

# Multivariate Regression on the Grassmannian for Predicting Novel Domains

Yongxin Yang, Timothy M. Hospedales  
Queen Mary, University of London

{yongxin.yang, t.hospedales}@qmul.ac.uk

## Abstract

*We study the problem of predicting how to recognise visual objects in novel domains with neither labelled nor unlabelled training data. Domain adaptation is now an established research area due to its value in ameliorating the issue of domain shift between train and test data. However, it is conventionally assumed that domains are discrete entities, and that at least unlabelled data is provided in testing domains. In this paper, we consider the case where domains are parametrised by a vector of continuous values (e.g., time, lighting or view angle). We aim to use such domain metadata to predict novel domains for recognition. This allows a recognition model to be pre-calibrated for a new domain in advance (e.g., future time or view angle) without waiting for data collection and re-training. We achieve this by posing the problem as one of multivariate regression on the Grassmannian, where we regress a domain’s subspace (point on the Grassmannian) against an independent vector of domain parameters. We derive two novel methodologies to achieve this challenging task: a direct kernel regression from  $\mathcal{R}^M \rightarrow \mathcal{G}$ , and an indirect method with better extrapolation properties. We evaluate our methods on two cross-domain visual recognition benchmarks, where they perform close to the upper bound of full data domain adaptation. This demonstrates that data is not necessary for domain adaptation if a domain can be parametrically described.*

## 1. Introduction

The issue of domain shift arises when the testing data on which we are interested to apply pattern recognition methods differs systematically from the training data available to train them – violating the underlying assumption of supervised learning. It is becoming increasingly clear that this issue is pervasive in practice and leads to serious drops in performance [35, 16]. This has motivated extensive work in the area of domain adaptation, which aims to ameliorate the negative impact of this shift by transforming the model or the data to bridge the train-test gap [32, 10, 28]. Although diverse, traditional domain adaptation (DA) approaches can

be grouped according to assumptions on supervision. Supervised approaches assume the target (test) domain has labels but the data volume is very small [32], while in unsupervised approaches the target domain is completely unlabelled [10]. Nevertheless, both of these categories share common assumptions of: (i) domains are discrete entities, e.g., corresponding to dataset [35], or capture device [32], and (ii) at least unlabelled data in the target domain.

Interest has recently grown in relaxing this strict assumption, and expanding the scope of domain adaptation to include a wider variety of practically valuable settings including: continuously varying rather than discrete domains [14]; domains parametrised by multiple factors rather than a single index [30, 38]; and predicting domains in advance of seeing any samples [18]. Continuous DA considers the situation where the domain evolves continuously, for example with time. In this case evolving the model online allows a recognition model to remain effective, e.g., at any time of day [14]. Multi-factor DA considers the situation where the domain is parametrised by a vector of multiple factors, e.g., [pose, illumination] [30] and [capture device, location] [38]. Here, the structured nature of the domain’s parameters can be used to improve performance compared to treating them as discrete entities. Finally, predictive DA considers *predicting a recognition model for a domain in advance of seeing any data*. This provides the powerful capability of pre-creating models suited for immediate use in novel domains, for example future data in a time-varying stream [18].

In this paper we provide a general framework for predictive domain adaptation – adapting a recognition model to a new setting for which some metadata (e.g., time or view angle) *but no data* is available in advance. This capability is important, because we may not be able to wait for extensive data collection, and re-training of models as would be required to apply conventional (un)supervised DA. Our framework takes as input a recognition model, and set of previously observed domains, each described by a parameter or parameter vector. It then builds a predictor for domains, that can be used to generate a recognition model for any novel domain purely based on its parameter(s). Our contribution is related to that of [38, 18], but it is signifi-

cantly more general because: (i) we can use as input a *vector* of any parameters, rather than a single time parameter only [18], (ii) we can use continuously varying rather than discrete parameters, which is important for domains defined by time or position [38], (iii) for continuous domains such as time, we can predict domains at an arbitrary point in the future rather than one time-step ahead as [18].

To provide this capability we frame the problem as one of multivariate regression on the Grassmannian. Points on the Grassmannian correspond to subspaces, and differing subspaces are a key cause of the domain shift problem – this is the key insight of subspace-based DA techniques that solve domain shift by aligning subspaces [10, 9, 5]. By regressing points on the Grassmannian against the independent parameters of each domain (such as time, view angle), we can predict the appropriate subspace for any novel domain in advance of seeing any data. Once the subspace is predicted, any existing subspace-based DA technique can be used to adapt a pre-trained model for application to the new domain. However, such regression on the Grassmannian is non-trivial to achieve. While methods have been proposed [3], they do not scale [17], or extend to multiple independent parameters [15]. We propose two different scalable approaches to multivariate regression on the Grassmannian: a direct kernel regression approach from  $\mathcal{R}^M \rightarrow \mathcal{G}$ , and an indirect approach with better extrapolation ability.

We compare our domain prediction approaches on the surveillance over time benchmark from [14], and on a car recognition task in the style of [14, 18] – but using both view and year as domain parameters. We demonstrate that we can extrapolate to predict domains multiple time-steps ahead in the future; and when applying our predictive domain adaptation framework to generate models for novel domains, performance approaches the upper bound of fully-observed DA, while not requiring any data or online re-training.

## 2. Related Work

### 2.1. Domain Adaptation

Domain Adaptation (DA) methods reduce the divergence between a source and target domain so that a model trained on the source performs well on the target. The area is now too big to review fully, but [28] provides a good survey.

**Unsupervised Domain Adaptation:** We focus here on unsupervised domain adaptation (UDA), as its lack of data annotation requirement makes it more generally applicable, and it is more closely connected to our contributions. In the absence of labels, UDA aims to exploit the marginal distributions  $P(X_{\mathcal{T}})$  and  $P(X_{\mathcal{S}})$  to align domains. There are two main approaches here: data and subspace centric.

