The AAAI-17 Workshop on
Knowledge-Based Techniques for Problem Solving and Reasoning
WS-17-12

Model Selection with Nonlinear Embedding for Unsupervised Domain Adaptation

Hemanth Venkateswara, Shayok Chakraborty,
Troy McDaniel, Sethuraman Panchanathan
Center for Cognitive Ubiquitous Computing, Arizona State University, Tempe, AZ, USA
{hemanthv, shayok.chakraborty, troy.mcdaniel, panch}@asu.edu

Abstract

Domain adaptation deals with adapting classifiers trained on data from a source distribution, to work effectively on data from a target distribution. In this paper, we introduce the Nonlinear Embedding Transform (NET) for unsupervised domain adaptation. The NET reduces cross-domain disparity through nonlinear domain alignment. It also embeds the domain-aligned data such that similar data points are clustered together. This results in enhanced classification. To determine the parameters in the NET model (and in other unsupervised domain adaptation models), we introduce a validation procedure by sampling source data points that are similar in distribution to the target data. We test the NET and the validation procedure using popular image datasets and compare the classification results across competitive procedures for unsupervised domain adaptation.

Introduction

There are large volumes of unlabeled data available online, owing to the exponential increase in the number of images and videos uploaded online. It would be easy to obtain labeled data if trained classifiers could predict the labels for unlabeled data. However, classifier models do not perform well when applied to unlabeled data from different distributions, owing to domain-shift (Torralba and Efros 2011). Domain adaptation deals with adapting classifiers trained on data from a source distribution, to work effectively on data from a target distribution (Pan and Yang 2010). Some domain adaptation techniques assume the presence of a few labels for the target data, to assist in training a domain adaptive classifier (Aytar and Zisserman 2011; Duan, Tsang, and Xu 2012; Hoffman et al. 2013). However, real world applications need not support labeled data in the target domain and adaptation here is termed as unsupervised domain adaptation.

Many of the unsupervised domain adaptation techniques can be organized into *linear* and *nonlinear* procedures, based on how the data is handled by the domain adaptation model. A *linear* domain adaptation model performs linear transformations on the data to align the source and target domains or, it trains an adaptive linear classifier for both the domains; for example a linear SVM (Bruzzone and Marconcini 2010). *Nonlinear* techniques are deployed in situations where the source and target domains cannot be aligned using linear transformations. These techniques apply nonlinear transformations on the source and target data in order to align them. For example, Maximum Mean Discrepancy (MMD) is applied to learn nonlinear representations, where the difference between the source and target distributions is minimized (Pan et al. 2011). Even though nonlinear transformations may align the domains, the resulting data may not be conducive to classification. If, after domain alignment, the data were to be clustered based on similarity, it can lead to effective classification. We demonstrate this intuition through a binary classification problem using a toy dataset. Figure (1a), displays the source and target domains of a two-moon dataset. Figure (1b), depicts the transformed data after KPCA (nonlinear projection). In trying to project the data onto a common 'subspace', the source data gets dispersed. Figure (1c), presents the data after domain alignment using Maximum Mean Discrepancy (MMD). Although

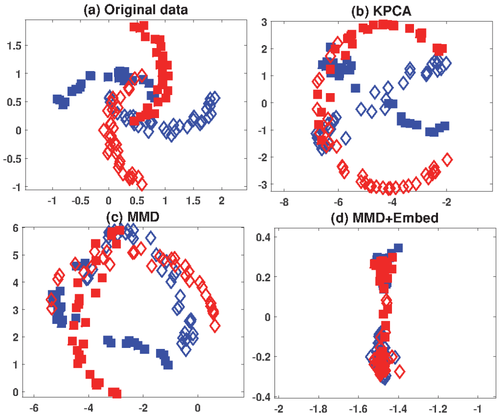


Figure 1: (Best viewed in color) Two-moon binary classification problem with source data in blue and target data in red. We assume the target labels are unknown. (a) Original data, (b) KPCA aligns the data along nonlinear directions of maximum variance, (c) MMD aligns the two domains, (d) MMD+Similarity-based Embedding aligns the domains and clusters the data to ensure easy classification.

the domains are now aligned, it does not necessarily ensure enhanced classification. Figure (1d), shows the data after MMD and similarity-based embedding, where data is clustered based on class label similarity. Cross-domain alignment along with similarity-based embedding, makes the data classification friendly.

In this work, we the present the Nonlinear Embedding Transform (NET) procedure for unsupervised domain adaptation. The NET performs a nonlinear transformation to align the source and target domains and also cluster the data based on label-similarity. The NET algorithm is a spectral (eigen) technique that requires certain parameters (like number of eigen bases, etc.) to be pre-determined. These parameters are often given random values which need not be optimal (Pan et al. 2011; Long et al. 2013; 2014). In this work, we also outline a validation procedure to fine-tune model parameters with a validation set created from the source data. In the following, we outline the two main contributions in our work:

- Nonlinear embedding transform (NET) algorithm for unsupervised domain adaptation.
- Validation procedure to estimate optimal parameters for an unsupervised domain adaptation algorithm.

We evaluate the validation procedure and the NET algorithm using 7 popular domain adaptation image datasets, including object, face, facial expression and digit recognition datasets. We conduct 50 different domain adaptation experiments to compare the proposed techniques with existing competitive procedures for unsupervised domain adaptation.

Related Work

For the purpose of this paper, we discuss the relevant literature under the categories *linear* domain adaptation methods and *nonlinear* domain adaptation methods. A detailed survey on transfer learning procedures can be found in (Pan and Yang 2010). A survey of domain adaptation techniques for vision data is provided by (Patel et al. 2015).

