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...
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 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 non-linearly 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 $\mathcal{S}$ and target domain $\mathcal{T}$. Let $\mathcal{D}_s = \{ (x_i^s, y_i^s) \}_{i=1}^{n_s} \subset \mathcal{S}$ be a subset of the source domain and $\mathcal{D}_t = \{ (x_i^t, y_i^t) \}_{i=1}^{n_t} \subset \mathcal{T}$ be the subset of the target domain. Let $X_S = [x_1^s, ..., x_{n_s}^s] \in \mathbb{R}^{d \times n_s}$ and $X_T = [x_1^t, ..., x_{n_t}^t] \in \mathbb{R}^{d \times n_t}$ be the source and target data points respectively. Let $Y_S = [y_1^s, ..., y_{n_s}^s]$ and $Y_T = [y_1^t, ..., y_{n_t}^t]$ be the source and target labels respectively. Here, $x_i^s \in \mathbb{R}^d$ are data points and $y_i^s \in \{1, ..., C\}$ are the associated labels. We define $X= [x_1, ..., x_n] = [X_S, X_T]$, 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 $\hat{Y}_T = [\hat{y}_1^t, ..., \hat{y}_{n_t}^t]$ corresponding to $X_T$ using $\mathcal{D}_s$ and $X_T$.

**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 $\phi(X) = \{\phi(x_1), ..., \phi(x_n)\}$. $\phi : \mathbb{R}^d \rightarrow \mathbb{H}$ is the mapping function and $\mathbb{H}$ is a RKHS (Reproducing Kernel Hilbert Space). The dot product between the mapped vectors $\phi(x)$ and $\phi(y)$, is estimated by a positive semi-definite (psd) kernel, $k(x, y) = \phi(x)^\top \phi(y)$. The dot product captures the similarity between $x$ and $y$. The kernel similarity (gram) matrix consisting of similarities between all the data points in $X$, 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,

\[
\max_{\mathbf{A}} \quad \text{tr}(\mathbf{A}^\top \mathbf{K} \mathbf{H} \mathbf{K}^\top \mathbf{A}).
\]

(1)

Here, \( \mathbf{H} \) is the \( n \times n \) centering matrix given by \( \mathbf{H} = \mathbf{I} - \frac{1}{n} \mathbf{1} \mathbf{1}^\top \), where \( \mathbf{I} \) is an identity matrix and \( \mathbf{1} \) is a \( n \times 1 \) matrix of 1s. \( \mathbf{A} \in \mathbb{R}^{n \times k} \), is the matrix of coefficients and the nonlinear projected data is given by \( \mathbf{Z} = [z_1, \ldots, z_n]^\top = \mathbf{A}^\top \mathbf{K} \in \mathbb{R}^{k \times n} \). 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 respectively.

\[
\min_{\mathbf{A}} \quad \frac{1}{n_t} \sum_{i=m+1}^{n} \mathbf{A}^\top \mathbf{k}_i - \frac{1}{n_t} \sum_{i=m+1}^{n} \mathbf{A}^\top \mathbf{k}_i = \text{tr}(\mathbf{A}^\top \mathbf{K} \mathbf{M}_s \mathbf{K}^\top \mathbf{A}).
\]

(2)

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

\[
(\mathbf{M}_0)_{ij} = \begin{cases} 
\frac{1}{n_{s(ij)}}, & x_i, x_j \in \mathcal{D}_s \\
\frac{1}{n_{ij}}, & x_i, x_j \in \mathcal{D}_t \\
0, & \text{otherwise}.
\end{cases}
\]

(3)

Likewise, the conditional distribution difference can also be minimized by introducing matrices \( \mathbf{M}_c \), with \( c = 1, \ldots, C \), defined as,

\[
(\mathbf{M}_c)_{ij} = \begin{cases} 
\frac{1}{n_{c(ij)}}, & x_i, x_j \in \mathcal{D}_{s(c)} \\
\frac{1}{n_{ij}}, & x_i, x_j \in \mathcal{D}_{t(c)} \\
\frac{1}{n_{ij}}, & x_i \in \mathcal{D}_{s(c)}, x_j \in \mathcal{D}_{t(c)} \\
0, & \text{otherwise}.
\end{cases}
\]

