionality (in log scale) and the right vertical axis reports accuracies on both unsupervised domain adaptation and semi-supervised domain adaptation. The left vertical axis reports the values of SDM.

The plots reveal two conflicting forces at play. As the dimensionality increases, SDM—as a proxy to difference in geometric structures—quickly rises and eventually reaches its maximum value of 1. Beyond that point, adaptation becomes difficult as the subspaces have orthogonal directions.

However, before the maximum value is reached, the geometric difference is countered by the increase in variances — a small dimensionality would capture very little variances in the source domain data and would result in poor accuracies on both domains. The tradeoff occurs at where the geometric difference is just being maximized, justifying our dimensionality selection criterion in eq. (8).

4.5. Characterizing datasets with ROD

Given a target domain and several choices of datasets as source domains, identifying which one is the best to be adapted not only has practical utility but also provides new insights about how datasets are related to each other: ease of adaptation functions as a barometer, indicating whether two datasets are similar both geometrically and statistically, and piercing through each dataset's own idiosyncrasies.

To this end, we examine whether the Rank of Domain (ROD) metric described in section 3.5 correlates with our empirical findings in Table 1 and 2. We compute ROD using PCA subspaces and report the values among the 4 domains in Table 4. In general, ROD correlates well with recognition accuracies on the target domains and can reliably identify the best source domains to adapt. For example, when Caltech is the target domain (the first column), Amazon has the smallest value and Amazon indeed leads to better classification accuracies on Caltech than DSLR or Webcam.

If we group Caltech and Amazon into a meta-category "Online" and DSLR and Webcam into another meta-category "Office", the distributions of ROD values with respect to the categories suggest that the domains with the same meta-category have stronger similarity than domain pairs crossing categories (such as Caltech and DSLR). Thus ROD can also be used as a measure to partition datasets into clusters, where datasets in the same cluster share latent properties that might be of surprise to their users — the presence of such properties is probably not by design.

4.6. Easy to adapt: a new perspective on datasets?

Torralba and Efros study the sources of dataset bias in several popular ones for object recognition [27]. To quantify the quality of each dataset, they devise a "market value" metric. Datasets with higher values are more diverse, and therefore are likely to reflect better the richness of real-world objects. In particular, they point out that PASCAL VOC 2007 and ImageNet have high values.

Building on their findings, we turn the table around and investigate: *how valuable are these datasets in improving a target domain's performance?*

Table 3 summarizes our preliminary results on a subset of datasets used in [27]; PASCAL VOC 2007 [12], ImageNet [8], and Caltech-101 [13]. The recognition tasks are to recognize the category PERSON and CAR. The cross-dataset generalization results are shown on the left side of the table, without using adaptation techniques (as in [27]); and the adaptation results using our kernel-based method are on the right side of the table.

The rows are the source domain datasets and the columns are the target domains. The "Drop" columns report the percentages of drop in recognition accuracies between the source and the averaged accuracy on target domains, ie, the "Mean Targets" columns. The rightmost "Improvement" column is the percentage of improvement on target domains due to the use of domain adaptation. Clearly, domain adaptation noticeably improves recognition accuracies on the target domains. Caltech-101 is the exception where the improvement is marginal (47% vs. 46%). This corroborates the low "market value" assigned to this dataset in [27].

PASCAL VOC 2007 has the smallest drop without domain adaptation so it would appear to be a better dataset than the other two. Once we have applied domain adaptation, we observe a negative drop — ie, the performance on the target domains is better than on the source domain itself! However, its improvement is not as high as ImageNet's.

Our conjecture is that the data in PASCAL VOC 2007 can be partitioned into two parts: one part is especially "hard" to be adapted to other domains and the other part is relatively "easy". The reverse of the performance drop suggests that the "easy" portion can be harvested by domain adaptation techniques. However, the benefit is limited due to the "hard" part. On the other end, for ImageNet, a larger portion of its data is perhaps amenable to adaptation. Hence, it attains a bigger improvement after adaptation.

In short, while PASCAL VOC 2007 and ImageNet are assigned the same "market value" in [27], their usefulness to building object recognition systems that can be applied to other domains needs to be carefully examined in the context of adaptation. It might be beneficial to incorporate the notion of "ease of adaptability" in the process of evaluating datasets — a concept worth further exploring and refining.