The Domain Adaptive SVM (DASVM) (Bruzzone and Marconcini 2010), is an unsupervised method that iteratively adapts a linear SVM from the source to the target. In recent years, the popular unsupervised linear domain adaptation procedures are Subspace Alignment (SA) (Fernando et al. 2013), and the Correlation Alignment (CA) (Sun, Feng, and Saenko 2015). The SA algorithm determines a linear transformation to project the source and target to a common subspace, where the domain disparity is minimized. The CA is an interesting technique which argues that aligning the correlation matrices of the source and target data is sufficient to reduce domain disparity. Both the SA and CA are linear procedures, whereas the NET is a nonlinear method.

Although deep learning procedures are inherently highly nonlinear, we limit the scope of our work to nonlinear transformation of data that usually involves a positive semi-definite kernel function. Such procedures are closely related to the NET. However, in our experiments, we do study the NET with deep features also. The Geodesic Flow Kernel (GFK) (Gong et al. 2012), is a popular domain adaptation method, where the subspace spanning the source data is gradually transformed into the target subspace along a path on the Grassmann manifold of subspaces. Spectral procedures like the Transfer Component Analysis (TCA) (Pan et al. 2011), the Joint Distribution Alignment (JDA) (Long et al. 2013) and Transfer Joint Matching (TJM) (Long et al. 2014), are the most closely related techniques to the NET. All of these procedures involve a solution to a generalized eigen-value problem in order to determine a projection matrix to nonlinearly align the source and target data. In these spectral methods, domain alignment is implemented using variants of MMD, which was first introduced in the TCA procedure. JDA introduces joint distribution alignment which is an improvement over TCA that only incorporates marginal distribution alignment. The TJM performs domain alignment along with instance selection by sampling only relevant source data points. In addition to domain alignment with MMD, the NET algorithm implements similarity-based embedding for enhanced classification. We also introduce a validation procedure to estimate the model parameters for unsupervised domain adaptation approaches.

Domain Adaptation With Nonlinear Embedding

In this section, we first outline the NET algorithm for unsupervised domain adaptation. We then describe a cross-validation procedure that is used to estimate the model parameters for the NET algorithm.

We begin with the problem definition where we consider two domains; source domain $\mathscr{S}$ and target domain $\mathscr{T}$. Let $\mathcal{D}_s = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{n_s} \subset \mathscr{S}$ be a subset of the source domain and $\mathcal{D}_t = \{(\mathbf{x}_i^t, y_i^t)\}_{i=1}^{n_t} \subset \mathscr{T}$ be the subset of the target domain. Let $\mathbf{X}_S = [\mathbf{x}_1^s, \ldots, \mathbf{x}_{n_s}^s] \in \mathbb{R}^{d \times n_s}$ and $\mathbf{X}_T = [\mathbf{x}_1^t, \ldots, \mathbf{x}_{n_t}^t] \in \mathbb{R}^{d \times n_t}$ be the source and target data points respectively. Let $Y_S = [y_1^s, \ldots, y_{n_s}^s]$ and $Y_T = [y_1^t, \ldots, y_{n_t}^t]$ be the source and target labels respectively. Here, $\mathbf{x}_i^s$ and $\mathbf{x}_i^t$ $\in \mathbb{R}^d$ are data points and y_i^s and $y_i^t \in \{1, \ldots, C\}$ are the associated labels. We define $\mathbf{X} := [\mathbf{x}_1, \ldots, \mathbf{x}_n] = [\mathbf{X}_S, \mathbf{X}_T]$, where $n = n_s + n_t$. The problem of domain adaptation deals with the situation where the joint distributions for the source and target domains are different, i.e. $P_S(X, Y) \neq P_T(X, Y)$, where X and Y denote random variables for data points and labels respectively. In the case of unsupervised domain adaptation, the labels Y_T are unknown. The goal of unsupervised domain adaptation is to estimate the labels of the target data $\hat{Y}_T = [\hat{y}_1^t, \ldots, \hat{y}_{n_t}^t]$ corresponding to $\mathbf{X}_T$ using $\mathcal{D}_s$ and $\mathbf{X}_T$.

Nonlinear Domain Alignment

A common procedure to align two datasets is to first project them to a common subspace. Kernel-PCA (KPCA) estimates a nonlinear basis for such a projection. In this case, data is internally mapped to a high-dimensional (possibly infinite-dimensional) space defined by $\Phi(\mathbf{X}) = [\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_n)]$. $\phi : \mathbb{R}^d \rightarrow \mathcal{H}$ is the mapping function and $\mathcal{H}$ is a RKHS (Reproducing Kernel Hilbert Space). The dot product between the mapped vectors $\phi(\mathbf{x})$ and $\phi(\mathbf{y})$, is estimated by a positive semi-definite (psd) kernel, $k(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x})^\top \phi(\mathbf{y})$. The dot product captures the similarity between $\mathbf{x}$ and $\mathbf{y}$. The kernel similarity (gram) matrix consisting of similarities between all the data points in $\mathbf{X}$, is given by,

$\mathbf{K} = \Phi(\mathbf{X})^\top\Phi(\mathbf{X}) \in \mathbb{R}^{n\times n}$. The matrix $\mathbf{K}$ is used to determine the projection matrix $\mathbf{A}$, by solving,

$$\max_{\mathbf{A}^\top\mathbf{A}=\mathbf{I}} \operatorname{tr}(\mathbf{A}^\top\mathbf{KHK}^\top\mathbf{A}). \tag{1}$$

