Bridging Heterogeneous Domains With Deep Parallel Transport

Anonymous CVPR submission

Paper ID 247

Abstract

Accounting for different feature types across datasets is a relatively under-studied problem in domain adaptation. We address this heterogeneous adaptation setting using principles from parallel transport and deep sparse coding. By learning generative subspaces from each domain, we first perform label-independent cross-domain feature mapping using parallel transport, and obtain a collection of paths (bridges) that could compensate domain shifts. We encode the information contained in these bridges into an expanded prior, and then integrate the prior into a hierarchical sparse coding framework to learn a selective subset of codes representing holistic data properties that are robust to domain change and feature type variations. We then utilize label information on the sparse codes to perform classification, or in the absence of labels perform clustering, and obtain improved results on several previously studied heterogeneous adaptation datasets. We highlight the flexibility of our approach by accounting for multiple heterogeneous domains in training as well as in testing, and by considering the zero-shot domain transfer scenario where there are data categories in testing which are not seen during training. In that process we also empirically show how existing heterogeneous adaptation solutions can benefit from the findings of our study.

1. Introduction

Domain adaptation, which addresses the problem of change in data characteristics across training (source domain) and testing (target domain) datasets, has received substantial attention over the last few years [11, 37, 27]. Its utility for visual recognition problems has been adequately demonstrated by several efforts that have utilized principles from distance transforms [30], max-margin methods [12], manifolds [16], dictionary learning [29] among others. Many of these approaches have addressed settings where the source domain is labeled and the target domain is unlabeled (unsupervised adaptation), when the target domain also has partial labels (semi-supervised adaptation) and when there is more than one domain in the source and/or target (multiple domain adaptation). However they assume that data, across domains, is represented by same type of features with same dimensions. This is not always possible in practice, and there has been relatively few efforts in the literature addressing the heterogeneous domain adaptation (HDA) setting which allows different feature types across domains.

One of the earliest efforts is by [10] that proposed a translated learning framework using risk minimization principles to bridge data across different feature types. [28] utilized structural correspondence learning by identifying pivot features that share similarities across domains. While such models have restrictions on the type of applications where they can be deployed, [34] proposed a more generic heterogeneous spectral mapping framework that learns an embedding to obtain a common domain-invariant feature subspace that optimally represents data from each domain. Along similar lines, [38] proposed a manifold alignment approach that bridges source and target domain manifolds through a latent space that in addition to preserving topology of each domain, maximizes the intra-class similarities and inter-class dis-similarities across domains. [21] approached this problem by learning asymmetric kernel transformations that perform cross-domain data mapping using semantic similarity. More recently, [23] proposed a feature augmentation strategy coupled with max-margin classifiers, whose formulation resembles multiple kernel learning that in turn guarantees global optimal solution, and [42] studied the utility of canonical correlation subspaces to this problem. Most of these methods either address the HDA problem by learning projections for each domain onto a common latent space where certain properties are satisfied, or by learning feature mapping from one domain onto another directly.

We take a rather different approach which, instead of learning a *few complex transformations* to map domains, learns *several simpler transformations* that explain how data from different domains can be bridged. In learning such transformations we bring the domains to a common dimension using simpler generative information, instead of more complex objectives tied to the domain structure and data similarities. What we gain by doing so is a richer 'expanded' set of prior information which could be harnessed by deep learning methodologies to perform inference. More specifically, given N domains, with each domain representing the data with different feature type (thus having different dimension), we group the data from each domain into k clusters based on their feature similarity and obtain generative subspaces corresponding to each cluster by doing principal component analysis (PCA). We fix all subspaces to have the same dimension p using a heuristic tied to the amount of energy preserved in doing the dimensionality reduction. We then perform parallel transport [15] between subspaces in every domain pair and obtain several intermediate representations that describe how data across domains can be bridged. We subsequently project the data from each domain onto all these intermediate representations to obtain an expanded prior, and integrate it with hierarchical sparse coding [19] to obtain compact codes on which we use label information, if any, to perform cross-domain inference. More details are provided in Section 2. The construction of our approach has the following benefits.

Accommodating Unlabeled Data: In many practical situations, with the widespread availability of multi-modal data on the web, we have very few (or at times no) labeled data and lots of unlabeled data. Our approach can readily utilize such big unlabeled data as we rely on generative modeling in addressing the heterogeneous domain shift. By doing so, when the source and target domains contain the same categories/classes, our final inference can range from the classification scenario where we have labels for all categories in source domain and the target domain may or may not have partial labels, to the clustering scenario where both the source and target domains are unlabeled. The label information is utilized while training a discriminative classifier such as support vector machines (SVM) on the learnt sparse codes, and if no labels are available we perform clustering on the sparse codes using methods such as k-means.

Zero-Shot Domain Transfer: We can also address the zero-shot scenario in which there are categories in the target domain that are not present in the source domain. This is somewhat different from the scenarios discussed above where we atleast had unlabeled data in source domain for all target categories to support inference models. We could handle such a zero-shot scenario as our model is generative and therefore the learned domain shift would have pertinent information for reasoning out new categories.

