Adaptive Transductive Transfer Machines

Nazli Farajidavar
http://personal.ee.surrey.ac.uk/Personal/N.Farajidavar
Teofilo deCampos
http://personal.ee.surrey.ac.uk/Personal/T.Decampos
Josef Kittler
http://www.surrey.ac.uk/cvssp/people/josef_kittler

CVSSP
University of Surrey
Guildford, GU2 7XH
UK

Abstract

Classification methods traditionally work under the assumption that the training and test sets are sampled from similar distributions (domains). However, when such methods are deployed in practise, the conditions in which test data is acquired do not exactly match those of the training set. In this paper, we exploit the fact that it is often possible to gather unlabeled samples from a test/target domain in order to improve the model built from the training source set. We propose Adaptive Transductive Transfer Machines, which approach this problem by combining four types of adaptation: a lower dimensional space that is shared between the two domains, a set of local transformations to further increase the domain similarity, a classifier parameter adaptation method which modifies the learner for the new domain and a set of class-conditional transformations aiming to increase the similarity between the posterior probability of samples in the source and target sets. We show that our pipeline leads to an improvement over the state-of-the-art in cross-domain image classification datasets, using raw images or basic features.

1 Introduction

In object classification, it is expensive to acquire vast amounts of labelled training samples in order to provide classifiers with a good coverage of the feature space. One possible way of dealing with this problem is to synthesise images of training objects using computer graphics techniques (e.g. [28]), but their appearance may not be realistic and it is not feasible to model all possible backgrounds. Practitioners often resort to crowd sourcing [5], but the annotations obtained are either costly or unreliable. The alternative exploited in this paper is to use transfer learning methods. Unlike traditional machine learning methods, transfer learning (TL) methods do not assume that training and test data are drawn from the same distribution [22]. The field of TL includes a range of problems in which there is a change of domain or task between source and target sets. TL techniques are becoming more popular in Computer Vision, particularly after Torralba and Efros [31] discovered significant biases in object classification datasets. However, much of the work focuses on *inductive* transfer learning problems, which assume that labelled samples are available both in source and target domains. In this paper we focus on the case in which only unlabelled samples are available in the target domain. This is a *transductive* transfer learning (TTL) problem, i.e., the joint probability distribution of samples and classes in the source domain $P(\mathtt{X}^{src}, \mathtt{Y}^{src})$ is assumed

to be different, but related to that of a target domain joint distribution, $P(\mathtt{X}^{trg}, \mathtt{Y}^{trg})$, while labels $\mathtt{Y}^{trg}$ are not available in the target set.

TTL methods can potentially improve a very wide range of classification tasks, as it is often the case that a domain change happens between training and application of algorithms, and it is also very common that unlabelled samples are available in the target domain. For example, in image classification, the training set may come from high quality images (e.g. from DSLR cameras) and the target test set may come from mobile devices. TTL methods can potentially generalise classification methods for a wide range of domains and make them scalable for big data problems.

In this paper, we propose Adaptive Transductive Transfer Machine (ATTM) which combines methods that adapt the marginal and the conditional distribution of the samples, so that source and target datasets become more similar, facilitating classification. This involves two terms, marginal and conditional distributions, the distribution of the data and the distribution of the data given the classes. We further introduce two unsupervised dissimilarity measures which are the backbones of our classifier adaptation approach. ATTM uses these measures to select the best classifier and to further optimise its parameters for a new target domain. We show that our method obtains state-of-the-art results in cross-domain vision datasets using naïve features, with a significant gain in computational efficiency in comparison to related methods.

In the next section, we briefly review related works and give an outline of our contribution. Section 3 presents the core components of our method and further discusses its relationship to previous works. This is followed by a description of our framework and an analysis of our algorithm. Experiments and conclusions follow in sections 4 and 5.

2 Related work

Pan and Yang [22] presented taxonomy of TL methods which include Inductive TL, when labelled samples are available in both source and target domains; Transductive TL, when labels are only available in the source set, and Unsupervised TL, when labelled data is not present. They also categorised the methods based on *instance re-weighting* (e.g. [10, 12]), *feature space transformation* (e.g. [4, 20]) and *learning parameters transformation* (e.g. [2, 6]).

For the reasons highlighted in Section 1, we focus on Transductive TL problems (TTL). They relate to sample selection bias correction methods [11, 18], where training and test data follow different distributions but the label sets remain the same. A popular method for TTL is Transductive SVM [19] and its extended version, domain adapted SVM [6], simultaneously learn a decision boundary and maximise the margin in the presence of unlabelled patterns, without requiring density estimation. In contrast, Gopalan et al. [16] used a method based on Grassmann manifold in order to generate intermediate data representations to model cross-domain shifts. In [10], Chu et al. proposed to search for an instance-based re-weighting matrix applied to the source samples. The weights are based on the similarity between the source and target distributions using the Kernel Mean Matching algorithm.