**Data-centric Approaches:** These seek a transformation  $\phi(\cdot)$  that projects two domains’ data into a space that reduces the discrepancy between the transformed target

$\phi(X_{\mathcal{T}})$  and source data  $\phi(X_{\mathcal{S}})$ . A typical pipeline is to perform PCA [24] or sparse coding [23] on the union of the domains with an additional objective that minimises maximum mean discrepancy (MMD) of the new representations a reproducing kernel Hilbert space  $\mathcal{H}$ , i.e.,  $\|\mathbb{E}[\phi(X_{\mathcal{T}})] - \mathbb{E}[\phi(X_{\mathcal{S}})]\|_{\mathcal{H}}^2$ . Domain generalisation (DG) approaches [25] find transformations that minimise distance between an arbitrary number of domains, with the aim of generalising to new domains. Thus DG is appropriate if no domain parameters are available, while our predictive DA is expected to outperform DG if metadata is available.

**Subspace based Approaches:** These approaches make use of the two domain’s subspaces rather than manipulating the data directly. Here, subspace refers to a  $D$ -by- $K$  matrix of the first  $K$  eigenvectors of the original  $D$ -dimensional data. We denote  $P_{\mathcal{S}}$  and  $P_{\mathcal{T}}$  as the source and target domain subspaces learned separately by PCA. Subspace Alignment (SA) [5] learns a linear map  $M$  for  $P_{\mathcal{S}}$  that minimise the Bregman matrix divergence  $\|P_{\mathcal{S}}M - P_{\mathcal{T}}\|_F^2$ . [10] samples several intermediate subspaces  $P_1, P_2, \dots, P_N$  from  $P_{\mathcal{S}}$  to  $P_{\mathcal{T}}$ . That is achieved by thinking of  $P_{\mathcal{S}}$  and  $P_{\mathcal{T}}$  as points on the Grassmann manifold  $\mathcal{G}(K, D)$  and finding a geodesic (shortest path on manifold) between them. Points (subspaces) are sampled from the geodesic and concatenated to form a richer linear operator  $[P_{\mathcal{S}}, P_1, P_2, \dots, P_N, P_{\mathcal{T}}]$  that projects two domains into a common space, where the source classifier generalises better to the target domain. A weakness of [10] is that the number of intermediate points is a hard-to-determine hyper-parameter. An elegant solution to this, [9] samples all the intermediate points. This produces infinitely long feature vectors, but their dot-product is defined, and thus any kernelised classifier can be used.

**New Settings:** All the previously discussed methods apply in the classic DA setting of a discrete source and target domain. The setting where domains are continuously evolving was recently considered by [14] using a sequential PCA and subspace-based DA method. This is important for many practical problems, however the proposed method has limitations: It does not extend to a vector of domain parameters, and most importantly it can not be used to predict future unseen domains. Predicting future domains was recently considered by [18]. The proposed data-centric approach re-weights the past training samples via making a prediction of their time-varying probability distribution. However, this again does not extend to more than one domain parameter, and predictions can only be made *one time-step into the future*. Both [14, 18] are constrained to fixed-size time-steps. Some previous studies considered vector-parametrised domains [30, 38]. [30] is limited in application to multi-view recognition type problems where the same instance (e.g., face) is seen in each domain, thus it does not extend to general recognition tasks. [38] is restricted to discrete domain parameters as it uses 1-of-K coding that applies to categor-

Table 1. Contrasting our direct (D) and indirect (ID) methods versus existing predictive/non-predictive domain adaptation studies.

Capability	D	ID	[14]	[18]	[38]
Predictive	✓	✓	✗	✓	✓
Multi-factor	✓	✓	✗	✗	✓
Continuous Parameters	✓	✓	✓	✓	✗
Extrapolation	✗	✓	✗	✓	✗
Extrapolate arbitrarily far	✗	✓	✗	✗	✗

ical parameters only. In this paper, we generalise all these settings by developing a model to predict a new subspace  $P \in \mathcal{G}$  given its corresponding parameter  $\mathbf{z} \in \mathcal{R}^M$ . Since  $\mathbf{z}$  is a vector, multiple factors can be used, and extrapolations made arbitrarily far in the future. Tab 1 contrasts our contribution with prior work.

## 2.2. Zero-Shot Learning

A related problem to predictive domain adaptation is Zero-Shot Learning (ZSL). In ZSL, a classifier is created for a novel task, in the absence of any training data for the task, by using task metadata. This has been extensively studied in applications such as character [20], object [19, 7], and action [22] recognition. Instead of building a map (classifier) directly from the image to label space, ZSL studies [20, 27] learn the classifier in terms of an intermediate semantic representation such as attributes [19, 22] or word-vectors [7]. Recognisers can then be built on-the-fly for novel objects given only their semantic representations. For example by assigning the attributes [‘black’, ‘white’, ‘stripes’] to the new object ‘zebra’. We aim to achieve a similar on-the-fly capability for DA: adapting a trained model to any target domain given only its (often freely available) metadata. Such metadata is termed herein as *domain parameter*.

## 2.3. Manifold-valued Data Regression

Several studies address regression in the setting that the independent variable is a point in Euclidean space and the dependent variable is a point in non-flat space such as Riemann or Grassmann manifolds. These can be grouped into: (i) Parametric approaches like [15, 6, 31, 13, 8] typically try to find a formulation for the geodesic and then provide a numerical solution for its estimation and (ii) Non-parametric approaches such as [3, 4] adapt kernel regression to the manifold case by observing that they are all essentially about searching for a point for which the sum of its (reweighted) distances with all training points is minimised.

For most parametric solutions, the independent variable is assumed to be scalar. This is because: (i) In applications where these methods are popular, e.g., medical imaging, one usually wants to find a pattern against a single factor (e.g., age) and (ii) it is technically challenging to extend these to the multivariate case [17], because the prediction no longer corresponds to a single geodesic curve, which makes the gradient derivation problematic.

The non-parametric method [3] provides the solution for a very special manifold 3D rotation group  $SO(3)$ , so it is not applicable for our problem. However, it inspires us to extend kernel regression, since the kernel function does not make assumptions on whether the input is a scalar or vector. This then forms the core of our direct prediction method.