Multiple Heterogeneous Domains: Finally we can easily accommodate multiple heterogeneous domains in the source as well as in target since we obtain the expanded prior by doing parallel transport between each domain pair. This does not pose a computational bottleneck as we are eventually learning sparse codes in a deep learning setting that could handle big data.

To our best knowledge, our proposed approach is the first to handle these varied aspects of the HDA problem, and while some existing methods could handle a subset of these in principle, we make explicit discussions on these practically relevant requirements. We tested our approach on existing heterogeneous adaptation datasets and obtained good performance improvement over previous results for diverse tasks such as object recognition, event classification, text categorization and sentiment analysis. Detailed experimental analysis is provided in Section 3, and concluding remarks are given in Section 4.

2. Approach

Problem Setting: We assume there are N heterogeneous domains $D = \{D_i\}_{i=1}^N$, where each domain $D_i = \{x_i^j, y_i^j\}_{j=1}^{n_i}$ contains n_i data samples with $x_i^j \in \mathbb{R}^{d_i}$ denoting the feature vector of dimension d_i and y_i^j denoting the corresponding label information (if any) belonging to one of M different categories. These domains could be partitioned into source and target domains depending on the problem situation. With this information, the goal of this work is to account for heterogeneous domain shift in inferring the labels of the unlabeled target domain data.

2.1. Preliminaries

Before we proceed, we will first review relevant details about the tools we use in our approach.

Parallel Transport: We will be working on subspaces derived from the data, and we will generally have multiple subspaces extracted from each domain. In domain adaptation literature, the notion of geodesic on the Grassmann manifold has been used as a bridge to connect a pair of subspaces [17]. When we need to bridge two 'sets' of subspaces instead, parallel transport [15] provides a way by learning multiple paths by which subspace sets can be bridged. More specifically, let $S_1 = \{S_1^i\}_i$ and $S_2 = \{S_2^i\}_i$ denote two sets of p-dimensional subspaces in $\mathbb{R}^d$ corresponding to domains D_1 and D_2 respectively, where each subspace say S_1^1 is a point on the Grassmannian $\mathcal{G}_{d,p}$. Let $g_A(t)$ denote the geodesic with the initial direction $A \in \mathbb{R}^{(d-p)\times p}$ connecting the means of S_1 and S_2, and $\bar{S}_1^1$ denote the tangent space representation of S_1^1 obtained using inverse exponential mapping computed at the mean of S_1. The parallel transport of $\bar{S}_1^1$ is then given as

$$\gamma \bar{S}_1^1(t) = Q_{S_1^1} \exp\left(t \begin{bmatrix} 0 & A^T \\ -A & 0 \end{bmatrix}\right) \begin{bmatrix} 0 \\ B \end{bmatrix} \quad (1)$$

where exp is the matrix exponential, $Q_{S_1^1} \in SO(d)$ is the orthogonal completion of S_1^1, and $B \in \mathbb{R}^{(d-p)\times p}$ is the initial direction to reach from S_1^1 to the exponential map of $\bar{S}_1^1$. Similar directions can be obtained for all subspaces in the sets S_1 and S_2 using the above tangent space approximation. Please refer to [15] for more details. We will discuss how to utilize these directions (bridges) for HDA in Section 2.2.

Deep Sparse Coding: In sparse coding [40] the goal is to represent each input vector $x \in \mathbb{R}^p$ as a sparse linear combination of basis vectors. Given a stacked input data matrix $X \in \mathbb{R}^{p\times n}$, where n is the number of data, it seeks to minimize:

$$\underset{Z\in\mathcal{Z},C}{\arg\min}||X - ZC||_2^2 + \lambda\Omega(C) \tag{2}$$

where $Z \in \mathbb{R}^{p\times r}$ is the dictionary of basis vectors, $\mathcal{Z}$ is the set of matrices whose columns have small ℓ_2 norm and $C \in \mathbb{R}^{r\times n}$ is the code matrix, λ is a regularization hyperparameter, and Ω is the regularizer. In hierarchical sparse coding, such a scheme is extended in a layered fashion using a combination of coding and pooling steps and we pursue the schema presented in [19]. Our modification comes in the way in which the dictionary is initialized and we present the details in Section 2.2.

While there has been an attempt in using parallel transport for unsupervised homogeneous domain adaptation [35], our construction vastly differs from that work as we handle multiple heterogeneous domains without using label information to bridge the domain shift. Moreover, ours is the first approach to integrate parallel transport information with deep sparse coding for adaptation problems.

2.2. Proposed Heterogeneous Adaptation Algorithm

Step 1: We first bring data from all N domains, $D = \{D_i\}_{i=1}^N$, onto a common dimension d by performing PCA on each D_i and choosing the resultant subspace dimension as the largest dimension required among all N subspaces such that 90% of the signal energy is preserved for that decomposition. Then we project data from each domain onto its corresponding subspace. We now have d-dimensional data across all domains, say $\bar{X} = \{\bar{x}_i^j\}_{i,j}$, where i ranges from 1 to N and j ranges from 1 to n_i.