Different types of methods can potentially be combined. In this paper, we focus on *feature space transformation* and *learning parameters adaptation*. We approach the TTL problem by finding a set of transformations that are applied to the source domain samples $G(\mathtt{X}^{src})$ such that the joint distribution of the transformed source samples becomes more similar to that of the target samples, i.e. $P(G(\mathtt{X}^{src})), \mathtt{Y}^{src}) \approx P(\mathtt{X}^{trg}, \mathtt{Y}^{trg*})$, where $\mathtt{Y}^{trg*}$ are the

labels estimated for target domain samples.

Following this line of work, Long et al. [20] proposed to do Joint Distribution Adaptation (JDA) by iteratively adapting both the marginal and conditional distributions using a procedure based on a modification of the Maximum Mean Discrepancy (MMD) algorithm [4]. JDA uses the pseudo target labels to define a shared subspace between the two domains. At each iteration, JDA requires the construction and eigen decomposition of an $n \times n$ matrix whose complexity can be up to $O(n^3)$, where n is the number of samples.

We propose the Adaptive Transductive Transfer Machine pipeline which first searches for a global transformation such that the marginal distributions of the two domains become more similar and then with the same objective applies a set of local transformations to each transformed source domain sample. Finally in an iterative scheme, the algorithm aims to reduce the difference between the conditional distributions in source and target spaces. We also propose two dissimilarity measures to select a proper classifier and adjust the learning parameters for the new domain. The complexity of the iterative step of the proposed pipeline is linear on the number of features in the space, i.e., $O(f)$.

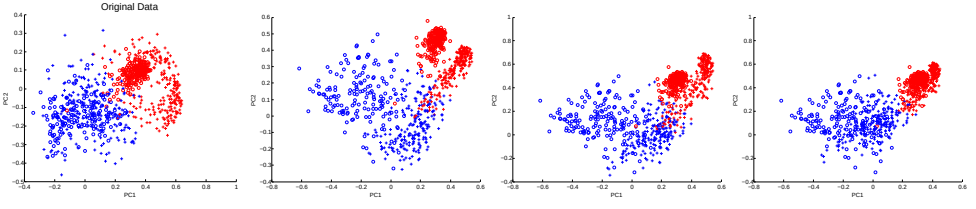
3 Marginal and conditional distribution adaptation

We propose the following pipeline, where the notation used is summarised in Table 1:

(a) A global linear transformation G^1 is applied to $\mathtt{X}^{src}$ and $\mathtt{X}^{trg}$ such that the marginal $P(G^1(\mathtt{X}^{src}))$ becomes more similar to $P(G^1(\mathtt{X}^{trg}))$.

(b) With the same objective, a local transformation is applied to each transformed source domain sample $G^2_i(G^1(\mathrm{x}^i_{src}))$.

(c) Finally, aiming to reduce the difference between the conditional distributions in source and target spaces, a class-based transformation is applied to each of the transformed source samples $G^3_{y^j}(G^2_i(G^1(\mathrm{x}^i_{src})))$.



(a) MMD (b) TransGrad (c) TST

Figure 1: The effect of different steps of our pipeline on digits 1 and 2 of the MNIST→USPS datasets, visualised in 2D through PCA. The source dataset (MNIST) is indicated by stars, the target dataset (USPS) is indicated by circles, red indicates samples of digit 1 and blue indicates digit 2 (better viewed on the screen).

Figure 1 illustrates the effect of the three steps of the pipeline above on a dataset composed of digits 1 and 2 samples selected from the MNIST and USPS datasets where data is projected into a two-dimensional subspace using PCA. The source dataset is indicated by stars, the target dataset is indicated by circles and red indicates class 1 and blue indicates 2. The original space projection is generated by using the first two principal components of the source data. The effect of step (a) is to bring the mean of the two distributions closer to

each other while it projects the data into its principal components directions of the full data including the source and target[1]. For marginal distribution adaptation, we adopt empirical Maximum Mean Discrepancy (MMD) measure [17, 20, 30] to compare different distributions and compute a lower-dimensional embedding that minimises the distance between the expected values of samples in source and target domains.

Table 1: Notation and acronyms used most frequently in this paper.