## 3. Methodology

In order to build predictive model for domains (subspaces), we need to build a mapping from an  $M$ -dimensional vector of independent variables to points on the Grassmannian (represented by a matrix with orthonormal columns):  $\mathcal{R}^M \rightarrow \mathcal{G}$ . The output constraint means it can not be treated as conventional Euclidean regression. In the following we present direct kernel regression (Sec 3.1) and an indirect solution (Sec 3.2) to achieve this.

### 3.1. Direct Kernel Regression on the Grassmannian

**Kernel Regression Review** We first review kernel regression. Assume we are given a set of (data, label) pairs,

$$\{(\mathbf{z}_1, P_1), (\mathbf{z}_2, P_2), \dots, (\mathbf{z}_N, P_N)\} \quad (1)$$

where  $\mathbf{z} \in \mathcal{R}^M$  and  $P \in \mathcal{R}^1$ ; and a kernel function  $k(\mathbf{z}_1, \mathbf{z}_2)$  that measures the similarity between  $\mathbf{z}_1$  and  $\mathbf{z}_2$ . The Nadaraya-Watson [26] kernel regression prediction for  $P$  given a test point  $\mathbf{z}$  is

$$P = \frac{\sum_{i=1}^N k(\mathbf{z}, \mathbf{z}_i) P_i}{\sum_{i=1}^N k(\mathbf{z}, \mathbf{z}_i)} \quad (2)$$

**From Euclidean to Grassmannian** When  $P \in \mathcal{M}$  where  $\mathcal{M}$  is a non-flat manifold and  $P$  is no longer a scalar, Eq. 2 can be invalid. So this does not provide a solution to our problem. For example, suppose  $\mathcal{M}$  is a Grassmann manifold  $\mathcal{G}(K, D)$ , so its numerical representation<sup>1</sup> is now matrices  $P \in \mathcal{R}^{D \times K}$  with constraints  $P^T P = I_K$ . Eq. 2 could be applied, but this is meaningless because adding two points on the Grassmannian does not necessarily give another point on the Grassmannian.

Inspired by [3], we propose to think of kernel regression as the solution of the following optimisation problem:

$$\arg \min_{P \in \mathcal{R}^1} \sum_{i=1}^N w_i (P - P_i)^2 \quad (3)$$

where  $w_i = \frac{k(\mathbf{z}, \mathbf{z}_i)}{\sum_{i=1}^N k(\mathbf{z}, \mathbf{z}_i)}$ . More generally, we have

<sup>1</sup>Strictly speaking, it is inaccurate to say a matrix with orthonormal columns is a point on the Grassmann manifold, though many papers use this terminology [10, 9]. The correct manifold to mention is Stiefel manifold, but this does not affect the correctness of these subspace-based DA methods because such a matrix is one of the *non-unique* numerical representations of a point on the Grassmann manifold.

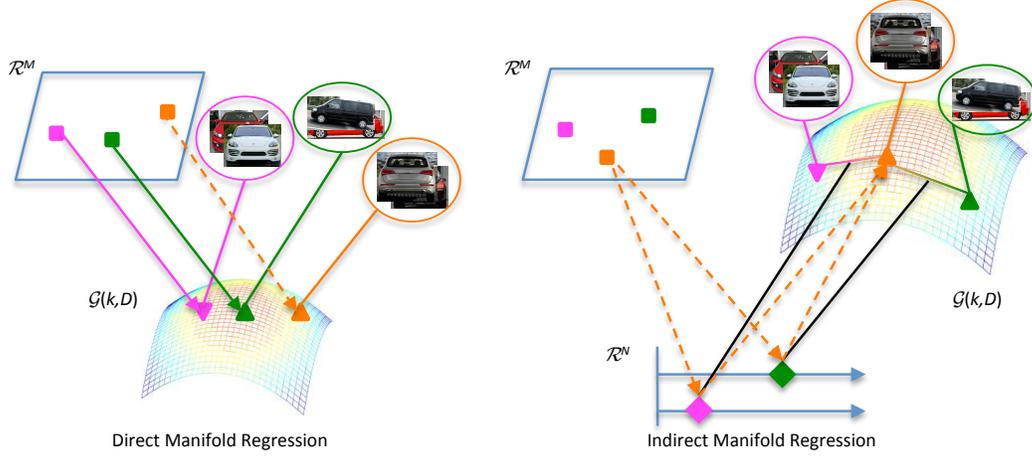


Figure 1. Illustration of the proposed methods. Direct kernel regression: Domain parameters to subspace:  $\mathcal{R}^M \rightarrow \mathcal{G}$ . Indirect regression: Domain parameters to distances of a reference set to subspace:  $\mathcal{R}^M \rightarrow \mathcal{R}^N, \mathcal{R}^N \rightarrow \mathcal{G}$ .

$$\arg \min_P \sum_{i=1}^N w_i d^2(P, P_i) \quad (4)$$

where  $d^2(\cdot, \cdot)$  is a metric (distance function).  $P$  is the Fréchet mean if the minimizer is unique (or Karcher mean when it is a local minimum). The Fréchet mean is defined in general metric space, thus it provides a way to work with manifold-valued data as long as we can find a well defined distance function for points on the manifold.

**Grassmann Manifold Background** We first review some concepts about the Grassmannian, before solving Eq 4. Many distances on the Grassmannian are defined based on the concept of ‘principal angle’, which can be calculated by SVD. E.g., for two points  $P_1$  and  $P_2$  on  $\mathcal{G}(K, D)$ ,

$$P_1^T P_2 = USV^T \quad (5)$$

where  $S = \text{diag}(\cos(\theta_1), \cos(\theta_2), \dots, \cos(\theta_K))$ . The angle  $\theta_k = \cos^{-1}(S_{k,k})$  is the  $k$ th principal angle. Multiple distance functions exist on the Grassmannian (Table 2). [12, 39] provide a variety of metrics and their derivations.