Step 2: From each domain D_i, we then derive k generative subspaces by partitioning $\{\bar{x}_i^j\}_{j=1}^{n_i}$ into k clusters using the k-means algorithm based on the similarity of the d dimensional features, and performing PCA on each cluster. We ensure all the subspaces are of dimension p, by choosing p as the largest dimension required for a subspace, amongst all subspaces obtained by doing k-means in each of the N domains, such that 90% of the signal energy is preserved by that decomposition[1]. Thus from every domain D_i we have a set of p-dimensional subspaces in $\mathbb{R}^d$ denoted by $S_i = \{S_i^j\}_{j=1}^k$. Each subspace in this set is a point on the Grassmann manifold, $\mathcal{G}_{d,p}$. Let $X \in \mathbb{R}^{p\times n}, n = \sum_{i=1}^{N} n_i$ denote the matrix containing the projections of each data in $\bar{X}$ onto its appropriate subspace in $S = \{S_i\}_{i=1}^N$. This is our input data matrix for sparse coding.

Step 3: We then perform parallel transport between S_i's using the method described in Section 2.1. As result we obtain a collection of directions between each pair of (S_i, S_j), i=1,..N-1, $j = i$+1,..,N. We then uniformly sample points along these directions using exponential mapping, which results in new subspaces that contain information on how domain shift information flows between domains. We project each data in $\bar{X}$ onto these subspaces to get the expanded prior $\mathcal{P} \in \mathbb{R}^{p\times r}$, which we in turn use to initialize the dictionary Z.

Step 4: Finally we perform hierarchical sparse coding [19] with the input data matrix X from Step 2 and the initial dictionary Z obtained from Step 3. At the output of each layer of hierarchical sparse coding, we apply Steps 2 and 3 obtain another set of expanded prior which is then used to complement the dictionary of the following layer. Let the final output (from the last layer) of deep sparse coding corresponding to the original data X be denoted by $\hat{X} = \{\hat{x}_i^j\}_{i,j}$, and their corresponding label information (if any) is denoted as before by $Y = \{y_i^j\}_{i,j}$. Note that we have not used any label information thus far.

2.3. Inference

We now perform cross-domain inference using the information contained in $W = (\hat{X}, Y)$. Note that W contains data from both source and target domains, and depending on the dataset we may have one or more domains in the source and target.

Classification: For the classification scenarios widely studied in HDA, source domain contains labeled data for all M categories, and the target domain may or may not have partial labels, and both the source and target domains have the same M categories. So we consider labeled data present in W to train a multi-class SVM [9] with default parameters for linear kernel, and then use the SVM similarity score to classify the unlabeled target domain data into one of M categories. Classification accuracy is computed as the percentage of unlabeled target data that were correctly assigned their category label (using ground truth). Note that while we could have used the label information in any of the previous stages, be it during parallel transport or in sparse coding, we did not because we would like the learned cross-domain

[1] While this is a simple heuristic in addressing variations in feature dimensions, we show that it works well in practice through comprehensive experiments. We also recall that, as stated in the introduction, our main proposal for HDA is by generating an expanded prior on how these domains interact, rather than addressing domain shift 'during' the process of bringing domains to a common dimension as done by most existing methods.

representations $\hat{X}$ to be generic to support other inference scenarios discussed next. Nevertheless, we make some observations regarding this later in Section 3.6.

Clustering: We also address the clustering scenario where both source and target domain data are unlabeled, and they have the same M categories. In this case we cluster all the data in $\hat{X}$ into M groups using k-means, and compute the clustering accuracy using the fairly standard method of labeling each of the resulting clusters with the majority class label according to the ground truth, and measuring the number of mis-classifications made by each cluster grouping.

Zero-shot Learning: We finally account for the zero-shot learning scenario where the target domain has some categories that are not a part of the M source domain categories. For this case, we use labels for M categories in the source domain and (if available) in the target domain to train the SVM as discussed for the classification scenario. We then threshold the SVM similiarity score for the unlabeled target data, with the hope that if such data comes outside of the M source categories, the similarity score will be less. We then cluster such data using k-means to obtain groupings, with the number of clusters set to the number of new target categories known apriori, and evaluate the accuracy as discussed above for the clustering scenario. If we do not even know the number of new target categories, then it becomes difficult to quantify clustering accuracy.

3. Experiments

We first discuss the classification scenario and experiment with the setup used by [23] for the problems of heterogeneous object recognition, text categorization and sentiment analysis. These experiments have only one domain each in the source and target. We then consider the event classification experiment designed by [8] which consists of multiple source domains and a single target domain. We then provide a detailed analysis of the findings from our study.