$\mathtt{X} = [\mathrm{x}^1, \cdots, \mathrm{x}^i, \cdots, \mathrm{x}^n]^\top \in \mathbb{R}^{n \times f}$	Input data matrix with n samples of f features
$\mathrm{x}^i = (x^i_1, \cdots, x^i_j, \cdots, x^i_f)^\top$	Feature vectors
$\mathtt{Y} = (y^1, \cdots, y^n)^\top$	Array of class labels associated to $\mathtt{X}$
$\mathcal{Y} = \{1, \cdots, C\}$	Set of classes
$\mathtt{X}^{src} \in \mathbb{R}^{n_{src} \times f}$, $\mathtt{X}^{trg} \in \mathbb{R}^{n_{trg} \times f}$	Source and target data matrices
Λ_{src}	Classification model trained with $\mathtt{X}^{src}$
$G(\mathtt{X})$	Transformation function
θ	transfer rate parameter
T	Number of iterations
$\lambda = \{w_k, \mu_k, \Sigma_k, k = 1, \cdots, K\}$	GMM parameters with K components
$E^{src}[x_j, y^i], E^{trg}[x_j, y^i]$	Joint expectation of feature j and label y^i
$D(p,q)$	Dissimilarity between two distributions
γ	TransGrad translation regulator
TL, ITL, TTL	Transfer Learning, Inductive TL, Transductive TL
MMD	Maximum Mean Discrepancy
TransGrad	Sample-based transformation using gradients
TST	Class-based Translation and Scaling Transform

For the second step of our pipeline (Fig. 1(b)), we propose a method that distorts the source probability density function towards target clusters. We employ a sample-wise transformation that uses likelihoods of source samples given a GMM that models target data. To our knowledge, this is the first time a sample-based transformation is proposed for transfer learning. In the final step (Fig. 1(c)), the source class-conditional distributions are iteratively transformed to become more similar to their corresponding target conditionals, following the work in [1, 13]. For *learning parameters adaptation*, we introduce two unsupervised dissimilarity measures which are used for selecting a proper classifier and for adapting its parameters. The next subsections describe each of the steps above. Further details and derivations are in the supplementary material.

3.1 Shared space detection with MMD

In the first step of our pipeline, we look for a shared space projection that reduces the dimensionality of the data whilst minimising the reconstruction error. This aims at minimising the marginal distribution differences between the source and target domains. While there are many shared space projection techniques available in literature [8, 21, 26], we follow [17, 20, 23, 30] and adopt the Maximum Mean Discrepancy (MMD) for comparing different distributions.

This algorithm searches for a projection matrix in $\mathbb{R}^{f \times k}$ which aims to minimise the distance between the sample means of the source and target domains. The effect is to obtain

[1] In Figure 1(a), the feature space is visualised with PCA projection and only two classes are shown, while the MMD computation was performed in a higher dimensional space on samples from 10 classes. For these reasons it may not be easy to see that the means of source and target samples became closer after MMD.

a lower dimensional shared space between the source and target domains. Under the new representation the marginal distributions of the two domains are thus drawn closer to each other.

3.2 TransGrad

We propose a sample-based transformation to refine the PDF of source domain samples. In general, target data may, but does not have to, lie in the same observation space. However, for the sake of simplicity, we shall assume that the transformation from the source to the target domain is locally linear, i.e. a sample's feature vector x^i from the source domain is mapped to the target space by

$$G_i^2(\mathrm{x}^i) = \mathrm{x}^i + \gamma \mathrm{b}^i \,, \tag{1}$$

where the f dimensional vector b^i represents a local translation in the target domain and γ is a translation regulator. In order to impose as few assumptions as possible, we shall model the unlabelled target data, $\mathtt{X}^{trg}$ by a mixture of Gaussian probability density functions $p(\mathrm{x}) = \sum_{k=1}^{K} w_k p(\mathrm{x}|\lambda_k)$ whose parameters are denoted by $\lambda = \{w_k, \mu_k, \Sigma_k, k = 1, \cdots, K\}$ where w_k, μ_k and Σ_k denote the weight, mean and covariance matrix of Gaussian component k respectively, K denotes the number of Gaussians and $p(\mathrm{x}|\lambda_k) = \mathcal{N}(\mu_k, \Sigma_k)$.

We formulate the problem of finding an optimal translation parameter b^i as one of maximising the likelihood of the translated source sample measured in the target domain. Under the assumption of x^i being independent and identically distributed, the likelihood of a source sample after transformation can be written as a weighted sum of the translated source sample posteriors given the target GMM. We wish to maximise:

$$P(G_i^2(\mathrm{x}^{src})|\lambda_{trg}) = \prod_{k=1}^{K} P(G_i^2(\mathrm{x}^{src})|\lambda_k) = \prod_{k=1}^{K} \frac{p(G_i^2(\mathrm{x}^{src})|\lambda_k) w_k}{\sum_{k=1}^{K} w_k p(G_i^2(\mathrm{x}^{src})|\lambda_k)}, \tag{2}$$

or more conveniently, its natural logarithm, with respect to the unknown parameter b^i. Setting the gradient of the log likelihood, $\mathcal{L}(\lambda_{trg}|\mathrm{x}^{src})$ with respect to b^i to zero, we get a relationship between the translation vector b^i and the Gaussian component parameters

$$\mathrm{b}^i = \frac{\sum_{k=1}^{K} P(\mathrm{x}^i + \mathrm{b}_0^i|\lambda_k)\Sigma_k^{-1}(\mathrm{x}^i - \mu_k)}{\sum_{k=1}^{K} P(\mathrm{x}^i + \mathrm{b}_0^i|\lambda_k)\Sigma_k^{-1}} \,, \tag{3}$$

where b_0^i is an initial value of b^i, which is set to a vector of zeros. In our experiments, we ran (3) only once, though one can iterate it further.[2]