Table 2. Distances  $d^2(P_1, P_2)$  on  $\mathcal{G}(K, D)$  in terms of principal angles and orthonormal bases

	Principal angles	Orthonormal bases
Binet–Cauchy	$1 - \prod_{k=1}^K \cos^2 \theta_k$	$1 - (\det(P_1^T P_2))^2$
Martin	$\log \prod_{k=1}^K (\cos^2 \theta_k)^{-1}$	$-\log((\det(P_1^T P_2))^2)$

### Manifold-valued Data Regression with Vector Input

For our manifold-valued data regression task, Binet–Cauchy (BC) and Martin distances are suitable because they are amenable to deriving gradients w.r.t. the target matrix,

and their sensitivity properties are more favourable than alternatives [12]. However, when the core part of BC distance, i.e., the determinant, is so small that underflow occurs, Martin distance is a better choice as it calculates the log-determinant. Substituting Martin distance into Eq. 4, we obtain the following objective function to optimise:

$$\arg \min_{P \in \mathcal{R}^{D \times K}} - \sum_{i=1}^N w_i \log((\det(P^T P_i))^2) \quad (6)$$

which is subject to constraint  $P^T P = I_K$ . The gradient with respect to  $P$  is:

$$\nabla_P = -2 \sum_{i=1}^N w_i P_i (P^T P_i)^{-1}. \quad (7)$$

Vanilla gradient descent is not applicable because of the orthogonality constraints. It is a non-trivial optimisation problem as the constraints lead to non-convexity. A simple solution is to do gradient descent and re-orthogonalise the matrix after each step, but it is numerically expensive. Some studies have addressed this issue, e.g., [34, 29, 33]. We apply the solution from [36]: an efficient update scheme based on the Cayley transformation that preserves the constraints.

Given a feasible point  $P$  and the gradient  $G = \nabla_P$ , a skew-symmetric matrix  $A$  is defined as,

$$A := GP^T - PG^T \quad (8)$$

The new trial point is determined by the Crank-Nicolson-like scheme,

$$P^{\text{Update}}(\eta) = P - \frac{\eta}{2} A(P + AP^{\text{Update}}(\eta)) \quad (9)$$

where  $\eta$  is the step size that can be found by curvilinear search, and  $P^{\text{Update}}(\eta)$  is given by the closed form,

$$P^{\text{Update}}(\eta) = QP \text{ where } Q = (I + \frac{\eta}{2}A)^{-1}(I - \frac{\eta}{2}A) \quad (10)$$

Iterating Eq. 10 is guaranteed to converge to a stationary point as a solution for Eq. 6. Thus we can predict an unseen domain’s subspace  $P$  given its domain parameter  $\mathbf{z}$ .

### 3.2. Indirect Regression on the Grassmannian

A limitation of the previous direct approach is that, similarly to Euclidean kernel regression, it is fundamentally *interpolation*-based. This makes it unable to extrapolate to out-of-sample (e.g., far future) subspaces. To address this, we propose an indirect approach for  $\mathcal{R}^M \rightarrow \mathcal{G}$  regression.

**Indirect Prediction with a Single Reference:** Instead of regressing domain parameter  $\mathbf{z} \in \mathcal{R}^M$  to subspace  $P \in \mathcal{G}$ , consider setting the dependent variable of our regression problem to be the *distance*  $l \in \mathcal{R}^1$  between  $P$  and a fixed reference point (subspace). Then the problem is reduced to standard multivariate regression  $\mathcal{R}^M \rightarrow \mathcal{R}^1$ .

A natural question arises: how to choose the reference point on the Grassmannian? A simple answer is to use the Karcher mean of all observed points (subspaces). Assume that the Karcher mean is  $\bar{P}$ , then the  $N$  training instances/labels for the regression model are  $\{(\mathbf{z}_1, d^2(P_1, \bar{P})), (\mathbf{z}_2, d^2(P_2, \bar{P})), \dots, (\mathbf{z}_N, d^2(P_N, \bar{P}))\}$ . For a given testing instance  $\mathbf{z}$ , the regression model predicts the *distance* between its associated subspace and the reference subspace:  $\hat{l}$ . We could then estimate the target subspace by solving the following optimisation problem.

$$\arg \min_{P \in \mathcal{R}^{D \times K}} (\hat{l} - d^2(P, \bar{P}))^2 \quad (11)$$

However, Eq. 11 is underdetermined, so does not guarantee a meaningful result.

**Indirect Prediction with Multiple References:** To find a unique optimum, rather than use a fixed reference point, we instead use all observed subspaces  $P$  as references. This results in a multi-output regression problem  $\mathcal{R}^M \rightarrow \mathcal{R}^N$ . For a given test instance  $\mathbf{z}$ , it will yield a  $N$ -dimensional vector  $\hat{\mathbf{l}} = [\hat{l}_1, \hat{l}_2, \dots, \hat{l}_N]$ , where  $\hat{l}_i$  is the estimated distance between the target subspace and the  $i$ th observed subspace. Thus rather than the direct regression  $\mathbf{z} \rightarrow P$  we now have  $\mathbf{z} \rightarrow \hat{\mathbf{l}} \rightarrow P$ . The objective for the second step is then

$$\arg \min_{P \in \mathcal{R}^{D \times K}} \sum_{i=1}^N (\hat{l}_i - d^2(P, P_i))^2 \quad (12)$$

Eq. 12 can be solved by the constrained gradient descent from Sec. 3.1, where the gradient with respect to  $P$  is:

$$\nabla_P = \sum_{i=1}^N 4(\hat{l}_i + \log((\det(P^T P_i))^2)) P_i (P^T P_i)^{-1} \quad (13)$$

### 3.3. Predictive Domain Adaptation

Using the methodology developed in Sec. 3.1-3.2, our goal of predictive domain adaptation becomes possible. We assume that we are given: (i) a classifier trained on any source domain, and (ii)  $N$  additional unlabelled domains:

$$\{(X_1, \mathbf{z}_1), (X_2, \mathbf{z}_2), \dots, (X_N, \mathbf{z}_N)\}, \quad (14)$$

where  $X_i$  is  $i$ th domain data, from which we can learn a subspace  $P_i$  by PCA; and  $\mathbf{z}_i$  is  $i$ th domain’s parameters.