(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 \( n_{c(ij)} = |\mathcal{D}_{s(c)}| \). Similarly, \( \mathcal{D}_{t(c)} \) is the subset of target data points with class label \( c \) and \( n_{c(ij)} = |\mathcal{D}_{t(c)}| \). Since the target labels being unknown, we use predicted target labels to determine \( \mathcal{D}_{s(c)} \). 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,

\[
\min_{\mathbf{A}} \quad \alpha \text{tr}(\mathbf{A}^\top \mathbf{K} \sum_{c=0}^{C} \mathbf{M}_c \mathbf{K}^\top \mathbf{A})
\]

\[
+ \beta \text{tr}(\mathbf{A}^\top \mathbf{K} \mathbf{L} \mathbf{K}^\top \mathbf{A}) + \gamma ||\mathbf{A}||_F^2.
\]

(5)

\[
\text{Similarity Based Embedding}
\]

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.

\[
L(\mathbf{A}, \mathbf{A}) = \alpha \text{tr}(\mathbf{A}^\top \mathbf{K} \sum_{c=0}^{C} \mathbf{M}_c \mathbf{K}^\top \mathbf{A}) + \beta \text{tr}(\mathbf{A}^\top \mathbf{K} \mathbf{L} \mathbf{K}^\top \mathbf{A})
\]

\[
+ \gamma ||\mathbf{A}||_F^2 + \text{tr}((\mathbf{I} - \mathbf{A} \mathbf{K} \mathbf{D} \mathbf{K}^\top \mathbf{A}) \mathbf{A}),
\]

(6)
The solution $\mathbf{A}$ in (8) are the $k$-smallest eigen-vectors of Equation (10). The projected data points are then given by $Z = \mathbf{X}^T \mathbf{A}$.

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

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

The solution $\mathbf{A}$ in (8) are the $k$-smallest eigen-vectors of Equation (10). The projected data points are then given by $Z = \mathbf{A}^T \mathbf{K}$. 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 $w_i$, $i = 1, \ldots, n_s$, by minimizing $\left\| \frac{1}{n_s} \sum_{i=1}^{n_s} w_i \phi(x_i) - \frac{1}{n_i} \sum_{i=1}^{n_i} \phi(x_i) \right\|_2^2$. In order to simplify, we define $\kappa_i := \frac{1}{n_i} \sum_{i=1}^{n_i} k(x_i, x_i)$, $i = 1, \ldots, n_s$ and $\mathbf{K}_{S_{iS}} = k(x_i, x_j)$. The minimization is then represented as a quadratic programming problem,

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

The first constraint limits the scope of discrepancy between source and target distributions, with $B \rightarrow 1$, 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.
Office-Caltech dataset; (ii) 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     |
| COIL2        | Object       | 1,440    | 1,024     | 20       | COIL1, COIL2 |
| PIE          | Face         | 11,554   | 1,024     | 68       | P01 ... P29 |
| OC-Cal SURF  | Object       | 2,533    | 800       | 10       | A. C. W. D |
| OC-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 a parameter 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 \((\gamma, k)\), 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 pre-determined.

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 \((\gamma, k)\), 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.

| Method          | Reference                        | NET Acc. |
|-----------------|----------------------------------|----------|
| Exp.            | MNIST(U)                         | 76.39    |
|                | SA                               | 75.39    |
|                | CA                               | 5.61     |
|                | GFK                              | 5.61     |
|                | TCA                              | 6.62     |
|                | TJM                              | 6.62     |
|                | JDA                              | 6.62     |