Here, $\mathbf{H}$ is the $n \times n$ centering matrix given by $\mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbf{1}$, where $\mathbf{I}$ is an identity matrix and $\mathbf{1}$ is a $n \times n$ matrix of 1s. $\mathbf{A} \in \mathbb{R}^{n\times k}$, is the matrix of coefficients and the nonlinear projected data is given by $\mathbf{Z} = [\mathbf{z}_1, \dots, \mathbf{z}_n] = \mathbf{A}^\top\mathbf{K} \in \mathbb{R}^{k\times n}$. Along with projecting the source and target data to a common subspace, the domain-disparity between the two datasets must also be reduced. We employ the Maximum Mean Discrepancy (MMD) (Gretton et al. 2009), which is a standard nonparametric measure to estimate domain disparity. We adopt the Joint Distribution Adaptation (JDA) (Long et al. 2013), algorithm which seeks to align both the the marginal and conditional probability distributions of the projected data. The marginal distributions are aligned by estimating the coefficient matrix $\mathbf{A}$, which minimizes:

$$\min_{\mathbf{A}} \left|\left|\frac{1}{n_s}\sum_{i=1}^{n_s}\mathbf{A}^\top\mathbf{k}_i - \frac{1}{n_t}\sum_{j=n_s+1}^{n}\mathbf{A}^\top\mathbf{k}_j\right|\right|_{\mathcal{H}}^2 = \operatorname{tr}(\mathbf{A}^\top\mathbf{KM}_0\mathbf{K}^\top\mathbf{A}). \tag{2}$$

$\mathbf{M}_0$, is the MMD matrix which given by,

$$(\mathbf{M}_0)_{ij} = \begin{cases} \frac{1}{n_s n_s}, & \mathbf{x}_i, \mathbf{x}_j \in \mathcal{D}_s \\ \frac{1}{n_t n_t}, & \mathbf{x}_i, \mathbf{x}_j \in \mathcal{D}_t \\ \frac{-1}{n_s n_t}, & \text{otherwise,} \end{cases} \tag{3}$$

Likewise, the conditional distribution difference can also be minimized by introducing matrices M_c, with $c = 1, \dots, C$, defined as,

$$(\mathbf{M}_c)_{ij} = \begin{cases} \frac{1}{n_s^{(c)} n_s^{(c)}}, & \mathbf{x}_i, \mathbf{x}_j \in \mathcal{D}_s^{(c)} \\ \frac{1}{n_t^{(c)} n_t^{(c)}}, & \mathbf{x}_i, \mathbf{x}_j \in \mathcal{D}_t^{(c)} \\ \frac{-1}{n_s^{(c)} n_t^{(c)}}, & \begin{cases} \mathbf{x}_i \in \mathcal{D}_s^{(c)}, \mathbf{x}_j \in \mathcal{D}_t^{(c)} \\ \mathbf{x}_j \in \mathcal{D}_s^{(c)}, \mathbf{x}_i \in \mathcal{D}_t^{(c)} \end{cases} \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

Here, $\mathcal{D}_s$ and $\mathcal{D}_t$ are the sets of source and target data points respectively. $\mathcal{D}_s^{(c)}$ is the subset of source data points with class label c and $n_s^{(c)} = |\mathcal{D}_s^{(c)}|$. Similarly, $\mathcal{D}_t^{(c)}$ is the subset of target data points with class label c and $n_t^{(c)} = |\mathcal{D}_t^{(c)}|$. Since the target labels being unknown, we use predicted target labels to determine $\mathcal{D}_t^{(c)}$. We initialize the target labels using a classifier trained on the source data and refine the labels over iterations. Combining both the marginal and conditional distribution terms leads us to the JDA model, which is given by,

$$\min_{\mathbf{A}} \sum_{c=0}^{C} \operatorname{tr}(\mathbf{A}^\top\mathbf{KM}_c\mathbf{K}^\top\mathbf{A}). \tag{5}$$

Similarity Based Embedding

In addition to domain alignment, the NET algorithm ensures that the projected data $\mathbf{Z}$, is classification friendly (easily classifiable). To this end we introduce laplacian eigenmaps in order to cluster datapoints based on class label similarity. The $(n \times n)$ adjacency matrix $\mathbf{W}$, captures the similarity relationships between datapoints, where,

$$\mathbf{W}_{ij} := \begin{cases} 1 & y_i^s = y_j^s \text{ or } i = j \\ 0 & y_i^s \neq y_j^s \text{ or labels unknown.} \end{cases} \tag{6}$$

To ensure that the projected data is clustered based on data similarity, we minimize the sum of squared distances between data points weighted by the adjacency matrix. This can be expressed as a minimization problem,

$$\min_{\mathbf{Z}} \frac{1}{2}\sum_{ij}\left|\left|\frac{\mathbf{z}_i}{\sqrt{d_i}} - \frac{\mathbf{z}_j}{\sqrt{d_j}}\right|\right|^2 \mathbf{W}_{ij} = \min_{\mathbf{A}} \operatorname{tr}(\mathbf{A}^\top\mathbf{KLK}^\top\mathbf{A}). \tag{7}$$

Here, $d_i = \sum_k \mathbf{W}_{ik}$ and $d_j = \sum_k \mathbf{W}_{jk}$. They form the diagonal entries of $\mathbf{D}$, the $(n\times n)$ diagonal matrix. $||\mathbf{z}_i/\sqrt{d_i} - \mathbf{z}_j/\sqrt{d_j}||^2$, is the squared normalized distance between the projected data points $\mathbf{z}_i$ and $\mathbf{z}_j$, which get clustered together when $\mathbf{W}_{ij} = 1$, (as they belong to the same category). The normalized distance is a more robust clustering measure as compared to the standard Euclidean distance $||\mathbf{z}_i - \mathbf{z}_j||^2$, (Chung 1997). Substituting $\mathbf{Z} = \mathbf{A}^\top\mathbf{K}$, yields the trace term, where $\mathbf{L}$, denotes the symmetric positive semi-definite graph laplacian matrix with $\mathbf{L} := \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$, and $\mathbf{I}$ is an identity matrix.