For an unseen domain with parameters  $\mathbf{z}_*$ , we can predict its subspace  $P_*$  based on the proposed method (Eq. 6 or Eq. 12) and the training data  $\{(\mathbf{z}_1, P_1), (\mathbf{z}_2, P_2), \dots, (\mathbf{z}_N, P_N)\}$ . Once  $P_*$  is obtained, any subspace-based DA method (e.g., [5, 10, 9]) can be applied to align the unseen (target) domain  $P_*$  to any labelled source domain  $P_S$  where a classifier was trained. An illustration of our approaches is given in Fig. 1.

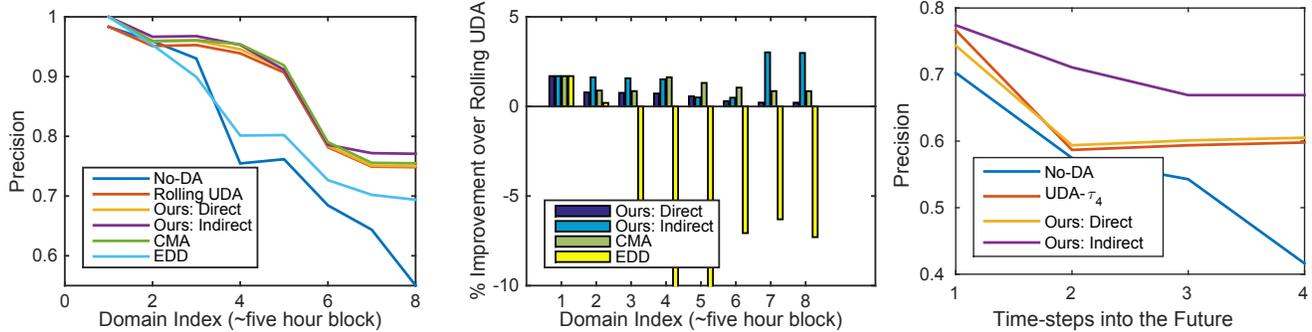
## 4. Experiments

We evaluate our contributions on two benchmark tasks: (i) surveillance scene classification, which is well suited to testing extrapolation and (ii) vehicle type classification over time and viewing angle, suited for testing multi-factor based prediction. For the direct method, RBF kernel is used for measuring the domain parameter similarity, and its bandwidth is set to be the median of pairwise distances of the domain parameters [11]. For the indirect method, the regression from domain parameter to reference set distances is done by kernel ridge regression with same RBF kernel above, and  $\ell_2$  regularisation weight chosen by cross validation. Given our methods’ predicted subspaces, we need to plug in a subspace-based DA method to complete the adaptation. We choose to use Geodesic Flow Kernel (GFK) [9] because of its better performance and fewer hyper-parameters to tune. GFK takes in source and target domain subspaces and produces a positive semi-definite matrix  $G$ , with which we can calculate the training and testing kernels by  $x_i^T G x_j$ . The precomputed kernel is then fed into LIB-SVM [1] for classifier training, for which the cost parameter is tuned by 10-fold cross validation.

### 4.1. Surveillance Scene Classification

**Dataset:** We use the benchmark dataset studied by [14]. It contains video frames (320x240 RGB, 20 per hour) of a traffic intersection captured by a fixed camera over an extended period of time. Over time, factors such as illumination change cause a challenging domain shift problem.

**Setup:** We use the 512-dimensional GIST feature provided by [14]. For direct comparison, we also adopt their task: recognising if one or more vehicles appear in the frame. For this binary classification, model performance is measured by Precision =  $\frac{\text{true positive}}{\text{true positive} + \text{false positive}}$ .



(a) Car detection performance with DA (CMA [14]) and Predictive-DA (Direct, Indirect, EDD [18]). (b) Far future extrapolation of car detection.

Figure 2. Surveillance experiment results

The source domain where the classifier is trained is the same as [14]: it contains 50 consecutive images (2.5 hours). Then we test on the immediately following 40 hours (800 images). The testing frames are split into eight domains equally in time, so each target domain has 100 images. For simplicity, we denote the source domain as  $\tau_0$ , and the  $i$ th target domain as  $\tau_i$ . Six methods are evaluated:

**No-DA:** A lower bound. The classifier trained on  $\tau_0$  is directly applied to every  $\tau_i$ .

**Rolling UDA Baseline:** Approximately adapt to  $\tau_i$  by using the subspace of its previous domain  $\tau_{i-1}$  as input to GFK. Note that this is only reasonable in such a continuous online application, not for arbitrary subspace prediction.

**Direct:** Our direct method (Sec 3.1) predicts  $\tau_i$ 's subspace  $\hat{P}_i$  given the observed subspaces from  $\{\tau_0, \tau_1, \dots, \tau_{i-1}\}$ .

**Indirect:** As above, but using the proposed indirect method (Sec 3.2) to predict the subspace of  $\tau_i$ .

**EDD [18]:** Predictive DA by modelling the data's time-varying distribution. The paper [18] assumed all domains are labelled because it essentially re-weights the classification loss on the level of individual domain. So we provide labels of all domains, giving it a significant advantage

**CMA [14]:** Rather than *predicting* the subspace from its timestamp, CMA assumes we have observed data in  $\tau_i$  from which the true subspace  $P_i$  is learned, and then fed into GFK for domain adaptation. By using the true rather than predicted subspace, this provides an upper bound.

Instead of learning the subspace of  $\tau_i$  from scratch every time, we use sequential PCA [21] that processes the data in  $\tau_i$  and its previous subspace  $P_{i-1}$ . The motivation is twofold: (i) it encourages subspace smoothing that fits the nature of this task well and (ii) by doing so CMA reproduces the result of [14]. The number of eigenvectors used for  $P_i$  is 10 (same as [14]). The domain parameter is intuitive: integers starting from 0 (i.e., the index of  $\tau$ ).

