Feature Selection for Transfer Learning

Selen Uguroglu and Jaime Carbonell

Language Technologies Institute, Carnegie Mellon University
{sugurogl,jgc}@cs.cmu.edu

Abstract. Common assumption in most machine learning algorithms is that, labeled (source) data and unlabeled (target) data are sampled from the same distribution. However, many real world tasks violate this assumption: in temporal domains, feature distributions may vary over time, clinical studies may have sampling bias, or sometimes sufficient labeled data for the domain of interest does not exist, and labeled data from a related domain must be utilized. In such settings, knowing in which dimensions source and target data vary is extremely important to reduce the distance between domains and accurately transfer knowledge. In this paper, we present a novel method to identify variant and invariant features between two datasets. Our contribution is two fold: First, we present a novel transfer learning approach for domain adaptation, and second, we formalize the problem of finding differently distributed features as a convex optimization problem. Experimental studies on synthetic and benchmark real world datasets show that our approach outperform other transfer learning approaches, and it aids the prediction accuracy significantly.

1 Introduction

In real life applications of supervised machine learning, the conditions in which the models are developed and used may differ. For instance, in clinical studies of drugs, the selection of patients may not be representative of the general population: the sample may have a gender bias, race bias, or patients in the study may have a lower health status. A network intrusion software or a spam detection software developed many years ago may not be predictive anymore, due to newer attack or spam patterns. A survey conducted in a region, may not be applicable to another region, due to differences in the populations. These examples conflict with the major assumption of machine learning, that training and testing data come from the same distribution. Moreover it is not always guaranteed that there is sufficient labeled data in the target (newer) domain to train a new model.

There have been many efforts to deal with such situations, including relatively new learning paradigms, such as transfer learning. Transfer learning tries to utilize the readily available labeled data from another domain for prediction in the target domain of interest. This approach is also known as domain adaptation. An example application area for domain adaptation is sentiment analysis, where one intends to use reviews in a particular domain, say stock reviews, to predict the sentiments in another domain, say computer reviews. How products

D. Gunopulos et al. (Eds.): ECML PKDD 2011, Part III, LNAI 6913, pp. 430–442, 2011.
© Springer-Verlag Berlin Heidelberg 2011

are described in the reviews differ across domains, and therefore two dataset distributions may be different [10]. In bioinformatics, one may want to utilize labeled clinical data from one institution, to predict high-risk patients in another institution. Although the two clinical datasets may share the same feature sets (age, blood pressure, BMI etc.) we have no apriori reason to believe that the sets of patients come from the same distribution.

An effective domain adaptation is only possible when common feature representations of source and target domains are found [2]. In the literature of statistics and machine learning, the question of whether two datasets come from the same distribution, also known as the two-sample or homogeneity problem, has been tackled for many years in the context of data integration [3]. However, identifying which features are variant between two datasets is a relatively unexplored problem. Solving this problem is crucial for domain adaptation, since once the invariant features are identified, source and target domains can be reduced to the same distribution, and supervised machine learning algorithms can be applied.

In this paper, we present a novel, reliable and efficient unsupervised method to distinguish variant and invariant features across source and target datasets. We formulate the problem as a convex optimization problem which has obvious advantages such as reliability and efficiency. Experiments on the synthetic and real-world datasets reveal that: 1. Our method can discriminate between variant and invariant features with perfect accuracy 2. Knowing which features are variant is extremely important for domain adaptation: there is 30% improvement in prediction accuracy when only invariant features are used for training as opposed to all features 3. Our method outperforms other state-of-the-art transfer learning approaches on benchmark real-world datasets. Finally, rather than projecting source and target datasets to the feature space, or re-weighting instances to reduce their distance, we introduce a novel transfer learning approach.

The rest of the paper is organized as follows: In the following section, we formalize the problem statement and our solution. In section 3, we describe the experiments we conducted on synthetic and real world datasets. In section 4, we review related work. In the last section, we conclude the paper and provide future prospects.

2 Feature Selection with MMD (f-MMD)

Current methods in information theory can provide a measure of distance between two domains. For example, Kullback-Leibler divergence is a widely used metric to measure distance between two distributions. However, it requires expensive distribution density calculation. As a non-parametric distance measure, Borgwardt et al. proposed Maximum Mean Discrepancy statistic [3]. The main idea is that, under a sufficiently rich reproducing kernel Hilbert space (RKHS), if feature means of the population are identical then it is guaranteed that the distributions are the same [6]. Based on this theorem, distance between samples of two distributions can be measured by the difference of the empirical means of the samples in a RKHS [10]. Formally:

Definition: Let X = {x_1,x_2,..,x_m} and Y= {y_1,y_2,..,y_n} be two sets of observations drawn from Borel probability distributions p and q. Let $\mathcal{F}$ be a class of functions f: $\mathcal{X} \rightarrow \mathbb{R}$ then the empirical estimate of MMD is defined as :

$$MMD[\mathcal{F}, X, Y] = sup_{f \in \mathcal{F}} \left(\frac{1}{m} \sum_{i=1}^{m} f(x_i) - \frac{1}{n} \sum_{i=1}^{n} f(y_i) \right) \tag{1}$$

As shown by Borgwardt et al. [3] when $\mathcal{F}$ is rich enough, MMD ($\mathcal{F}$, X, Y) will be zero if and only if p = q. However, if it is too rich, for most finite samples of X and Y, MMD will differ significantly from 0. It has been shown that the unit ball in a universal RKHS is a sufficiently large function class that can be chosen as $\mathcal{F}$. Let $\phi(x)$ be the feature map defined as $\phi(x)$: $\mathcal{X} \rightarrow \mathcal{H}$, where $\mathcal{H}$ is a universal RKHS. Function evaluation can then be written as f(x) = $\langle \phi(x), f \rangle$. Based on this argument, the distance between two domains, S and T, can be measured by the squared difference in the empirical means:

$$Dist(X_S, X_T) = \left|\left| \frac{1}{n_S} \sum_{i=1}^{n_S} \phi(X_{S_i}) - \frac{1}{n_T} \sum_{i=1}^{n_T} \phi(X_{T_i}) \right|\right|^2_{\mathcal{H}} \tag{2}$$

Let x and y denote instances from source and target domains respectively: $x \in X_S, y \in X_T$. Following Pan et al., equation (2) can be written in the following form using the kernel trick, i.e. k(z_i, z_j^T) = $\phi(z_i)\phi(z_j^T)$, where k is a positive definite kernel [9]:

such that:
$$Dist(X_S, X_T) = tr(KL) \tag{3}$$

$$K = \begin{bmatrix} K_{xx} & K_{xy} \\ K_{yx} & K_{yy} \end{bmatrix} \quad \text{and} \quad L = \begin{bmatrix} L_{xx} & L_{xy} \\ L_{yx} & L_{yy} \end{bmatrix} \tag{4}$$

where K $\in \mathbb{R}^{(n_S+n_T)\times(n_S+n_T)}$ is a composite kernel matrix and K_{xx}, K_{yy} and K_{xy} are kernel matrices defined by k on the source domain, target domain and cross domains respectively. n_S and n_T denotes the number of instances in the source and target domains. L $\succeq 0$ is a coefficient matrix with $L_{xx} = \frac{1}{n_S^2}$, $L_{yy} = \frac{1}{n_T^2}$ and $L_{xy} = \frac{-1}{n_S * n_T}$.

Our goal is to find the features whose distributions are variant between the two domains. In other words, we are trying to find features which contribute to the distance between two domains the most. Let's define a diagonal weight matrix W, whose diagonal entries correspond to feature weights. Let W $\in \mathbb{R}^{d\times d}$ be this diagonal weight matrix and $x \in \mathbb{R}^{d\times 1}$ be a d dimensional sample vector. Let σ be a feature map such that:

$$\sigma(x) \rightarrow x\, W^{1/2}.$$

Finally, we can define the new (positive definite) kernel, $k'(x_i, x_j^T)$ as:

$$k'(x_i, x_j^T) = \langle \phi \circ \sigma(x_i), \phi \circ \sigma(x_j^T) \rangle_{\mathcal{H}}$$

where $\mathcal{H}$ is a universal RKHS.

The new kernel matrices K'_{xx}, K'_{yy}, K'_{xy} can be defined by k' on the source domain, target domain and cross domain respectively. The new composite kernel matrix K' can be computed with equation (4) on K'_{xx}, K'_{yy}, K'_{xy} (note that $K'_{yx} = (K'_{xy})^T$).

To illustrate the argument with an example, let $\phi(x)$: X $\rightarrow \mathcal{H}$ be a polynomial kernel with degree, d. Then, K'_{xx} is:

$$K'_{xx} = ((xW^{1/2}) * (xW^{1/2})^T \quad +1)^d$$
$$K'_{xx} = (xWx + 1)^d$$

K'_{yy}, K'_{xy} can be computed in a similar fashion.

To solve for the matrix W, we present the following convex optimization problem:

$$\begin{aligned} W^* = \arg\min_{W} \quad & -trace(K'L) \\ \text{subject to} \quad & diag(W)^T * diag(W) \leq 1 \\ & W > 0 \end{aligned} \tag{5}$$

where diag(W) $\in \mathbb{R}^{d\times 1}$ is the diagonal of the weight matrix. Intuition behind this optimization problem is the following:

- By assigning higher weights to features which minimize the negative MMD score in the objective function, we create a gap between the weights of the variant and invariant features across domains. Our assumption here is that there is at least one feature that differs across domains, i.e. the domains are not identical.
- With the first constraint, we are constraining the size of weights by applying a ridge penalty.