In practice, equation 1 translates x_i^{src} using the combination of the translations between x_i^{src} and μ_k, weighted by the likelihood of $G_i(\mathrm{x}_i^{src})$ given λ_k.

3.3 Conditional distribution adaptation with TST

In order to adapt the class-conditional distribution mismatch between the corresponding clusters of the two domains, we followed Farajidavar et al. [13]. The authors proposed a Translation+Scaling transformation (TST) which assumes that a Gaussian Mixture Model fitted to the source classes can be adapted in a way that it matches the target classes. TST adaptation

[2] Full details of the derivations that lead to the equations above are described in a manuscript under review. Please check the authors' websites for an upcoming publication with these details.

is introduced by means of a class-based transformation $G^3_{y^i}(\mathbf{X})$ which aims to adjust the mean and standard deviation of the corresponding clusters from the source domain.

Matching the marginal distributions does not guarantee that the conditional distribution of the target can be approximated to that of the source. To our knowledge, most of the recent works related to this issue [6, 9, 25, 33] are Inductive TL methods and they have access to some labelled data in the target domain which in practice facilitates the posteriors' estimations. Instead, the TST method of [13] reduces the difference between the likelihoods $P(G^3_y(\mathrm{x}^{src})|y=c)$ and $P(\mathrm{x}^{trg}|y=c)$. These approximations will not be reliable unless we iterate over the whole distribution adaptation step and retrain the classifier model using the adapted source samples.

3.4 Stopping criterion

In order to automatically control the number of the iterations in our pipeline, we introduce a domain dissimilarity measure inspired by sample selection bias correction techniques [11, 27]. Many of those techniques are based on weighting samples x^{src}_i using the ratio $w(\mathrm{x}^{src}_i) = P(\mathrm{x}^{trg}_i)/P(\mathrm{x}^{src}_i)$. This ratio can be estimated using a classifier that is trained to distinguish between source and target domains, i.e., samples are labelled as either belonging to class *src* or *trg*. Based on this idea, we use this classification performance as a measure of dissimilarity between two domains, i.e., if it is easy to distinguish between source and target samples, it means they are dissimilar. We coin this measure as **Global Dissimilarity**, $D^{\text{global}}(\mathbf{X}^{src}, X^{trg})$. The intuition is that if the domain dissimilarity is high, then more iterations should be needed to achieve a better match between the domains.

3.5 Classifier selection and model adaptation

We do not assume that source and target domain samples follow the same distribution, so the best performing learner for the source set may not be the best for the target set. We propose to use dissimilarity measures between source and target sets in order to select the classifier and adjust its kernel parameter. Empirical results showed that the optimisation of SVM using grid search on the parameter space with cross-validation on the training set leads to overfitting. We therefore prefer to use Kernel LDA (KDA) [7] and PCA+NN classifiers as the main learners.

To select between these classifiers and to adapt the KDA kernel lengthscale parameter, we propose to use two measures. The first is the **Global Dissimilarity** between the source and target distributions, described in Section 3.4. The second measure, coined **Clusters Dissimilarity** ($D^{\text{clusters}}(\mathbf{X}^{src}, \mathbf{X}^{trg})$), is proportional to the average dissimilarity between the source and target clusters, computed using the average of the distances between the source class centers and their nearest target cluster center. The target clusters centers are obtained using K-means on the target data, initialised using source class centers. We therefore assume that there is no shuffle in the placement of the clusters from one domain to another. Table 3 shows these two measures computed on all datasets.

When both dissimilarity measures indicate that the cross-domain datasets are very different, we suggest that it is better to use a non-parametric classifier, like Nearest Neighbour, so no optimisation is employed at training. When the two domains are similar at global levels, it is sensible to use a classifier such as KDA, whose parameters optimised on the source domain have a better chance of working on the target space. For those cases, we propose to adapt the lengthscale σ of the RBF kernel of KDA using a linear function of the cluster

dissimilarity measure. Following the common practice in the vision community (e.g. [32]), we initially set

$$\sigma = \frac{1}{n_{src}^2} \sum_{i,j}^{n_{src}} |\mathrm{x}_i - \mathrm{x}_j|_1, \forall \mathrm{x}_i, \mathrm{x}_j \in \mathrm{X}^{src} \tag{4}$$

(we used ℓ^1 norm in the kernel function). This is then adapted using

$$\sigma' = \sigma \times \frac{const}{D^{\text{clusters}}(\mathrm{X}^{src}, \mathrm{X}^{trg})}, \tag{5}$$

where *const* is empirically set to be the average cluster dissimilarity obtained in a set of cross-domain comparisons. This was devised based on the fact that the credibility of a classifier is inversely proportional to the dissimilarity between trainining and test samples. In the case of KDA, the best way to tune its generalisation ability is via the kernel lengthscale.