**Domain Prediction Results:** Fig. 2(a) summarises the performance of the methods as time proceeds. The results are cumulative so, for example, the 5th result is the precision evaluated on the first 5 target domains. From the raw

results on the left, we can see that: (i) Without DA, performance drops significantly as time passes and domain shift increases and (ii) All alternatives are better than the baseline of No-DA. For easier comparison, Fig. 2(a) also presents the results as % improvement over the simple baseline of Rolling UDA. From here we can see that: (i) In each case our methods improve on Rolling UDA, indicating successful prediction of future subspaces. (ii) The performance of EDD is considerably worse than predicted/actual subspace-based DA methods, though it surpassed the baseline of No-DA. The reason might be that its adaptation strategy (re-weight the loss of each domain) is more crude than GFK. (iii) Our direct method is comparable to the CMA upper bound in early predictions. (iv) Our indirect method surpasses the direct method, and occasionally the CMA upper bound, particularly as the domain grows more distant ( $\tau_7$  and  $\tau_8$ ) from the initial source  $\tau_0$ .

**Extrapolating Far Future Domains:** The previous experiment predicted domains *one* time-step ahead. This strong assumption enabled Rolling UDA as competitive baseline because with only one time-step change, the domain shift was always small. To test the ability to extrapolate and predict domains arbitrarily far ahead in time, the next experiment fixed a set of observed domains  $\{\tau_0, \tau_1, \dots, \tau_4\}$ . We then tested on the following four domains without seeing any data (given their parameters only). The proposed methods are compared with two baselines: No-DA (classifier trained on  $\tau_0$  is directly used) and UDA- $\tau_4$  (GFK using the last available domain  $\tau_4$ 's subspace in place of subspaces of unseen test domains). The results in Fig. 2(b) show that when required to extrapolate further into the future, our indirect method is clearly superior to our direct method and other baselines.

**Summary:** Overall, both our proposed methods for predicting domains from metadata (in this case future timestamp), perform comparably or better than CMA-based DA, which requires the much stronger assumption that data in these domains was available for training models. Moreover, we are not restricted to small domain-shifts: our indirect

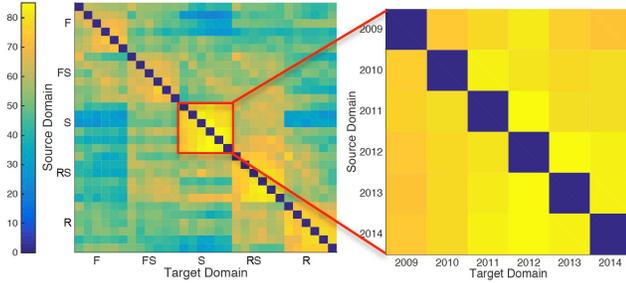


Figure 3. Demonstration of domain shift: Accuracies of every source-target combination without DA. Each off-diagonal cell corresponds to one of 870 experiments. Rows indicate domain for classifier training, and columns correspond to the target domain.

method can effectively predict domains that are distant from available training domains. These capabilities enable use cases where we can calibrate (domain adapt) a model and deploy it for immediate use in a new context without waiting to collect data and retrain the model.

## 4.2. Vehicle Type Classification

**Dataset:** Comprehensive Cars [37] is a recent large-scale dataset of vehicle photos. We use the first subset (Part-I in [37]) in our second experiment. This subset contains 30,955 images of entire cars, in which there are 431 unique models. The manufacturing year ranges from 2004 to 2015, and each image is associated with one of five view points.

**Setup:** We use the state-of-the-art CNN model VGG-full [2] as feature extractor. The image is preprocessed by cropping the given bounding box, rescaling to  $224 \times 224$  and subtracting the mean. Then it is fed into the CNN, where values in the penultimate layer (4096 neurons) is used as the feature vector for further experiments. We define a multi-class recognition task: to determine if the vehicle is an *MPV*, *SUV*, *sedan*, or *hatchback*. For the domain parameters, we use a two-dimensional vector [Year, Viewpoint]. We encode years 2009-2014 as integers 1-6, and viewpoint (Front, Front-Side, Side, Rear-Side, Rear) as integers 1-5. We exclude years 2004-08 and 2015 because there are too few examples, producing extremely small domains.

The two domain factors are orthogonal, so they produce  $6 \times 5 = 30$  domains in total. The number of images per domain ranges from 347 to 1410, with 24151 in total – about 80% of the dataset. To evaluate Predictive-DA, we use a hold-one-domain-out strategy. In each round, we observe all domains’ data (not necessarily labels) except the held-out target domain. For this target domain, we only know the domain parameters, but use the proposed methods to estimate its subspace. Subsequently we perform a standard UDA analysis: from one of the  $30 - 1 = 29$  source domains, we train a classifier with or without the help of the predicted subspace and domain adaptation (via GFK).

For parsimony, we exhaustively consider each possible

source and target domain, resulting in a total of  $30 \times 29 = 870$  experiments. In each round of the experiment, we report Accuracy of four methods: (i) **No-DA** No domain adaptation lower bound, (ii) **Direct** subspace prediction (iii) **Indirect** subspace prediction and (iv) **GT-DA** Upper bound of DA with ground-truth subspace. No-DA versus GT-DA is the typical comparison made by studies of unsupervised domain adaptation. Meanwhile our Direct and Indirect methods address the new predictive domain adaptation setting, and will perform somewhere between the lower and upper bounds of No-DA and GT-DA respectively.

**Domain Shift Analysis:** We first verify the existence of the domain shift problem. Fig. 3 shows the performance of all 870 experiments without domain adaptation. We can see that performance is better closer to the diagonal. This is because the domains are ordered by view and by year. We can also find five ‘block’ patterns along the diagonal, that correspond to the five different viewpoints: The performance is better if the target domain’s viewpoint is the same as the training domain viewpoint. Zooming in on one view block, e.g., ([2009-2014, Side]), we see that performance drops as the gap between source and target year grows (Fig. 3, inset). These visualisations verify the existence of two independent sources of domain shift, and thus support the value of our multi-variate regression contribution.

Table 3. Vehicle Classification Results. Accuracy average over all 870 source-target domain combinations.