Equation (5) is a quadratically constrained quadratic program (QCQP) which can be cast as a Semidefinite Program (SDP). When interior point methods are used, QCQP can have polynomial worst-case complexity [14]. After solving equation (5), using a QCQP solver (in our experiments we used CVX [5][4]), variant features can be found by applying a threshold function to the diagonal entries of W^*. Denoting the set of variant features as V, and the set of invariant features as N, Algorithm 1 describes how we populate V and N. For the rest of this paper we will refer our approach as f-MMD.

3 Experiments

We tested the performance of our algorithm first on the synthetic datasets we designed, and then on the real world datasets. With the synthetic datasets, our purpose is to see how well our algorithm can distinguish between variant and

Algorithm 1. Feature Separation with MMD (f-MMD)

Input: Samples from source domain, X_S, and target domain, X_T, weight threshold λ
Output: Variant feature set V, and invariant feature set N.
Solve the optimization problem (5) to obtain the weight matrix W^*.
w $\leftarrow$ diag(W^*)
for $i = 1 : d$ where d is the number of features **do**
  if $w_i \geq \lambda$ **then**
    V $\leftarrow$ V $\cup$ i
  else $\{w_i < \lambda\}$
    N $\leftarrow$ N $\cup$ i
  end if
end for

invariant features, and how this information can aid the classification performance. With the experiments on the real world datasets, since we don't know apriori which features are identically distributed (or differently distributed), we only measured the improvement in the prediction performance, after applying our algorithm to select invariant features.

3.1 Synthetic Datasets

Synthetic datasets are designed to address the following questions:

- Can f-MMD identify features whose distribution vary between domains?
- How does the removal of the variant features affect prediction performance?

Synthetic data is generated as follows: Given m, the number of samples from each domain, d, the number of dimensions, k, the number of variant dimensions and t, the number of dimensions related to the class label, we sample invariant (d-k) features from (d-k) randomly picked distributions with zero mean and unit variance. For the first domain, k variant dimensions are sampled from randomly picked k distributions with zero mean and unit variance. For the second domain, these dimensions are sampled from the same k distributions but with linear shift in sample mean. Similar to the random signal generation used in [1], there are 18 distributions from which a feature is sampled: exponential, student (degrees of freedom = 3 or 5), Laplace, mixture of 2 double exponentials, symmetric 2 gauss (multimodal, transmodal, unimodal), uniform, asymmetric 2 gauss (multimodal, transmodal, or unimodal), asymmetric 4 gauss (multimodal, transmodal, or unimodal) and symmetric 4 gauss (multimodal, transmodal, or unimodal). To create class labels, d dimensional weight vector, $v \in R^d$ is drawn from the standard uniform distribution. t features that are related to the class label are randomly selected from all d features. A d dimensional indicator vector is constructed, where $I_i = 1$ if i^{th} feature is related to the class label, 0 otherwise. Consequently, $\sum_{j=1}^{d} I_j = t$. For i = 1...d, we set $v_i = 0$ if $I_i = 0$, otherwise, we left it unchanged. Class labels are then found by applying sign function to the data sample x, i.e. y = sign(v*x).

The illustration of the synthetic data in 2 dimensions are shown in Figure 1. Source data is shown in red, target data is shown in blue. x_1 is the invariant dimension (identically distributed in both domains), x_2 is the variant dimension - there is a linear shift in the means across domains. x_2 is also the predictive dimension for class labels in this example.

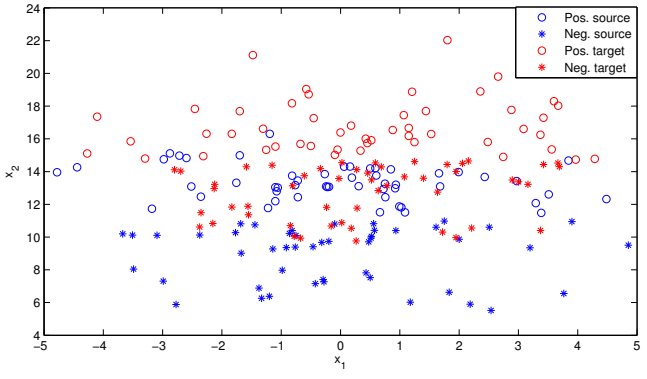


Fig. 1. 2D synthetic data, source data is shown in red, target data is shown in blue. Positive samples are shown with stars, and negative samples are shown with dots.

Next, we created d = 100 dimensional synthetic dataset with 200 samples (100 instances from each domain) by using the procedure described above. Each subfigure in Figure 2 shows the output weights for each dimension enumerated from 1 to 100. As evident from each subfigure, the features that are given significantly higher weights by our algorithm, are indeed variant features.