Note that the clusters dissimilarity measure can only be computed if enough samples are available in both source and target sets or if they are not too unbalanced. When these conditions are not satisfied, our algorithm avoids kernel-based method and selects the nearest neighbour classifier.

3.6 The TTM algorithm and its computation complexity

The proposed method is described in algorithm 1. Its computational cost is as follows, where n is the size of the dataset, f is its dimensionality and K is the number of GMM components: (1) MMD: $O(n^2)$ for constructing the MMD matrix, $O(nf^2)$ for covariance computation and $O(f^3)$ for eigendecomposition; (2) TransGrad: $O(nK)$ for the Expectation step of GMM computation, $O(nKf^2)$ for the computation of covariance matrices and $O(K)$ for the Maximization step of the GMM computation. Once the GMM is built, the TransGrad transformation itself is $O(nK)$; (3) TST: $O(Kf)$ for class specific TST transformations and (4) Classification: zero for training NN classifier and $O(n^2)$ for KDA.

For each iteration, the classifier is re-applied and TST is computed. Therefore, the overall complexity of our training algorithm is dominated by the cost of training a GMM (which is low by using diagonal covariances) and by the cost of iteratively training and applying a classifier. The core transformations proposed in this paper, TransGrand and TST are $O(nKf)$ and $O(nf)$, respectively, i.e., much cheaper than most methods in the literature.

Algorithm 1 ATTM: Adaptive Transductive Transfer Machine

Input: $\mathrm{X}^{src}, \mathrm{Y}^{src}, \mathrm{X}^{trg}$
Output: Y^{trg}
1. Search for the shared subspace between the two domains (Sec. 3.1)
2. Adjust the marginal distribution mismatch between the two domains (Sec. 3.2)
3. Select the appropriate classifier (Sec. 3.5), if it is kernel-based, tune σ using (5)
while $T < 10$ and $|D^{\text{global}}(G^t(\mathrm{X}^{src}), \mathrm{X}^{trg})| > threshold$ **do**
   4. Find the feature-wise TST transformation (Sec. 3.3)
   5. Transform the source domain clusters
   6. Retrain the classifier using the transformed source
end while

4 Experimental evaluation

We used four benchmark datasets that are widely adopted to evaluate computer vision and transfer learning algorithms: USPS, MNIST, COIL20 and Caltech+office.

USPS and MNIST datasets have very different marginal distributions but they share 10 classes of digits. We follow the settings of [20] for USPS→MNIST and MNIST→USPS experiments.

COIL20 contains 20 objects classes with 1,440 images. The images of each object were taken 5 degrees apart as the object is rotated on a turntable and each object has 72 images. Each image is 32×32 pixels with 256 gray levels. In our experiments, we follow the settings of [20] and partition the dataset into two subsets. (1): COIL1 contains all images taken with objects in the orientations of $[0°, 85°] \cup [180°, 265°]$ (quadrants 1 and 3); (2) COIL2 contains all images taken in the orientations of $[90°, 175°] \cup [270°, 355°]$ (quadrants 2 and 4).

CALTECH+OFFICE is composed of a 10-class sampling of four datasets; Amazon (images downloaded from online merchants), Webcam (low-resolution images by web camera), DSLR (high-resolution images by a digital SLR camera) and Calthech-256. For the settings we followed [9, 14]. Each dataset is assumed as a different domain and there are between 8 and 151 samples per category per domain, and 2533 images in total.

Table 2: Classifiers' evaluations on individual domains: 5-fold cross validation accuracy of a nearest neighbour classifier. All the datasets are l_2-normalized.

Classifier	MNIST	USPS	COIL1	COIL2	Caltech	Amazon	Webcam	DSLR
PCA+NN	91.97	93.64	99.02	98.91	38.80	60.59	79.58	**76.95**
LR	86.15	89.22	92.36	92.22	56.27	72.46	80.01	67.49
KDA	**94.05**	94.84	**100.00**	**99.71**	**58.16**	**78.73**	**89.54**	63.94
SVM	91.80	**95.28**	99.72	99.44	57.17	74.86	86.44	75.80

We have evaluated the performance of a set of widely used classifiers on all the datasets based on a 5-fold cross validation mean accuracy measure. In the case of applying the NN classifier we further projected our full space into its principal components (PCA), retaining 90% of the energy. The results are presented in Table 2. As one can note in most of the experiments KDA is the winning classifier. SVM[3] is also a strong learner but it requires optimisation of parameters C and σ, which can make it optimal for the source domain, but not necessarily for the target. It is worth noting that PCA+NN's performance is remarkably close to that of KDA on the first two datasets and it is even superior on the DSLR dataset.