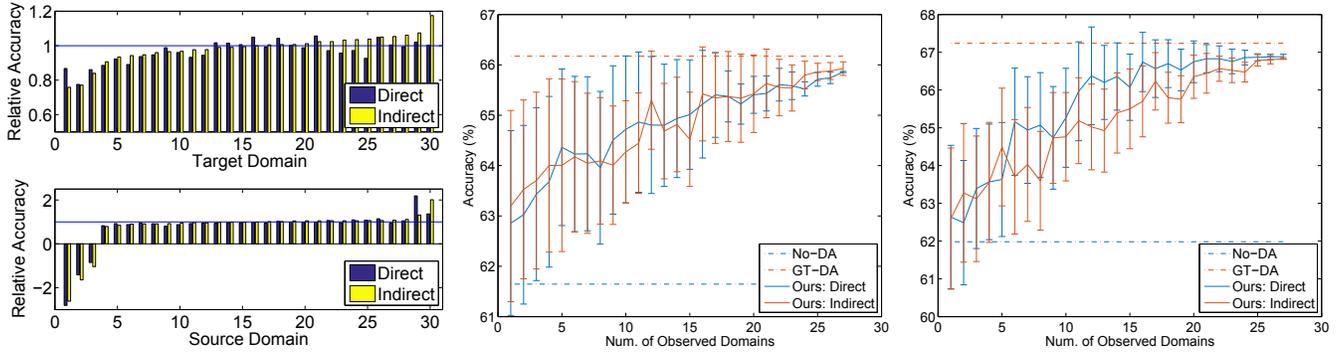
No-DA	Direct	Indirect	GT-DA
$54.06 \pm 12.09$	$58.12 \pm 9.69$	$58.21 \pm 9.56$	$58.23 \pm 9.68$

**Predictive DA Analysis:** We next investigate whether the proposed methods can predict the target subspace in order to alleviate the domain-shift problem visualised in Fig. 3. Table 3 reports the mean and standard deviation of the accuracies calculated from 870 experiments. From this we can observe that: (i) The DA methods surpass the No-DA baseline and (ii) Our Direct and Indirect methods match the upper bound of fully observed DA in performance and stability. It strongly suggests that our proposed methods can synthesise useful subspaces for use with UDA methods.

**Further Analysis:** To analyse the previous results in more detail, we plot the accuracy broken down over target or source domain (Fig. 4(a)). Since the number of domains is large, we rescale the accuracies as:

$$\text{acc} = \frac{\text{acc.} - \text{acc. of No-DA}}{|\text{acc. of DA-with-true-subspace} - \text{acc. of No-DA}|}$$

To interpret the results, divide the rescaled figures into three intervals:  $[1, +\infty]$  means the proposed method outperforms DA with ground-truth subspace;  $[0, 1]$  means the proposed method makes positive contribution, but less than DA with ground-truth subspace;  $[-\infty, 0]$  means negative transfer happens – using DA is worse than not using it.



(a) Performance over each target (top) and source (below) domains. Domains sorted by accuracy. (b) Examples of dependence of accuracy on number of observed domains to train Grassmann regression. Left: 2010 Front-side→2013 Rear-side Right: 2011 Side→2011 Rear-side

Figure 4. Evaluating multivariate domain prediction on comprehensive cars database

Fig. 4(a,top) shows that for all target domains, the proposed methods surpass the baseline of not using DA, and sometimes even surpass DA-with-ground-truth-subspace. Similarly, Fig. 4(a,bottom) shows that most source domains provide positive transfer on average. However, three source domains produce ‘hard-to-transfer’ classification models.

**Negative transfer analysis:** The reason behind negative transfer in Fig. 4(a,bottom) could be: (i) Two domains have fewer examples (438, 621) than average (805), (ii) The third domain has enough examples (1082), but its label set is imbalanced by a very low number of MPV cars. However, note that for these source domains, negative transfer is also observed when DA-with-ground-truth-subspace is used as well, so these are examples of downstream UDA failure, rather than failures of our subspace prediction.

For reference, in 870 experiments, DA (with ground-truth subspace) improved 711 (81.72%). Of these 711 experiments, Direct improves 687 (of its 700 positive transfer cases in total) and Indirect improves 678 (of its total 690 positive transfer cases). Thus overall, as long as DA can alleviate the domain shift, it is likely that our predictive DA can as well – but without accessing the domain’s data.

**Effect of number of observed domains:** The experiments so far have considered a hold-one-out setting, with all domains except the target being observed. In practice, we may not have such a such dense collection of known domains, so we investigate the relation between number of training domains and performance. We select two representative examples from the 870 available source→target pairs: 2010/Front-Side→2013/Rear-Side and 2011/Side→2011/Rear-Side. Then we sample an increasing number (1 to 27) of domains (together with source domain’s subspace) for training our models. We run each experiment 20 times randomly choosing the sources each time. Fig. 4(b) illustrates the results. Generally more observed domains leads to better performance. However this dependence is weak once a few domains have been ob-

served: The accuracy is close to the upper bound once the number of observed domains is larger than 15. This reassuringly suggests that dense observations of prior domains is not critical for the efficacy of predictive-DA.

## 5. Conclusion

We proposed the problem of vector-parametrised predictive domain adaptation and developed two solutions based on manifold-valued data regression. This allows us to predict a test-time subspace and thus align a source classifier to a test-domain *in advance of seeing any data*. Results on two benchmarks demonstrate that our approach matches, and sometimes surpasses the upper bound of using the true test-time subspace. This could impact a variety of areas where it would be useful to be able to pre-calibrate a model, or calibrate it on the fly based on sensor metadata.

There are numerous areas for future work. We studied vector domain parameters, but kernel regression could apply to any domain parameter where kernels exist (e.g., trees, strings). While the proposed method uses a set of available source domains’ data to learn the subspace regressor, it can only exploit a single source domain’s labels. A useful extension would be to exploit multiple source domains’ labels. It is also interesting to see if the predicted subspace can act as a regulariser that still helps when target data are available but limited. We assume (in common with most other DA work) that the domain parameter is observed and accurate. Relaxing this assumption to deal with missing or noisy parameters is also an interesting direction. Finally, although our application was predictive-DA, our methods for regression on the Grassmannian are general contributions that could be used in other areas such as medical imaging.