Optimization Problem

To arrive at the optimization problem, we consider the nonlinear projection in Equation (1), the joint distribution alignment in Equation (5) and the similarity based embedding in Equation (7). Maximizing Equation (1) and minimizing Equations (5) and (7) can also be achieved by maintaining Equation (1) constant and minimizing Equations (5) and (7). Minimizing the similarity embedding in Equation (7) can result in the projected vectors being embedded in a low dimensional subspace. In order to maintain the subspace dimensionality, we introduce a new constraint in place of Equation (1). The optimization problem for NET is obtained by minimizing Equations (5) and (7). The goal is to estimate the $(n \times k)$ projection matrix, $\mathbf{A}$. Along with regularization and the dimensionality constraint, we get,

$$\min_{\mathbf{A}^\top\mathbf{KDK}^\top\mathbf{A}=\mathbf{I}} \alpha.\operatorname{tr}(\mathbf{A}^\top\mathbf{K}\sum_{c=0}^{C}\mathbf{M}_c\mathbf{K}^\top\mathbf{A}) + \beta.\operatorname{tr}(\mathbf{A}^\top\mathbf{KLK}^\top\mathbf{A}) + \gamma||\mathbf{A}||_F^2. \tag{8}$$

The first term controls the domain alignment and is weighted by α. The second term ensures similarity based embedding and is weighted by β. The third term is the regularization (Frobenius norm) that ensures a smooth projection matrix $\mathbf{A}$ and it is weighted by γ. The constraint on $\mathbf{A}$ (in place of $\mathbf{A}^\top\mathbf{KHK}^\top\mathbf{A} = \mathbf{I}$), prevents the projection from collapsing onto a subspace with dimensionality less than k, (Belkin and Niyogi 2003). We solve Equation (8) by forming the Lagrangian,

$$L(\mathbf{A}, \mathbf{\Lambda}) = \alpha.\operatorname{tr}\big(\mathbf{A}^\top\mathbf{K}\sum_{c=0}^{C}\mathbf{M}_c\mathbf{K}^\top\mathbf{A}\big) + \beta.\operatorname{tr}(\mathbf{A}^\top\mathbf{KLK}^\top\mathbf{A}) + \gamma||\mathbf{A}||_F^2 + \operatorname{tr}((\mathbf{I} - \mathbf{A}^\top\mathbf{KDK}^\top\mathbf{A})\mathbf{\Lambda}), \tag{9}$$

Algorithm 1 Nonlinear Embedding Transform

```
Require: $\mathbf{X}$, $Y_S$, constants $\alpha, \beta$, regularization $\gamma$ and projection dimension $k$.
Ensure: Projection matrix $\mathbf{A}$, projected data $\mathbf{Z}$.
Compute kernel matrix $\mathbf{K}$, for predefined kernel $k(.,.)$
Define the adjacency matrix $\mathbf{W}$ (Eq. (6))
Compute $\mathbf{D} = \text{diag}(d_1, \dots, d_n)$, where $d_i = \sum_j \mathbf{W}_{ij}$
Compute normalized graph laplacian $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$
Solve Eq (10) and select $k$ smallest eigen-vectors as columns of $\mathbf{A}$
Estimate $\mathbf{Z} \leftarrow \mathbf{A}^\top \mathbf{K}$
Train a classifier with modified data $\{[\mathbf{z}_1, \dots, \mathbf{z}_{n_s}], Y_S\}$
```

where the Lagrangian constants are represented by the diagonal matrix $\mathbf{\Lambda} = diag(\lambda_1, \dots, \lambda_k)$. Setting the derivative $\frac{\partial L}{\partial \mathbf{A}} = 0$, yields the generalized eigen-value problem,

$$\Big(\alpha \mathbf{K} \sum_{c=0}^{C} \mathbf{M}_c \mathbf{K}^\top + \beta \mathbf{K}\mathbf{L}\mathbf{K}^\top + \gamma \mathbf{I}\Big)\mathbf{A} = \mathbf{K}\mathbf{D}\mathbf{K}^\top \mathbf{A}\mathbf{\Lambda}. \quad (10)$$

The solution $\mathbf{A}$ in (8) are the k-smallest eigen-vectors of Equation (10). The projected data points are then given by $\mathbf{Z} = \mathbf{A}^\top \mathbf{K}$. The NET algorithm is outlined in Algorithm 1.

Model Selection

In unsupervised domain adaptation the target labels are treated as unknown. Current domain adaptation methods that need to validate the optimum parameters for their models, inherently assume the availability of target labels (Long et al. 2013), (Long et al. 2014). However, in the case of real world applications, when target labels are not available, it is difficult to verify if the model parameters are optimal. In the case of the NET model, we have 4 parameters $(\alpha, \beta, \gamma, k)$, that we want to pre-determine. We introduce a technique using Kernel Mean Matching (KMM) to sample the source data to create a validation set. KMM has been used to weight source data points in order to reduce the distribution difference between the source and target data (Fernando et al. 2013), (Gong, Grauman, and Sha 2013). Source data points with large weights have a similar marginal distribution to the target data. These data points are chosen to form the validation set. The KMM estimates the weights $w_i, i = 1, \dots, n_s$, by minimizing $||\frac{1}{n_s}\sum_{i=1}^{n_s} w_i \phi(\mathbf{x}_i^s) - \frac{1}{n_t}\sum_{j=1}^{n_t} \phi(\mathbf{x}_j^t)||_{\mathcal{H}}^2$. In order to simplify, we define $\kappa_i := \frac{n_s}{n_t}\sum_{j=1}^{n_t} k(\mathbf{x}_i^s, \mathbf{x}_j^t)$, $i = 1, \dots, n_s$ and $\mathbf{K}_{S_{ij}} = k(\mathbf{x}_i^s, \mathbf{x}_j^s)$. The minimization is then represented as a quadratic programming problem,

$$\min_{\mathbf{w}} = \frac{1}{2}\mathbf{w}^\top \mathbf{K}_S \mathbf{w} - \kappa^\top \mathbf{w},$$
$$\text{s.t. } w_i \in [0, B], \ \Big|\sum_{i=1}^{n_s} w_i - n_s\Big| \leq n_s \epsilon. \quad (11)$$