The two cross-domain dissimilarity measures from Sec. 3.4 are shown in Table 3, where the datasets are abbreviated as M: MNIST, U: USPS, C: Caltech, A: Amazon, W: Webcam, and D: DSLR.

4.1 Experiments and Results

We coin the iterative version of different combinations of our proposed algorithms as Transductive Transfer Machine (TTM). TTM0 refers to an iterative version of TST adaptations, TTM1 is the combination of the MMD and TST and finally TTM2 is TTM1 with a sample-wise marginal adaptation (TransGrad) applied before TST. We have also carried out experiments to show that our proposed classifier selection and model adaptation techniques (ATTM) improve the performance of both TTM and JDA algorithms significantly. We

[3] The LIBSVM library is used for SVM classification.

compared our methods with two state-of-the-art approaches [14, 20] using the same public datasets and the same settings as theirs. The results are presented in Table 3. Further comparisons with other TTL methods such as Transfer Component Analysis [24], Transfer Subspace Learning [29] and Sampling Geodesic Flow (SGF) using the Grassmann manifolds [15] are reported in [14, 20].

Table 3: Dissimilarity measures and recognition accuracies with datasets abbreviated as M: MNIST, U: USPS, C: Caltech, A: Amazon, W: Webcam, and D: DSLR. Comparisons start with columns 2 and 3 demonstrating the cross-domain dissimilarities and then the baseline accuracy with NN followed by the results of the discussed TTL algorithms. The last two columns show the effect of the classifier selection and model adaptation techniques (3.5) on JDA and TTM algorithms.

TTL test	Cluster diss.	Global diss.	NN base-line	GFK (PLS, PCA) [14]	JDA (1NN) [20]	TTM0 (TST NN)	TTM1 (MMD + TTM0)	TTM2 (Trans-Grad + TTM1)	AJDA (Adapt. JDA)	ATTM (Adapt. TTM2)
M→ U	0.034	0.984	65.94	67.22	67.28	75.94	76.61	77.94	67.28	**77.94**
U→M	0.032	0.981	44.70	46.45	59.65	59.79	59.41	61.15	59.65	**61.15**
COIL1→2	0.026	0.627	83.61	72.50	89.31	88.89	88.75	93.19	**94.31**	92.64
COIL2→1	0.025	0.556	82.78	74.17	88.47	88.89	88.61	88.75	**92.36**	91.11
C→ A	0.032	0.548	23.70	41.4	44.78	39.87	44.25	46.76	58.56	**60.85**
C→ D	0.031	0.786	25.48	41.1	45.22	50.31	44.58	47.13	45.86	**50.32**
A→ C	0.035	0.604	26.00	37.9	39.36	36.24	35.53	39.62	40.43	**42.92**
A→ W	0.035	0.743	29.83	35.7	37.97	37.63	42.37	39.32	49.83	**50.51**
W→ C	0.037	0.752	19.86	29.3	31.17	26.99	29.83	30.36	**35.80**	34.02
W→ A	0.035	0.717	22.96	35.5	32.78	29.12	30.69	31.11	38.94	**39.67**
D→ A	0.034	0.790	28.50	36.1	33.09	31.21	29.75	30.27	37.47	**38.73**
D→ W	0.033	0.471	63.39	79.1	89.49	85.08	**90.84**	88.81	89.49	88.81
Average	-	-	43.06	50.00	54.88	54.12	55.10	56.20	59.17	**60.72**

As one can note, all the TTL methods improve the accuracy over the baseline. Furthermore, our ATTM method generally outperforms all the other methods. The main reason for that is that our method combines three different feature adaptation techniques with a further classifier parameter adaptation step.

In most of the tasks, both TTM1,2 algorithms show comparative performance with respect to the state-of-the-art approach of JDA [20]. The average performance accuracy of the TTM1 and TTM2 on 12 transfer tasks is **55.10%** and **56.20%** respectively. The performance improved by **0.22%** and **1.32%** compared to the best performing baseline method JDA [14]. Moreover in almost all datasets, TTM2 wins over TTM1 due to its initial domain dissimilarity adjustments using the TransGrad. The JDA method of Long et al. [20] also benefits from jointly adapting the marginal and conditional distributions but their approach has the global and class specific adaptations along each other at each iteration which in practice might cancel their respective effects, limiting the final model from being well fitted to the target clusters. While in JDA [14] the number of iterations is fixed to 10, in our algorithm we based this number on a sensible measure of domain dissimilarity described earlier. Moreover, the proposed TTM guarantees an acceptable level of performance about five times faster than the best performing state-of-the-art approach. GFK performs well on some of the Office+Caltech experiments but poorly on the others. The reason is that the subspace dimension should be small enough to ensure that different sub-spaces transit smoothly along the geodesic flow, which may not be an accurate representation of the input data. JDA and TTM perform much better by learning a more accurate shared space.