3.1. Object Recognition

We work with the Office dataset [30] that contains a total of 4106 images from 31 categories collected from three sources: amazon (object images downloaded from Amazon), dslr (high-resolution images taken from a digital SLR camera) and webcam (low-resolution images taken from a web camera). SURF features are extracted for all the images. The images from amazon and webcam are clustered into 800 visual words by using k-means. After vector quantization, each image is represented as a 800 dimensional histogram feature. Similarly, we represent each image from dslr as a 600-dimensional histogram feature. In the experiments, dslr is used as the target domain, while amazon and webcam are considered as two individual source domains. For training the SVM, we randomly select 20 labeled images per category for the source domain amazon, and 8 labeled images per category for webcam as source domain. For the target domain dslr, 3 labeled images are randomly selected from each category. The remaining target domain data is used for testing. We present results of our HDA approach in Table 1 along with other methods studied in [23].

Methods	Source Domain	
	amazon	webcam
[34]	42.8±2.4	42.2±2.6
[38]	53.3±2.3	53.2±3.2
[21]	53.1±2.4	53.0±3.2
[23]	55.4±2.9	54.3±3.6
Ours	62.1±1.7	61.5±2.1

Table 1. Mean and standard deviation of classification accuracy (%) on the Object Recognition Dataset with target domain dslr.

3.2. Text Categorization

We use the Reuters multilingual dataset [2] which contains about 11K newswire articles from 6 categories in 5 languages namely, English, French, German, Italian and Spanish. All documents are represented by using the TF-IDF feature. We perform PCA based on the TF-IDF features from each domain with 60% energy preserved and thus the features for each language have the following dimensions respectively, 1131, 1230, 1417, 1041 and 807. We consider Spanish as the target domain in the experiments and use each of the other four languages as individual source domains. For each category, we randomly sample 100 labeled documents from the source domain and either 10 or 20 labeled documents from the target domain to train the SVM. The remaining documents in the target domain are used as the test data. We report classification results of our HDA approach in Table 2 along with the results of the other approaches discussed in [23].

3.3. Sentiment Analysis

We use the Cross-Lingual Sentiment (CLS) dataset [28], which is an extended version of the Multi-Domain Sentiment Dataset [5] widely used for domain adaptation. It is collected from Amazon and contains about 800,000 reviews of three product categories: Books, DVDs and Music, and written in four languages: English, German, French, and Japanese. The English reviews were sampled from the Multi-Domain Sentiment Dataset and reviews in other languages are crawled from Amazon. For each category and each language, the dataset is partitioned into a training set, a test set consisting of 2,000 reviews each. We take English as the source domain and each of the other three languages as an individual target domain in the experiment. We randomly sample 500 reviews from the training set of the source domain and 100 reviews from the training set of

Methods	Source Domain							
	10 labels per target class				20 labels per target class			
	English	French	German	Italian	English	French	German	Italian
[34]	54.7±7.4	55.0±9.4	58.0±7.9	59.4±3.7	65.7±3.1	64.2±4.2	64.6±3.6	65.8±2.3
[38]	65.0±2.9	66.9±2.1	67.5±2.1	68.5±2.8	72.4±2.4	72.8±2.0	72.9±2.3	73.3±2.1
[21]	65.7±2.7	66.9±1.7	68.7±2.9	67.9±2.8	72.9±2.0	73.5±1.8	74.7±1.6	74.0±2.0
[23]	68.6±2.3	69.5±1.9	69.8±2.7	69.8±2.5	75.3±1.7	75.7±1.6	76.1±1.5	75.8±1.8
Ours	73.2±1.6	73.8±1.9	74.1±2.3	74.0±1.7	83.5±2.4	84.1±1.9	83.8±2.1	84.2±1.5

Table 2. Mean and standard deviation of classification accuracy (%) on the Reuters Multilingual Dataset with target domain Spanish.

Methods	Target Domain		
	German	French	Japanese
[34]	50.4±0.6	49.8±0.6	51.3±1.0
[38]	64.6±1.9	65.7±1.8	64.4±1.8
[21]	58.3±3.0	59.4±4.3	57.5±1.9
[23]	66.5±2.2	66.9±2.1	64.2±2.5
Ours	71.2±2.1	71.5±1.8	70.9±1.5

Table 3. Mean and standard deviation of classification accuracies (%) on the Cross-lingual Sentiment Dataset with source domain English.

Methods	Target Domain	
	Kodak	CCV
[6]	43.49	41.55
[14]	44.21	38.56
[13]	46.21	43.44
[8]	49.61	44.52
Ours	54.56	51.24

Table 4. Mean average precision (%) on the Event classification dataset with multiple heterogeneous source domains from Google image search, Bing image search and YouTube video search.

the target domain as the labeled data to train the SVM. The test set is the official test set for each category and each language. As with text categorization experiment, we extracted the TF-IDF features and performed PCA with 60% energy preserved to result in dimensions 715, 929, 964 and 874 respectively for the four languages discussed above. Results on this dataset are presented in Table 3.