The first constraint limits the scope of discrepancy between source and target distributions, with $B \to 1$, leading to an unweighted solution. The second constraint ensures the measure $w(x)P_S(x)$, is a probability distribution (Gretton et al. 2009). In our experiments, we select 10% of the source data with the largest weights to create the validation set. We fine tune the values of $(\alpha, \beta, \gamma, k)$, using the validation set. For fixed values of $(\alpha, \beta, \gamma, k)$, the NET model is trained using the source data (without the validation set) and target data. The model is tested on the validation set to estimate parameters yielding highest classification accuracies.

Experiments

In this section, we evaluate the NET algorithm and the model selection proposition across multiple image classification datasets and several existing procedures for unsupervised domain adaptation.

Datasets

We conduct our experiments across 7 different datasets. Their characteristics are outlined in Table (1).

***MNIST-USPS* datasets**: These are popular handwritten digit recognition datasets. Here, the digit images are subsampled to 16×16 pixels. Based on (Long et al. 2014), we consider two domains `MNIST` (2,000 images from MNIST) and `USPS` (1,800 images from USPS).

***CKPlus-MMI* datasets**: The CKPlus (Lucey et al. 2010), and MMI (Pantic et al. 2005) are popular Facial Expression recognition datasets. They contain videos of facial expressions. We choose 6 categories of facial expression, viz., *anger, disgust, fear, happy, sad, surprise*. We create two domains, `CKPlus` and `MMI`, by selecting video frames with the most intense expressions. We use a pre-trained deep convolutional neural network (CNN), to extract features from these images. In our experiments, we use the VGG-F model (Chatfield et al. 2014), trained on the popular ImageNet object recognition dataset. The VGG-F network is similar in architecture to the popular AlexNet (Krizhevsky, Sutskever, and Hinton 2012). We extract the 4096-dimensional features that are fed into the fully-connected $fc8$ layer. We apply PCA on the combined source and target data to reduce the dimension to 500 and use these features across all the experiments.

***COIL20* dataset**: It is an object recognition dataset consisting of 20 categories with two domains, `COIL1` and `COIL2`. The domains consist of images of objects captured from views that are 5 degrees apart. The images are 32×32 pixels with gray scale values (Long et al. 2013).

***PIE* dataset**: The "Pose, Illumination and Expression" (PIE) dataset consists of face images (32×32 pixels) of 68 individuals. The images were captured with different head-pose, illumination and expression. Similar to (Long et al. 2013), we select 5 subsets with differing head-pose to create 5 domains, namely, `P05` (C05, left pose), `P07` (C07, upward pose), `P09` (C09, downward pose), `P27` (C27, frontal pose) and `P29` (C29, right pose).

***Office-Caltech* dataset**: This is currently the most popular benchmark dataset for object recognition in the domain adaptation computer vision community. The dataset consists of images of everyday objects. It consists of 4 domains; `Amazon`, `Dslr` and `Webcam` from the *Office* dataset and `Caltech` domain from the *Caltech-256* dataset. The `Amazon` domain has images downloaded from the www.amazon.com website. The `Dslr` and `Webcam` domains have images captured using a DSLR camera and a webcam respectively. The `Caltech` domain is a subset of the Caltech-256 dataset that was created by selecting categories common with the *Office* dataset. The *Office-Caltech* dataset has 10 categories of objects and a total of 2533 images (data points). We experiment with two kinds of features for the

Office-Caltech dataset; (i) 800-dimensional SURF features (Gong et al. 2012), (ii) Deep features. The deep features are extracted using a pre-trained network similar to the *CKPlus-MMI* datasets.

Table 1: Statistics for the benchmark datasets

Dataset	Type	#Samples	#Features	#Classes	Subsets
MNIST	Digit	2,000	256	10	MNIST
USPS	Digit	1,800	256	10	USPS
CKPlus	Face Exp.	1,496	4096	6	CKPlus
MMI	Face Exp.	1,565	4096	6	MMI
COIL20	Object	1,440	1,024	20	COIL1, COIL2
PIE	Face	11,554	1,024	68	P05, ..., P29
Ofc-Cal SURF	Object	2,533	800	10	A, C, W, D
Ofc-Cal Deep	Object	2,505	4096	10	A, C, W, D

Existing Baselines

We compare the NET algorithm with the following baseline and state-of-the-art methods. Like NET, the TCA, TJM and JDA are all spectral methods. While all the four algorithms use MMD to align the source and target datasets, the NET, in addition, uses nonlinear embedding for classification enhancement. TCA, TJM and JDA, solve for $\mathbf{A}$ in a setting similar to Equation (10). However, unlike NET, they do not have the similarity based embedding term and $\alpha = 1$, is fixed in all the three algorithms. Therefore, these models have only 2 free parameters (γ and k), that need to be pre-determined in contrast to NET, which has 4 parameters, $(\alpha, \beta, \gamma, k)$. Since TCA, TJM and JDA, are all quite similar to each other, for the sake of brevity, we evaluate model selection (estimating optimal model parameters) using only JDA and NET. The SA, CA and GFK algorithms, do not have any critical free model parameters that need to be predetermined.

Table 2: Baseline methods that are compared with the NET.