The effect of the proposed classifier selection and model adaptation techniques is also apparent from Table 3. We have tested our proposed methodologies on both JDA and TTM

algorithms as AJDA and ATTM. The AJDA performance shows that the model adaptation drastically enhances the final classifier. One should note that in the cases where our model adaptation technique selects the NN classifier as the main learner of the algorithm, the results remain steady. The performance gains of **4.59** and **4.29** in ATTM and AJDA respectively validates the proposed dissimilarity measures for model selection and adaptation.

Our method selected Nearest Neighbour for MNIST↔USPS and for DSLR→Webcam. For all other transfer problems, KDA was chosen and σ adaptation was used.

We have also compared the time complexity of our TTM algorithm against JDA [20] in MNIST to USPS transfer task. Both algorithms were implemented in Matlab, and were evaluated on a Intel Core2 64bit, 3GHz machine running Linux. We averaged time measurements of 5 experiments. The JDA algorithm took 21.38 ± 0.26s, whereas our full TTM framework took 4.42 ± 0.12s, broken down as: 0.40 ± 0.01s for MMD, 1.90 ± 0.06s for TransGrad and 2.42 ± 0.12s for TST (including all iterations used).

5 Conclusions

In this paper, we proposed an approach called Adaptive Transductive Transfer Machine (ATTM), which adapts both the marginal and conditional distributions of the source samples so that they become more similar to those of target samples for a problem in which labeled data is only available in the source domain. This leads to an improvement in the classification results in transfer learning scenarios. Furthermore, we proposed to automatically select between two classifiers, one that does not require any tunning (Nearest Neighbour) and a kernel-based method (KDA). When a kernel method is chosen, it automatically tunes its parameter (lengthscale) based on the dissimilarity between source and target sets. In addition, we evaluated this classifier selection and adaptation method with JDA, another state-of-the-art transfer learning method. This also lead to performance gain.

It is worth pointing out that ATTM is a general framework with applicability beyond image classification and could be easily applied to other domains, even outside Computer Vision. For future work, we suggest studying combinations of our method with instance reweighting methods, feature learning algorithms and multi-source transfer learning.

Acknowledgements

We are grateful for the support of the following grants from the EPSRC-UK: *Adaptive Cognition for Automated Sports Video Annotation* (ACASVA, EP/F069421/1), *Signal Processing Solutions for the Networked Battlespace* (EP/K014307/1, with DSL) and *Future Spatial Audio for an Immersive Listener Experience at Home* (S3A, EP/L000539/1).

References

[1] A. Arnold, R. Nallapati, and W. W. Cohen. A comparative study of methods for transductive transfer learning. In *Proceedings of the Seventh IEEE International Conference on Data Mining Workshops*, ICDMW, pages 77–82, Washington, DC, USA, 2007. IEEE Computer Society. ISBN 0-7695-3033-8. URL http://dx.doi.org/10.1109/ICDMW.2007.2.

[2] Y. Aytar and A. Zisserman. Tabula rasa: Model transfer for object category detection. In *IEEE International Conference on Computer Vision*, 2011.

[3] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *Comput. Vis. Image Underst.*, 110(3):346–359, 2008. ISSN 1077-3142.

[4] K. M. Borgwardt, A. Gretton, M. J. Rasch, H. Kriegel, B. Schalkopf, and A. J. Smola. Integrating structured biological data by kernel maximum mean discrepancy. In *IN ISMB*, 2006.

[5] S. Branson, G. Horn, C. Wah, P. Perona, and S. Belongie. The ignorant led by the blind: A hybrid human-machine vision system for fine-grained categorization. *International Journal of Computer Vision*, pages 1–27, 2014. ISSN 0920-5691. URL http://dx.doi.org/10.1007/s11263-014-0698-4.

[6] L. Bruzzone and M. Marconcini. Domain adaptation problems: A dasvm classification technique and a circular validation strategy. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(5):770–787, 2010.

[7] D. Cai, X. He, and J. Han. Efficient kernel discriminant analysis via spectral regression. In *International Conference on Data Mining*, 2007.