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 \( M_{\ast} \), we choose 10 iterations to converge to the predicted test/validation labels. Wherever necessary, we use a Gaussian kernel for \( k(\cdot, \cdot) \), 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 algorithm has 4 parameters \((\alpha, \beta, \gamma, k)\), and the JDA has 2 parameters \((\gamma, k)\), 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 \((\alpha, \beta, \gamma)\). The parameter space consists of \( k \in \{10, 20, \ldots, 100, 200\} \) and \( \alpha, \beta, \gamma \) from the set \( \{0, 0.0001, \ldots \} \).
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\rightarrow 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 | NET | NET | SA | CA | GFK | TCA | TJM | JDA | NET | NET |
| C\rightarrow A | 43.11 | 46.33 | 36.72 | 45.87 | 48.56 | 47.79 | 45.41 | 46.24 | 36.27 | 36.12 | 39.00 | 38.13 | 39.01 | 39.07 | 39.14 | 39.48 | 90.70 |
| D\rightarrow A | 29.65 | 28.39 | 26.10 | 31.53 | 32.78 | 31.09 | 29.85 | 39.67 | 35.60 | 44.22 | 86.63 | 88.40 | 88.19 | 87.72 | 91.02 | 90.18 | 91.54 | 91.43 |
| W\rightarrow A | 32.50 | 31.84 | 32.17 | 29.84 | 29.96 | 32.78 | 29.33 | 41.65 | 39.46 | 44.01 | 82.76 | 88.61 | 88.21 | 89.09 | 91.43 | 87.04 | 92.58 | 91.95 |
| A\rightarrow C | 38.56 | 31.84 | 39.27 | 39.89 | 39.45 | 39.36 | 39.27 | **43.54** | 43.10 | 40.55 | 82.47 | 81.93 | 75.53 | 78.08 | 83.01 | 78.27 | **83.31** | 82.28 |
| D\rightarrow C | 31.88 | 29.56 | 10.03 | 30.99 | 31.43 | 31.52 | 31.08 | 35.71 | 34.11 | 17.26 | 75.98 | 78.63 | 78.42 | 76.07 | 80.09 | 78.17 | 82.10 | **83.38** |
| W\rightarrow C | 29.92 | 28.76 | 28.41 | 32.15 | 30.19 | 31.17 | 31.43 | 35.89 | 32.22 | 18.90 | 74.98 | 76.80 | 76.71 | 79.18 | **82.74** | 78.90 | 82.56 | 82.82 |
| A\rightarrow D | 37.58 | 36.94 | 34.40 | 33.76 | 45.24 | 39.49 | 31.85 | 40.75 | 36.31 | 82.17 | 87.90 | 82.80 | 82.17 | 87.26 | 89.81 | 77.07 | **91.08** | 80.89 |
| C\rightarrow 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 | **92.26** | 90.45 |
| W\rightarrow D | 90.45 | 82.35 | 82.17 | 85.25 | 89.17 | 89.17 | 88.53 | 89.81 | 91.72 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 99.56 | 100.00 |
| A\rightarrow 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\rightarrow W | 36.21 | 29.97 | 35.39 | 32.88 | 38.98 | 41.69 | 37.97 | **44.41** | 33.56 | 71.29 | 79.32 | 78.94 | 78.31 | 85.45 | 85.76 | 77.97 | **90.85** | 84.07 |
| D\rightarrow W | 87.80 | 83.39 | 79.66 | 85.42 | 85.42 | 89.49 | 86.78 | 87.80 | **90.51** | 98.98 | 99.22 | 98.31 | 97.97 | 98.31 | 98.98 | 98.98 | **99.66** | 99.66 |
| **Average** | 44.90 | 41.07 | 42.88 | 43.07 | 46.55 | 46.31 | 46.22 | **49.86** | 46.24 | 48.55 | 85.30 | 85.44 | 83.42 | 87.09 | 89.72 | 84.63 | **90.70** | 88.11 |

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 ($\alpha, \beta, \gamma$) 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).

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 $\gamma$ 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 column 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 | 0.001 | 0.01  | 0.01  | 20    |
| MNIST & CK+  | 0.01  | 0.01  | 1.0   | 20    |
| COIL       | 0.01  | 0.01  | 0.005 | 200   |
| PIE        | 0.001 | 0.01  | 0.01  | 20    |
| Ofc-SURF   | 0.01  | 0.01  | 0.01  | 20    |
| Ofc-Deep   | 0.01  | 0.01  | 0.01  | 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.