Method	Reference
SA	Subspace Alignment (Fernando et al. 2013)
CA	Correlation Alignment (Sun, Feng, and Saenko 2015)
GFK	Geodesic Flow Kernel (Gong et al. 2012)
TCA	Transfer Component Analysis (Pan et al. 2011)
TJM	Transfer Joint Matching (Long et al. 2014)
JDA	Joint Distribution Adaptation (Long et al. 2013)

In our experiments, NET$_v$ is a special case of the NET, where model parameters $(\alpha, \beta, \gamma, k)$, have been determined using a validation set derived from Equation (11). Similarly, JDA$_v$ is a special case of JDA, where (γ, k), have been determined using a validation set derived from Equation (11). In order to ascertain the optimal nature of the parameters determined with a source-based validation set, we estimate the best model parameters using the target data (with labels) as a validation set. These results are represented by NET in the figures and tables. The results for the rest of the algorithms (SA, CA, GFK, TCA, TJM and JDA), are obtained with the parameter settings described in their respective works.

Table 3: Recognition accuracies (%) for domain adaptation experiments on the digit and face datasets. {MNIST(M), USPS(U), CKPlus(CK), MMI(MM), COIL1(C1) and COIL2(C2). M→U implies M is source domain and U is target domain. The best and second best results in every experiment (row) are in **bold** and *italic* respectively. The shaded columns indicate accuracies obtained with model selection.

Expt.	SA	CA	GFK	TCA	TJM	JDA	JDA$_v$	NET	NET$_v$
M→U	67.39	59.33	66.06	60.17	64.94	67.28	71.94	**75.39**	*72.72*
U→M	51.85	50.80	47.40	39.85	52.80	59.65	59.65	**62.60**	*61.35*
C1→C2	85.97	84.72	85.00	90.14	91.67	92.64	**95.28**	*93.89*	90.42
C2→C1	84.17	82.78	84.72	88.33	89.86	*93.75*	**93.89**	92.64	88.61
CK→MM	*31.12*	**31.89**	28.75	32.72	30.35	29.78	25.82	29.97	30.54
MM→CK	39.75	37.74	37.94	31.33	*40.62*	28.39	26.79	**45.83**	40.08
P05→P07	26.64	40.33	26.21	40.76	10.80	58.81	*77.53*	**77.84**	69.00
P05→P09	27.39	41.97	27.27	41.79	7.29	54.23	*66.42*	**70.96**	57.41
P05→P27	30.28	55.36	31.15	59.60	15.14	84.50	*90.78*	**91.86**	84.68
P05→P29	19.24	29.04	17.59	29.29	4.72	49.75	**52.70**	*52.08*	45.40
P07→P05	25.42	41.51	25.27	41.78	16.63	57.62	**74.70**	*74.55*	57.92
P07→P09	47.24	53.43	47.37	51.47	21.69	62.93	**79.66**	*77.08*	54.60
P07→P27	53.47	63.77	54.22	64.73	26.04	75.82	81.14	*83.84*	**86.09**
P07→P29	26.84	35.72	27.02	33.70	10.36	39.89	*63.73*	**69.24**	47.30
P09→P05	23.26	35.47	21.88	34.69	14.98	50.96	**77.16**	*73.98*	68.67
P09→P07	41.87	47.08	43.09	47.70	27.26	57.95	*78.39*	**79.01**	67.34
P09→P27	44.97	53.71	46.38	56.23	27.55	68.45	*84.92*	83.48	**87.47**
P09→P29	28.13	34.68	26.84	33.09	8.15	39.95	65.93	**70.04**	*67.65*
P27→P05	35.62	51.17	34.27	55.61	25.96	80.58	*92.83*	**93.07**	92.44
P27→P07	63.66	66.05	62.92	67.83	28.73	82.63	*90.18*	89.99	**93.68**
P27→P09	72.24	73.96	73.35	75.86	38.36	87.25	*90.14*	89.71	**90.20**
P27→P29	36.03	40.50	37.38	40.26	7.97	54.66	72.18	*76.84*	**79.53**
P29→P05	23.05	26.89	20.35	27.01	9.54	46.46	*60.20*	**67.32**	52.67
P29→P07	26.03	31.74	24.62	29.90	8.41	42.05	**71.39**	*70.23*	57.52
P29→P09	27.76	31.92	28.49	29.90	6.68	53.31	*74.02*	**74.63**	62.81
P29→P27	30.31	34.70	31.27	33.67	10.06	57.01	*76.66*	75.43	**80.98**
Average	41.14	47.55	40.65	47.59	26.79	60.63	*72.85*	**74.67**	68.73

Experimental Details

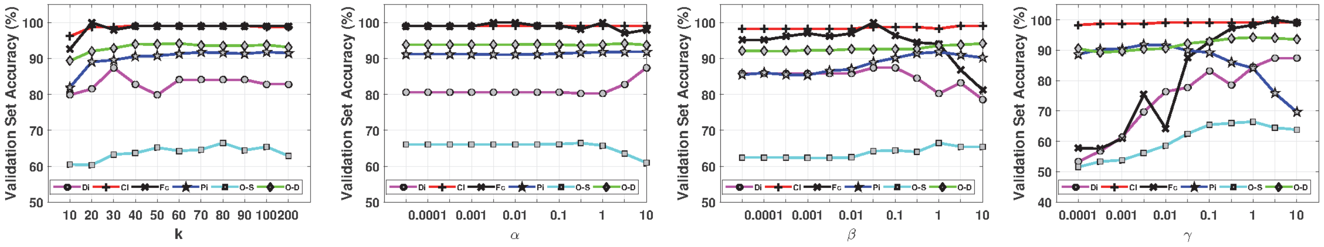
For fair comparison with existing methods, we follow the same experimental protocol as in (Gong et al. 2012; Long et al. 2014). We conduct 50 different domain adaptation experiments with the previously mentioned datasets. Each of these is an unsupervised domain adaptation experiment with one source domain (data points and labels) and one target domain (data points only). When estimating $\mathbf{M}_c$, we choose 10 iterations to converge to the predicted test/validation labels. Wherever necessary, we use a Gaussian kernel for $k(.,.)$, with a standard width equal to the median of the squared distances over the dataset. We train a 1-Nearest Neighbor (NN) classifier using the projected source data and test on the projected target data for all the experiments. We choose a NN classifier as in (Gong et al. 2012; Long et al. 2014), since it does not require tuning of cross-validation parameters. The accuracies reflect the percentage of correctly classified target data points.