3.4. Event Classification

We then worked on the event classification dataset of [8] that accounts for multiple heterogeneous source domains and a single target domain. The events pertain to six classes namely, birthday, picnic, parade, show, sports and wedding. Three source domains correspond to Google and Bing image search, and Youtube video search corresponding to these events. Kodak and CCV dataset serve as the individual target domain. The Google and Bing domains are represented by a 4000 dimensional bag-of-words codebooks learnt on SIFT features, while YouTube, Kodak and CCV datasets are represented by 6000 dimensional spatio-temporal features corresponding to histogram of oriented gradient, histogram of optical flow and motion boundary histogram. By considering the source domains to be labeled and target domain completely unlabeled we trained the SVM on the source domain samples to perform separate inference on CCV and Kodak datasets as the target domain. The cross-domain event classification results are presented in Table 4.

3.5. Discussion

Clustering and Zero-shot Learning: We see that our HDA approach outperforms existing methods on diverse heterogeneous classification tasks. We also performed these experiments for the clustering scenario by not considering any labeled data from the source and target domains using the approach discussed in Section 2.3. The performance decreased by around 15% on average from the classification results. While this is reasonable since label information always helps in performing class-specific inference, it also shows that our generative heterogeneous model outputs $\hat{X}$ contain information agnostic to domain and feature variations and thus is relevant in grouping data categories.

We then considered the zero-shot learning scenario, by considering the classification experiments and holding out 10%, 20% and 30% of the categories as being exclusive to the target which the source domain has not seen. As per the discussion in Section 2.3, we first thresholded the SVM similarity scores for data from these new categories. With the similarity score ranging from 0 (low) to 1(high), we tried three thresholds namely 0.3, 0.2 and 0.1. Ideally these new categories should have lower similarities since they are not part of the trained model. On average we obtained a 95% filtering accuracy with these thresholds, across all datasets discussed before, and then we grouped the filtered data to get an accuracy of 85% on average. Note that these results are only for the new data categories, which are much less in number than those considered for classification scenario and hence can not be compared with those results. As a sanity check we tested the unlabeled data from the target categories that are present in the source, and observed that their SVM similarity scores were always greater than 0.3 for all datasets. These results convey that our model outputs are quite useful in reasoning out never seen before categories, which is very important in practice.

Parameter Tuning: For all the results discussed thus far, we used $k = 10$ clusters within each domain, and uniformly sampled 10 points on the parallel transport directions. We tried other values 8 and 12 for both clusters and sampling on directions, and PCA energy of 80% and 85% in learning the generative subspaces from the domains, and found the results decreased at the most by 2%. We used default parameters for other tools we have employed in the approach. This sheds light on the robustness of our approach. It takes about 5 to 10 seconds on a single 2GHz machine to perform inference over the range of scenarios discussed here.

Design Choice Analysis: We now analyze the rationale behind some choices we made in the approach. Firstly, we tested whether hierarchical sparse coding is necessary, or will a single layer sparse coding be sufficient. We also tested how many layers are necessary by experimenting upto five layers. The results reported in the paper are with three layers, and when we used four and five layers, the results reduced by 2.5% and 3% on average, using two layers saw average performance reduction of around 8% and using one layer saw a reduction of 17%. This suggests that hierarchical feature learning is useful, and the results reach a plateau around three layers for our experiments.

Then we inspect whether learning multiple bridges across domains using parallel transport is necessary, or just a single bridge using the geodesic will suffice. Results using only the geodesic was inferior by around 21% which highlights the utility of parallel transport to the HDA problem. We also test whether we need to do parallel transport to obtain expanded prior on the outputs of each layer of deep sparse coding, by just initializing the first layer with the prior and doing hierarchical feature learning on it. That resulted in a performance drop of about 12% on average.

Multiple Source and Multiple Target Domains: Our experiments so far contained single source and single target domain, or multiple source domains and single target domain. We now pursue multiple source domains and multiple target domains on these datasets, where possible. Given N domains, we try all possible combinations across source and target domains. For example, if we have four domains, we consider 1 to 3 source domains that are accompanied by 3 to 1 target domains respectively. Our results for such a setting improves the results discussed in Sec 3.1 to 3.4 by atleast 3% and upto 12%. This explains the utility of our method for HDA with increasing availability of domains.

3.6. Extensions

In this section we discuss some extensions of our approach, by relaxing certain assumptions that were made to facilitate its generalizability to different adaptation settings.

Using Label Information In Modeling Stage: Till now the labels were used only in the classification stage (Sec 2.3) and not in the modeling stage (Sec 2.2) as we wanted the model to handle classification, clustering and zero-shot learning. But in cases where say, the goal is just classification and there are labels available for training the model, it makes sense to use them in the model building stage itself. To support such a scenario, we modify our approach (in Step 4) by performing discriminative hierarchical sparse coding. We use the method of [20] and feed it with data labels contained in Y. Thus, we obtain the sparse codes output $\hat{X}$ which in addition to minimizing the reconstruction error of the data samples, also separates samples belonging to one class from other classes. Let us call this Case A. With this modification, the performance for classification experiments reported in Section 3.1 to 3.4 improved by atleast 3% and upto 15% on average.