**Acknowledgements** This work was supported by EPSRC (EP/L023385/1), and the European Union’s Horizon 2020 research and innovation program under grant agreement No 640891.

## References

- [1] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2011.
- [2] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference (BMVC)*, 2014.
- [3] B. C. Davis, P. T. Fletcher, E. Bullitt, and S. Joshi. Population shape regression from random design data. In *International Conference on Computer Vision (ICCV)*, 2007.
- [4] B. C. Davis and S. Lazebnik. Analysis of human attractiveness using manifold kernel regression. In *International Conference on Image Processing (ICIP)*, 2008.
- [5] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *International Conference on Computer Vision (ICCV)*, 2013.
- [6] P. T. Fletcher. Geodesic regression and the theory of least squares on riemannian manifolds. *International Journal of Computer Vision (IJCV)*, 105:171–185, 2013.
- [7] Y. Fu, T. Hospedales, T. Xiang, Z. Fu, and S. Gong. Transductive multi-view embedding for zero-shot recognition and annotation. In *European Conference on Computer Vision (ECCV)*, 2014.
- [8] K. Gallivan, A. Srivastava, X. Liu, and P. Van Dooren. Efficient algorithms for inferences on grassmann manifolds. In *IEEE Workshop on Statistical Signal Processing*, 2003.
- [9] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [10] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *International Conference on Computer Vision (ICCV)*, 2011.
- [11] A. Gretten, O. Bousquet, A. Smola, and B. Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *Proceedings of the 16th international conference on Algorithmic Learning Theory*, 2005.
- [12] J. Ham and D. D. Lee. Grassmann discriminant analysis: a unifying view on subspace-based learning. In *International Conference on Machine Learning (ICML)*, 2008.
- [13] J. Hinkle, P. Muralidharan, P. T. Fletcher, and S. C. Joshi. Polynomial regression on riemannian manifolds. In *European Conference on Computer Vision (ECCV)*, 2012.
- [14] J. Hoffman, T. Darrell, and K. Saenko. Continuous manifold based adaptation for evolving visual domains. In *Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [15] Y. Hong, R. Kwitt, N. Singh, B. Davis, N. Vasconcelos, and M. Niethammer. Geodesic regression on the grassmannian. In *European Conference on Computer Vision (ECCV)*, 2014.
- [16] A. Khosla, T. Zhou, T. Malisiewicz, A. Efros, and A. Torralba. Undoing the damage of dataset bias. In *European Conference on Computer Vision (ECCV)*, 2012.
- [17] H. J. Kim, B. B. Bendlin, N. Adluru, M. D. Collins, M. K. Chung, S. C. Johnson, R. J. Davidson, and V. Singh. Multivariate general linear models (MGLM) on riemannian manifolds with applications to statistical analysis of diffusion weighted images. In *Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [18] C. H. Lampert. Predicting the future behavior of a time-varying probability distribution. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [19] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *Computer Vision and Pattern Recognition (CVPR)*, 2009.
- [20] H. Larochelle, D. Erhan, and Y. Bengio. Zero-data learning of new tasks. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2008.
- [21] A. Levey and M. Lindenbaum. Sequential karhunen-loeve basis extraction and its application to images. *Image Processing, IEEE Transactions on*, 9(8):1371–1374, 2000.
- [22] J. Liu, B. Kuipers, and S. Savarese. Recognizing human actions by attributes. In *Computer Vision and Pattern Recognition (CVPR)*, 2011.
- [23] M. Long, G. Ding, J. Wang, J. Sun, Y. Guo, and P. Yu. Transfer sparse coding for robust image representation. In *Computer Vision and Pattern Recognition (CVPR)*, 2013.
- [24] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu. Transfer feature learning with joint distribution adaptation. In *International Conference on Computer Vision (ICCV)*, 2013.
- [25] K. Muandet, D. Balduzzi, and B. Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning (ICML)*, 2013.
- [26] E. A. Nadaraya. On estimating regression. *Theory of Probability & Its Applications*, 9(1):141–142, 1964.
- [27] M. Palatucci, G. Hinton, D. Pomerleau, and T. M. Mitchell. Zero-shot learning with semantic output codes. In *Neural Information Processing Systems (NIPS)*, 2009.
- [28] V. Patel, R. Gopalan, R. Li, and R. Chellappa. Visual domain adaptation: A survey of recent advances. *Signal Processing Magazine, IEEE*, 2015.
- [29] M. D. Plumbley. Lie group methods for optimization with orthogonality constraints. In *Independent Component Analysis and Blind Signal Separation*, 2004.
- [30] Q. Qiu, V. M. Patel, P. Turaga, and R. Chellappa. Domain adaptive dictionary learning. In *European Conference on Computer Vision (ECCV)*, 2012.
- [31] Q. Rentmeesters. A gradient method for geodesic data fitting on some symmetric riemannian manifolds. In *Conference on Decision and Control and European Control Conference (CDC-ECC)*, 2011.
- [32] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *European Conference on Computer Vision (ECCV)*, 2010.
- [33] S. T. Smith. Optimization Techniques on Riemannian Manifolds. *ArXiv e-prints*, 2014.
- [34] V. M. Tkachuk. Supersymmetric Method for Constructing Quasi-Exactly Solvable Potentials. *eprint arXiv:quant-ph/9806030*, 1998.
- [35] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *Computer Vision and Pattern Recognition (CVPR)*, 2011.
- [36] Z. Wen and W. Yin. A feasible method for optimization with orthogonality constraints. Technical report, Rice University, 2010.

- [37] L. Yang, P. Luo, C. Change Loy, and X. Tang. A large-scale car dataset for fine-grained categorization and verification. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [38] Y. Yang and T. M. Hospedales. A unified perspective on multi-domain and multi-task learning. In *International Conference on Learning Representations (ICLR)*, 2015.
- [39] K. Ye and L.-H. Lim. Distance between subspaces of different dimensions. *ArXiv e-prints*, 2014.