Parameter Estimation Study

Here we evaluate our model selection procedure. The NET algorithm has 4 parameters $(\alpha, \beta, \gamma, k)$, and the JDA has 2 parameters (γ, k), that need to be pre-determined. To determine these parameters, we weight the source data points using Equation (11) and select 10% of the source data points with the largest weights. These source data points have a distribution similar to the target and they are used as a validation set to determine the optimal values for the model parameters $(\alpha, \beta, \gamma, k)$. The parameter space consists of $k \in \{10, 20, \ldots, 100, 200\}$ and α, β, γ from the set {0, 0.0001,

Table 4: Recognition accuracies (%) for domain adaptation experiments on the *Office-Caltech* dataset with SURF and Deep features. {`Amazon`(A), `Webcam`(W), `Dslr`(D), `Caltech`(C)}. A→W implies A is source and W is target. The best and second best results in every experiment (row) are in **bold** and *italic* respectively. The shaded columns indicate accuracies obtained with model selection.

Expt.	**SURF Features**									**Deep Features**								
	SA	**CA**	**GFK**	**TCA**	**TJM**	**JDA**	**JDA$_v$**	**NET**	**NET$_v$**	**SA**	**CA**	**GFK**	**TCA**	**TJM**	**JDA**	**JDA$_v$**	**NET**	**NET$_v$**
C→A	43.11	36.33	45.72	44.47	**46.76**	44.78	45.41	*46.45*	46.24	88.82	*91.12*	90.60	89.13	91.01	90.07	89.34	**92.48**	90.70
D→A	29.65	28.39	26.10	31.63	32.78	33.09	29.85	**39.67**	*35.60*	84.33	86.63	88.40	88.19	88.72	91.22	90.18	**91.54**	*91.43*
W→A	32.36	31.42	27.77	29.44	29.96	32.78	29.33	**41.65**	*39.46*	84.01	82.76	88.61	86.21	88.09	91.43	87.04	**92.58**	*91.95*
A→C	38.56	33.84	39.27	39.89	39.45	39.36	39.27	**43.54**	*43.10*	80.55	*82.47*	81.01	75.53	78.08	83.01	78.27	**83.01**	82.28
D→C	31.88	29.56	30.45	30.99	31.43	31.52	31.08	**35.71**	*34.11*	76.26	75.98	78.63	74.43	76.07	80.09	78.17	*82.10*	**83.38**
W→C	29.92	28.76	28.41	32.15	30.19	31.17	31.43	**35.89**	*32.77*	78.90	74.98	76.80	76.71	79.18	**82.74**	78.90	*82.56*	82.28
A→D	37.58	36.94	34.40	33.76	**45.22**	39.49	31.85	*40.76*	36.31	82.17	*87.90*	82.80	82.17	87.26	89.81	77.07	**91.08**	80.89
C→D	43.95	38.22	43.31	36.94	44.59	*45.22*	40.13	**45.86**	36.31	80.89	82.80	77.07	75.80	82.80	89.17	80.25	**92.36**	*90.45*
W→D	*90.45*	85.35	82.17	85.35	89.17	89.17	88.53	89.81	**91.72**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	*99.36*	**100.00**
A→W	37.29	31.19	41.70	33.90	*42.03*	37.97	38.98	**44.41**	35.25	82.37	80.34	84.41	76.61	87.12	87.12	79.32	**90.85**	*87.46*
C→W	36.27	29.49	35.59	32.88	38.98	*41.69*	37.97	**44.41**	33.56	77.29	79.32	78.64	78.31	*88.48*	85.76	77.97	**90.85**	84.07
D→W	87.80	83.39	79.66	85.42	85.42	*89.49*	86.78	87.80	**90.51**	98.98	*99.32*	98.31	97.97	98.31	98.98	98.98	**99.66**	**99.66**
Average	44.90	41.07	42.88	43.07	*46.33*	46.31	44.22	**49.66**	46.24	84.55	85.30	85.44	83.42	87.09	*89.12*	84.63	**90.70**	88.71



(a) # bases k

(b) MMD weight α

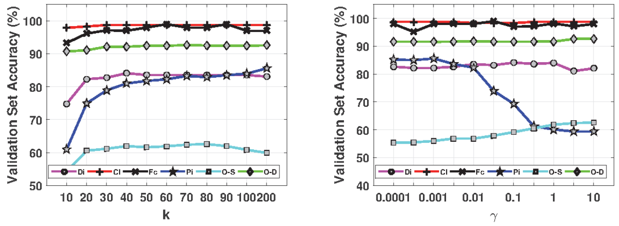
(c) Embed weight β

(d) Regularization γ

Figure 2: NET Validation Study. Each figure depicts the accuracies over the source-based validation set. When studying a parameter (say k), the remaining parameters (α, β, γ) are fixed at the optimum value. The legend is, Digit (Di), Coil (Cl), MMI&CK+ Face (Fc), PIE (Pi), Office-Caltech SURF (O-S) and Office-Caltech Deep (O-D).