Another way to utilize label information is in forming the clusters within each domain (Step 2). Instead of using the similarity of the d dimensional features to group the data into k clusters, we group the data using their labels into M clusters and then perform the remaining steps as outlined in Section 2.2. So in this case the parallel transport information will have a notion of class discrimination in traversing the domain shift. Let us call this Case B. This resulted in an average performance improvement of atleast 1.8% and upto 9% for the classification experiments reported in Section 3.1 to 3.4. We then used Case A and Case B together, and this improved the results by atleast 5% and upto 20% on average.

Integrating With Other Heterogeneous Adaptation Strategies: As mentioned in the introduction, one of the goals of our study was to see how a large number of simple transformations would fare against few complex transformations to handle heterogeneous domain shift, which was the reason to map all domains to a common dimension using simpler generative information (Step 1). This strategy was shown to be successful through detailed experiments. Now we study how to get the best of both approaches. For this, instead of Step 1, we use outputs from existing heterogeneous adaptation techniques which map different dimensions onto a common one using more involved objectives related to domain structure. We tried two such techniques, one based on spectral mapping [34] and the other based on manifold alignment [38]. The mapped output of these techniques, where the data from all domains $\{D_i\}_{i=1}^N$ will have the transformed to the same dimension d, signifies $\bar{X}$ which is then fed into Step 2, and the remaining steps from Sec 2.2 and 2.3 are followed. This resulted in a performance improvement of atleast 5% and upto 18%, and atleast 4.2% and upto 16.5% while using [34] and [38] respectively, for classification, clustering, and zero-shot learning experiments reported in Section 3.1 to 3.5. These results hold promise on the utility of our approach to existing adaptation solutions.

Heterogeneous View Of Homogeneous Adaptation: Encouraged by these observations, we considered homogeneous adaptation problems that have been extensively studied in the literature [30], which assumes the data is represented by same features (same dimensions) in both source and target domains. The goal of our study here is to represent such data with many different features, with each feature forming a separate domain, and empirically investigate whether multiple features can make adaptation problems easier to handle. We experiment with different combinations of heterogeneous features in the source and target and at the same time making sure that the same feature type is not used for both source and target. The results presented below are averaged over such combinations.

We first consider the Office-Caltech dataset [16] for adaptive object recognition that contains 10 objects with four domains namely, amazon, dslr, webcam, and Caltech. The data is represented by bag-of-words codebooks learnt from SURF features. We additionally extracted histogram of oriented gradients, local binary patterns, local phase quantization, and GIST descriptors. Thus we have five domains each in the source and target domain. We then followed the adaptation protocol of [16] that first considered labeled data from source domain and no labels from target domain, and then allows few labeled samples from target as well. Our results are provided in Table 5.

We then worked on adaptive face recognition and considered the following facial features, image intensities in RGB and HSV color spaces, edge magnitudes and gradient direction, multi-scale block LBP, self-quotient image and Gabor wavelets. We first followed the protocol of [33] that used the Multi-PIE dataset [18] with images of 129 subjects in frontal pose as the source domain, and five other off-frontal poses as the target domain. Images under five illumination conditions across source and target domains were used for training with which images from remaining 15 illumination conditions in the target domain were recognized. Face recognition accuracy for this experiment is given in Table 6. We also performed an experiment on the PIE dataset [36] using the protocol of [26], where the domain change is caused by illumination and blur. Frontal pose faces of 34 subjects under 11 different illumination conditions formed the source domain, while the unlabeled target domain consisted of frontal images of the same subjects under 10 other lighting conditions that were also blurred using four kernels, a Gaussian with standard deviation of 3 and 4 and motion blur with length 9, angle 135° and length 11, angle 45°. The results are provided in Table 7.

Method	Mean classification accuracy (%) Target domain pose				
	15°	30°	45°	60°	75°
[31]	92.1	89.7	88.0	86.1	83.0
[32]	99.7	99.2	98.6	94.9	95.4
[41]	96.8	90.6	94.4	91.4	90.5
[33]	98.4	98.2	98.9	99.1	98.8
Ours	99.5	99.3	99.1	99.4	98.9

Table 6. Comparison with homogeneous adaptation methods on the CMU Multi-PIE dataset for face recognition across pose and illumination variations with few labeled target data.

Method	Mean classification accuracy (%)			
	$\sigma = 3$	$\sigma = 4$	$len = 9$	$len = 11$
[1]	66.5	32.9	73.8	62.1
[17]	70.9	60.3	72.4	67.9
[16]	78.5	77.7	82.4	77.7
[26]	80.3	77.9	85.9	81.2
Ours	86.3	85.2	87.9	87.2

Table 7. Comparison with homogeneous adaptation methods on the CMU PIE dataset for face recognition across blur and illumination variations with no labeled target data. σ refers to standard deviation of the Gaussian blur, and len refers to the length of linear motion blur.