5. Conclusion

We propose a kernel-based technique for domain adaptation. The techniquesembed datasets into Grassmann manifolds and constructing geodesic flows between them to model domain shift. The propose methods integrate an infinite number of subspaces to learn new feature representations that are robust to change in domains. On standard benchmark tasks of object recognition, our methods consistently outperform other competing algorithms.

For future work, we plan to exploit latent structures beyond linear subspaces for domain adaptation.

Acknowledgements

This work is partially supported by NSF IIS#1065243 and CSSG (B. G., Y. S. and F. S.), and ONR ATL (K. G.).

References

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

[2] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. In *Proc. of NIPS*, pages 137–144, 2007. 1, 2

[3] A. Bergamo and L. Torresani. Exploiting weakly-labeled web images to improve object classification: a domain adaptation approach. In *Proc. of NIPS*, pages 181–189, 2010. 2

[4] J. Blitzer, M. Dredze, and F. Pereira. Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification. In *Proc. of ACL*, pages 440–447, 2007. 1, 2

[5] J. Blitzer, R. McDonald, and F. Pereira. Domain adaptation with structural correspondence learning. In *Proc. of EMNLP*, pages 120–128, 2006. 1, 2

[6] L. Bruzzone and M. Marconcini. Domain adaptation problems: A DASVM classification technique and a circular validation strategy. *IEEE PAMI*, 32(5):770–787, 2010. 2

[7] H. Daumé III. Frustratingly easy domain adaptation. In *Proc. of ACL*, pages 256–263, 2007. 1, 2

[8] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proc. of CVPR*, pages 248–255, 2009. 7

[9] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: a benchmark. In *Proc. of CVPR*, pages 304–311, 2009. 1

[10] L. Duan, I. Tsang, D. Xu, and S. Maybank. Domain transfer SVM for video concept detection. In *Proc. of CVPR*, pages 1375–1381, 2009. 1

[11] L. Duan, D. Xu, I. Tsang, and J. Luo. Visual event recognition in videos by learning from web data. In *Proc. of CVPR*, pages 1959–1966, 2010. 1

[12] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007. 7

[13] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. *Comp. Vis. & Img. Under.*, 106(1):59–70, 2007. 7

[14] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *Proc. of ICCV*, pages 999–1006, 2011. 1, 2, 3, 5, 6

[15] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical report, Caltech, 2007. 5

[16] J. Hamm and D. Lee. Grassmann discriminant analysis: a unifying view on subspace-based learning. In *Proc. of ICML*, pages 376–383, 2008. 4

[17] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2009. 6

[18] J. Huang, A. Smola, A. Gretton, K. Borgwardt, and B. Scholkopf. Correcting sample selection bias by unlabeled data. In *Proc. of NIPS*, pages 601–608, 2006. 2

[19] V. Jain and E. Learned-Miller. Online domain adaptation of a pre-trained cascade of classifiers. In *Proc. of CVPR*, pages 577–584, 2011. 1

[20] B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *Proc. of CVPR*, pages 1785–1792, 2011. 2, 5

[21] C. Leggetter and P. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. *Computer Speech and Language*, 9(2):171–185, 1995. 1, 2

[22] S. Pan, I. Tsang, J. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. *IEEE Trans. Neural Nets.*, 99:1–12, 2009. 1, 2

[23] S. Pan and Q. Yang. A survey on transfer learning. *IEEE Trans. Knowl. & Data Eng.*, pages 1345–1359, 2009. 2

[24] F. Perronnin, J. Sánchez, and Y. Liu. Large-scale image categorization with explicit data embedding. In *Proc. of CVPR*, pages 2297–2304, 2010. 1

[25] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *Proc. of ECCV*, pages 213–226, 2010. 2, 5, 6

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

[27] A. Torralba and A. Efros. Unbiased look at dataset bias. In *Proc. of CVPR*, pages 1521–1528, 2011. 1, 2, 5, 7, 8

[28] M. Wang and X. Wang. Automatic adaptation of a generic pedestrian detector to a specific traffic scene. In *Proc. of CVPR*, pages 3401–3408, 2011. 1