0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5, 10}. For the sake of brevity, we present one set of parameters for every dataset, although in practice, a unique set of parameters can be evaluated for every domain adaptation experiment. Given a set of model parameters, we conduct the domain adaptation experiment using the entire source data (data and labels) and the target data (data only). The accuracies obtained are represented as shaded columns JDA$_v$ and NET$_v$ in Tables (3) and (4).

In order to evaluate the validity of our proposed model selection method, we also determine the parameters using the target data as a validation set. These results are represented by the NET column in Tables (3) and (4). Since the NET column values have been determined using the target data, they can be considered as the best accuracies for the NET model. The rest of the column values SA, CA, GFK, TCA, TJM and JDA, were estimated with model parameters suggested in their respective papers. The recognition accuracies for NET$_v$ is greater than that of the other domain adaptation methods and is nearly comparable to the NET. In Table (3), the JDA$_v$ has better performance than the JDA. This goes to show that a proper validation procedure does help select the best set of model parameters. It demonstrates that the proposed model selection procedure is a valid technique for evaluating an unsupervised domain adaptation algorithm in the absence of target data labels. Figures (2) and (3), depict the variation of average validation set accuracies over the model parameters. Based on these curves, the optimal parameters are chosen for each of the datasets.



(a) # bases k

(b) Regularization γ

Figure 3: JDA Validation Study. Each figure depicts the accuracies over the source-based validation set. When studying a parameter (say k), the remaining parameter γ is fixed at the optimum value. The legend is, Digit (Di), Coil (Cl), MMI&CK+ Face (Fc), PIE (Pi), Office-Caltech SURF (O-S) and Office-Caltech Deep (O-D).

NET Algorithm Evaluation

The NET algorithm has been compared to existing unsupervised domain adaptation procedures across multiple datasets. The results of the NET algorithm are depicted under the NET column in Tables (3) and (4). The parameters used to obtain these results are depicted in Table (5). The accuracies obtained with the NET algorithm are consistently better than any of the other spectral methods (TCA, TJM and JDA). NET also consistently performs better compared to non-spectral methods like SA, CA and GFK.

Table 5: Parameters used for the NET model.

Dataset	α	β	γ	k
MNIST & USPS	1.0	0.01	1.0	20
MMI & CK+	0.01	0.01	1.0	20
COIL	1.0	1.0	1.0	60
PIE	10.0	0.001	0.005	200
Ofc-SURF	1.0	1.0	1.0	20
Ofc-Deep	1.0	1.0	1.0	20

Discussion and Conclusions

The average accuracies obtained with JDA and NET using the validation set are comparable to the best accuracies with JDA and NET. This empirically validates the model selection proposition. However, there is no theoretical guarantee that the parameters selected are the best. In the absence of theoretical validation, further empirical analysis is advised when using the proposed technique for model selection.

In this paper, we have proposed the Nonlinear Embedding Transform algorithm and a model selection procedure for unsupervised domain adaptation. The NET performs favorably compared to competitive visual domain adaptation methods across multiple datasets.

This material is based upon work supported by the National Science Foundation (NSF) under Grant No:1116360. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

References

Aytar, Y., and Zisserman, A. 2011. Tabula rasa: Model transfer for object category detection. In *Intl. Conference on Computer Vision*.

Belkin, M., and Niyogi, P. 2003. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation* 15(6):1373–1396.

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

Chatfield, K.; Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*.

Chung, F. R. 1997. *Spectral graph theory*, volume 92. American Mathematical Soc.

Duan, L.; Tsang, I. W.; and Xu, D. 2012. Domain transfer multiple kernel learning. *Pattern Analysis and Machine Intelligence, IEEE Trans. on* 34(3):465–479.

Fernando, B.; Habrard, A.; Sebban, M.; and Tuytelaars, T. 2013. Unsupervised visual domain adaptation using subspace alignment. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2960–2967.

Gong, B.; Shi, Y.; Sha, F.; and Grauman, K. 2012. Geodesic flow kernel for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Gong, B.; Grauman, K.; and Sha, F. 2013. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *Intl. Conference on Machine learning*, 222–230.

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

Hoffman, J.; Rodner, E.; Donahue, J.; Saenko, K.; and Darrell, T. 2013. Efficient learning of domain-invariant image representations. In *Intl. Conference on Learning Representations*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Neural Information Processing Systems*, 1097–1105.

Long, M.; Wang, J.; Ding, G.; Sun, J.; and Yu, P. S. 2013. Transfer feature learning with joint distribution adaptation. In *Intl. Conference on Machine Learning*, 2200–2207.

Long, M.; Wang, J.; Ding, G.; Sun, J.; and Yu, P. 2014. Transfer joint matching for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1410–1417.

Lucey, P.; Cohn, J. F.; Kanade, T.; Saragih, J.; Ambadar, Z.; and Matthews, I. 2010. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *IEEE Conference on Computer Vision and Pattern Recognition*, 94–101.

Pan, S. J., and Yang, Q. 2010. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Trans. on* 22(10):1345–1359.

Pan, S. J.; Tsang, I. W.; Kwok, J. T.; and Yang, Q. 2011. Domain adaptation via transfer component analysis. *Neural Networks, IEEE Trans. on* 22(2):199–210.

Pantic, M.; Valstar, M.; Rademaker, R.; and Maat, L. 2005. Web-based database for facial expression analysis. In *IEEE Conference on Multimedia and Expo*. IEEE.

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

Sun, B.; Feng, J.; and Saenko, K. 2015. Return of frustratingly easy domain adaptation. In *Intl. Conference on Computer Vision, TASK-CV*.

Torralba, A., and Efros, A. A. 2011. Unbiased look at dataset bias. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1521–1528.