Finally we performed adaptation experiments for action recognition on the IXMAS multi-view action dataset [39] that contains eleven action categories including walk, kick and throw. Each action was performed three times by twelve actors taken from five different views, which include four side views and one top view. From each action video we extracted the following features, histograms of oriented gradients and optical flow (HOG/HOF), 3DHOG that is based on 3D gradient orientations, and extended SURF descriptor for videos. The experiment was to recognize actions with the source and target domain pertaining to different views, thereby making 20 source-target combinations. The average cross-view recognition accuracy results for all these view combinations are given in Table 8. The correspondence mode contains matched instances across source and target domains, partial labels mode having fewer target labels and the non-discriminative virtual views (NDVV) comprising of partially-labeled and unlabeled target data, as studied in [22].

All these results show our method outperforming other homogeneous adaptation approaches, which suggests that such approaches could, in some form, benefit by expanding the type of features used to represent the data. The features we have used are only somewhat representative of the vast literature and more of them can be used with our approach.

Domain		Classification accuracy (in %), mean±std. deviation					
Source	Target	No labeled target data			Few labeled target data		
		[4]	[25]	Ours	[26]	[3]	Ours
Caltech	Amazon	52.3±1.1	46.76	56.3±1.1	49.5±2.6	61.8±2.5	65.3±1.2
Caltech	Dslr	53.0±2.3	44.59	55.2±1.8	76.7±3.9	65.8±3.5	79.8±1.2
Amazon	Caltech	44.4±1.4	39.45	49.2±1.3	27.4±2.4	47.8±1.5	51.1±0.3
Amazon	Webcam	48.5±2.6	42.03	51.6±2.1	72.0±4.8	72.5±3.1	75.5±1.1
Webcam	Caltech	39.3±0.5	30.19	42.8±1.8	29.7±1.9	43.6±1.2	45.5±2.8
Webcam	Amazon	44.3±0.9	29.96	45.2±2.5	49.4±2.1	53.4±1.9	55.5±1.7
Dslr	Amazon	39.4±1.1	32.78	44.3±1.2	48.9±3.8	56.9±1.6	57.5±1.5
Dslr	Webcam	88.8±1.0	85.42	91.2±1.7	72.6±2.1	89.1±1.6	92.3±2.6

Table 5. Performance comparison with homogeneous adaptation methods on the Office-Caltech dataset consisting 10 object categories.

Method	Mean classification accuracy (%)		
	correspondence	partial labels	NDVV
[24]	75.3	-	-
[7]	78.8	-	-
[22]	81.2	61.2	61.1, 26.0
[43]	85.8	69.0	69.2, 35.3
Ours	89.5	74.3	75.3, 45.4

Table 8. Comparison with homogeneous adaptation methods on the IXMAS dataset for cross-view action recognition.

4. Conclusion

We have approached the problem of bridging heterogeneous domains by learning a large set of intermediate representations, born out of simpler generative information, and integrated them with deep feature learning mechanisms to perform inference pertaining to classification, clustering and zero-shot learning. We demonstrated superior empirical performance over existing methods on a wide range of problems involving different data modalities such as images, videos and natural language. We also discussed the utility of our design choices, and the robustness of the approach in dealing with multiple domains, feature types, unlabeled data and new unseen data categories. While our approach offers an alternative to many existing heterogeneous solutions that address domain shift through a latent space modeling, it also opens up opportunities for harnessing the benefits of the two strategies, which we highlighted through some initial studies. Such a mechanism could eventually pave way for establishing error bounds on the nature of heterogeneous shifts an approach can handle, within a reasonable set of domain shift assumptions reflecting practical data acquisition constraints.

References

[1] T. Ahonen, E. Rahtu, V. Ojansivu, and J. Heikkila. Recognition of blurred faces using local phase quantization. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008. 7

[2] M. Amini, N. Usunier, and C. Goutte. Learning from multiple partially observed views-an application to multilingual text categorization. In *Advances in neural information processing systems*, pages 28–36, 2009. 4

[3] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann. Unsupervised domain adaptation by domain invariant projection. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 769–776. IEEE, 2013. 8

[4] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann. Domain adaptation on the statistical manifold. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2481–2488. IEEE, 2014. 8

[5] J. Blitzer, M. Dredze, F. Pereira, et al. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*, volume 7, pages 440–447, 2007. 4

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

[7] C.-Y. Chen and K. Grauman. Inferring unseen views of people. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2011–2018. IEEE, 2014. 8

[8] L. Chen, L. Duan, D. Xu, and D. Xu. Event recognition in videos by learning from heterogeneous web sources. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2666–2673. IEEE, 2013. 4, 5

[9] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *The Journal of Machine Learning Research*, 2:265–292, 2002. 3