[8] S. Chang. Robust visual domain adaptation with low-rank reconstruction. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, pages 2168–2175, Washington, DC, USA, 2012. IEEE Computer Society. ISBN 978-1-4673-1226-4. URL http://dl.acm.org/citation.cfm?id=2354409.2355115.

[9] M. Chen, K. Q. Weinberger, and J. Blitzer. Co-training for domain adaptation. In *NIPS*, pages 2456–2464, 2011. URL http://dblp.uni-trier.de/db/conf/nips/nips2011.html#ChenWB11.

[10] W. S. Chu, F. De la Torre, and J. F. Cohn. Selective transfer machine for personalized facial action unit detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[11] C. Cortes, M. Mohri, M. Riley, and A. Rostamizadeh. Sample selection bias correction theory. In *Proceedings of the 19th international conference on Algorithmic Learning Theory*, ALT '08, pages 38–53, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 978-3-540-87986-2. URL http://dx.doi.org/10.1007/978-3-540-87987-9_8.

[12] W. Dai, Y. Chen, G. Xue, Q. Yang, and Y. Yu. Translated learning: Transfer learning across different feature spaces. In *Neural Information Processing Systems*, pages 353–360, 2008.

[13] N. FarajiDavar, T. deCampos, J. Kittler, and F. Yang. Transductive transfer learning for action recognition in tennis games. In *ICCV,VECTaR workshop*, 2011.

[14] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, pages 2066–2073. IEEE, 2012. ISBN 978-1-4673-1226-4. URL http://dblp.uni-trier.de/db/conf/cvpr/cvpr2012.html#GongSSG12.

[15] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In Dimitris N. Metaxas, Long Quan, Alberto Sanfeliu, and Luc J. Van Gool, editors, *ICCV*, pages 999–1006. IEEE, 2011. ISBN 978-1-4577-1101-5. URL http://dblp.uni-trier.de/db/conf/iccv/iccv2011.html#GopalanLC11.

[16] R. Gopalan, R. Li, and R. Chellappa. Unsupervised adaptation across domain shifts by generating intermediate data representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014. DOI 10.1109/TPAMI.2013.249.

[17] A. Gretton, K. Borgwardt, M. Rasch, B. Scholkopf, and A. Smola. A kernel method for the two sample problem. In *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 19*, pages 513–520. MIT Press, 2007.

[18] A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Schölkopf. Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3(4):5, 2009.

[19] T. Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the Sixteenth International Conference on Machine Learning*, ICML, pages 200–209, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. ISBN 1-55860-612-2.

[20] M. Long, J. Wang, G. Ding, and P. Yu. Transfer learning with joint distribution adaptation. In *IEEE International Conference on Computer Vision (ICCV)*, 2013.

[21] J. Ni, Q. Qiu, and R. Chellappa. Subspace interpolation via dictionary learning for unsupervised domain adaptation. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 692–699. IEEE, 2013. URL http://dblp.uni-trier.de/db/conf/cvpr/cvpr2013.html#NiQC13.

[22] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22:1345–1359, 2010. ISSN 1041-4347.

[23] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. In *Proceedings of the 21st international jont conference on Artifical intelligence*, pages 1187–1192, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc.

[24] S. J. Pan, Ivor W. Tsang, James T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2011.

[25] B. Quanz, J. Huan, and M. Mishra. Knowledge transfer with low-quality data: A feature extraction issue. In S. Abiteboul, K. Bolhm, Ch. Koch, and K. Tan, editors, *ICDE*, pages 769–779. IEEE Computer Society, 2011. ISBN 978-1-4244-8958-9. URL http://dblp.uni-trier.de/db/conf/icde/icde2011.html#QuanzHM11.

[26] M. Shao, D. Kit, and Y. Fu. Generalized transfer subspace learning through low-rank constraint. *International Journal of Computer Vision*, 109(1-2):74–93, 2014.

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

[28] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *Proc of the IEEE Conf on Computer Vision and Pattern Recognition*, 2011.

[29] Si Si, Dacheng Tao, and Bo Geng. Bregman divergence-based regularization for transfer subspace learning. *IEEE Trans. Knowl. Data Eng.*, 22(7):929–942, 2010. URL http://dblp.uni-trier.de/db/journals/tkde/tkde22.html#SiTG10.

[30] Q. Sun, R. Chattopadhyay, S. Panchanathan, and J. Ye. A two-stage weighting framework for multi-source domain adaptation. In *NIPS*, pages 505–513, 2011. URL http://dblp.uni-trier.de/db/conf/nips/nips2011.html#SunCPY11.

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

[32] H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *BMVC2009*, 2009.

[33] E. Zhong, W. Fan, J. Peng, K. Zhang, J. Ren, D. S. Turaga, and O. Verscheure. Cross domain distribution adaptation via kernel mapping. In *KDD*, pages 1027–1036. ACM, 2009. ISBN 978-1-60558-495-9.