[10] W. Dai, Y. Chen, G.-R. Xue, Q. Yang, and Y. Yu. Translated learning: Transfer learning across different feature spaces. In *Advances in neural information processing systems*, pages 353–360, 2008. 1

[11] H. Daume III and D. Marcu. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, pages 101–126, 2006. 1

[12] L. Duan, I. W. Tsang, D. Xu, and S. J. Maybank. Domain transfer svm for video concept detection. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1375–1381. IEEE, 2009. 1

[13] L. Duan, D. Xu, and S.-F. Chang. Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1338–1345. IEEE, 2012. 5

[14] L. Duan, D. Xu, and I. W. Tsang. Domain adaptation from multiple sources: A domain-dependent regularization approach. *Neural Networks and Learning Systems, IEEE Transactions on*, 23(3):504–518, 2012. 5

[15] A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM journal on Matrix Analysis and Applications*, 20(2):303–353, 1998. 2, 3

[16] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2066–2073. IEEE, 2012. 1, 7

[17] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 999–1006. IEEE, 2011. 2, 7

[18] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-pie. *Image and Vision Computing*, 28(5):807–813, 2010. 7

[19] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach. Proximal methods for hierarchical sparse coding. *The Journal of Machine Learning Research*, 12:2297–2334, 2011. 2, 3

[20] Z. Ji, W. Huang, G. Kenyon, and L. Bettencourt. Hierarchical discriminative sparse coding via bidirectional connections. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 2844–2851, 2011. 6

[21] B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1785–1792. IEEE, 2011. 1, 4, 5

[22] R. Li and T. Zickler. Discriminative virtual views for cross-view action recognition. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2855–2862. IEEE, 2012. 7, 8

[23] W. Li, L. Duan, D. Xu, and I. W. Tsang. Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(6):1134–1148, 2014. 1, 4, 5

[24] J. Liu, M. Shah, B. Kuipers, and S. Savarese. Cross-view action recognition via view knowledge transfer. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE conference on*, pages 3209–3216. IEEE, 2011. 8

[25] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu. Transfer joint matching for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1410–1417. IEEE, 2014. 8

[26] J. Ni, Q. Qiu, and R. Chellappa. Subspace interpolation via dictionary learning for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 692–699. IEEE, 2013. 7, 8

[27] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa. Visual domain adaptation: A survey of recent advances. *IEEE Signal Processing Magazine*, 32(3):53–69, 2015. 1

[28] P. Prettenhofer and B. Stein. Cross-language text classification using structural correspondence learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1118–1127. Association for Computational Linguistics, 2010. 1, 4

[29] Q. Qiu, V. M. Patel, P. Turaga, and R. Chellappa. Domain adaptive dictionary learning. In *Computer Vision–ECCV 2012*, pages 631–645. Springer, 2012. 1

[30] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *Computer Vision–ECCV 2010*, pages 213–226. Springer, 2010. 1, 4, 7

[31] A. Sharma and D. W. Jacobs. Bypassing synthesis: Pls for face recognition with pose, low-resolution and sketch. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 593–600. IEEE, 2011. 7

[32] A. Sharma, A. Kumar, H. Daume, and D. W. Jacobs. Generalized multiview analysis: A discriminative latent space. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2160–2167. IEEE, 2012. 7

[33] S. Shekhar, V. M. Patel, H. V. Nguyen, and R. Chellappa. Generalized domain-adaptive dictionaries. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 361–368. IEEE, 2013. 7

[34] X. Shi, Q. Liu, W. Fan, P. S. Yu, and R. Zhu. Transfer learning on heterogenous feature spaces via spectral transformation. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 1049–1054. IEEE, 2010. 1, 4, 5, 6, 7

[35] A. Shrivastava, S. Shekhar, and V. M. Patel. Unsupervised domain adaptation using parallel transport on grassmann manifold. In *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*, pages 277–284. IEEE, 2014. 3

[36] T. Sim, S. Baker, and M. Bsat. The cmu pose, illumination, and expression (pie) database. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pages 46–51. IEEE, 2002. 7

[37] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1521–1528. IEEE, 2011. 1

[38] C. Wang and S. Mahadevan. Heterogeneous domain adaptation using manifold alignment. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 1541, 2011. 1, 4, 5, 6, 7

[39] D. Weinland, E. Boyer, and R. Ronfard. Action recognition from arbitrary views using 3d exemplars. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–7. IEEE, 2007. 7

[40] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1794–1801. IEEE, 2009. 3

[41] M. Yang, D. Zhang, and X. Feng. Fisher discrimination dictionary learning for sparse representation. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 543–550. IEEE, 2011. 7

[42] Y.-R. Yeh, C.-H. Huang, and Y.-C. F. Wang. Heterogeneous domain adaptation and classification by exploiting the correlation subspace. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, 23(5):2009, 2014. 1

[43] Z. Zhang, C. Wang, B. Xiao, W. Zhou, S. Liu, and C. Shi. Cross-view action recognition via a continuous virtual path. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2690–2697. IEEE, 2013. 8