To illustrate the weight distribution between variant and invariant features, we generated another synthetic dataset with 10 dimensions. This time we varied k from 1 to 10, i.e. we generated 10 source and target datasets with 1 to 10 variant dimensions. The output weights sorted in descending order are given in Table 1. λ parameter for the threshold function can be found empirically by observing the output weight distribution, and picking a value from the range with largest difference in weights. In our experiments, we used $\lambda = 0.1$. Note that the number of features that have weights above 0.1 is exactly the same as k, our algorithm successfully identifies all of the variant features between datasets.

To address the second question, whether prediction performance improves after the removal of variant features, we trained linear SVM and logistic regression on source domain to predict class labels in the target domain. Linear classifiers are chosen to capture the contribution of each feature independently to the classification prediction. We used a source (training) dataset of size 300, with 20 dimensions (10 variant, 10 invariant). We randomly picked 10 features to be related to the class variable, y.

Prediction performances of linear SVM and logistic regression on the synthetic dataset are shown in Table 2 and in Table 3 respectively. First column indicates the prediction accuracy when the classifier is trained only with the invariant

Table 1. Feature weights for each synthetic dataset sorted in descending order. k is the number of variant dimensions in each dataset. The weights for the invariant features are significantly lower than the variant features as expected. Dimensions that have weights above 0.1 are indeed the correct variant dimensions.

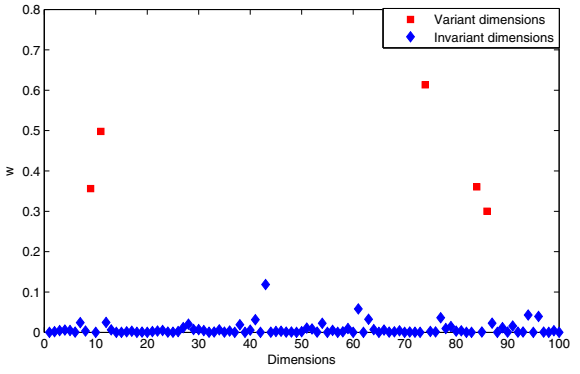
Dimension	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
w_1	0.999	0.827	0.793	0.620	0.531	0.503	0.466	0.621	0.412	0.484
w_2	0.028	0.563	0.470	0.597	0.498	0.464	0.435	0.339	0.406	0.335
w_3	0.027	0.008	0.386	0.420	0.462	0.420	0.414	0.325	0.402	0.311
w_4	0.024	0.007	0.006	0.287	0.440	0.390	0.399	0.292	0.341	0.306
w_5	0.004	0.005	0.003	0.034	0.251	0.328	0.372	0.291	0.315	0.305
w_6	0.003	0.004	0.001	0.000	0.002	0.310	0.285	0.287	0.309	0.295
w_7	0.003	0.002	0.001	0.000	0.001	0.002	0.206	0.267	0.289	0.293
w_8	0.001	0.002	0.000	0.000	0.001	0.001	0.002	0.262	0.289	0.283
w_9	0.000	0.001	0.000	0.000	0.000	0.001	0.000	0.046	0.163	0.266
w_{10}	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.016	0.001	0.213

features whose f-MMD weights smaller than $\lambda = 0.1$. Second column indicates the accuracy when its trained with all the features. Training sample size is fixed at 300 instances, while the target dataset size is varied from 100 to 300 samples. As can be seen from Table 2 and Table 3, there is a drastic improvement in prediction accuracy when only invariant features are used in training and testing, as opposed to using all the features. This further supports our intuition that using only the invariant features can have significant benefits in domain adaptation.

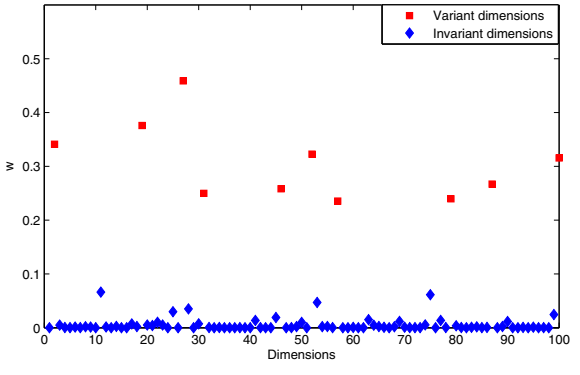
We also compared dimensionality reduction with f-MMD to other benchmark methods: Transfer Component Analysis (TCA) [10] and kernel PCA (KPCA). TCA is dimensionality reduction method that uses MMD as a distance measure across domains. It learns transfer components that reduce the distance across domains in a RKHS and performs mapping onto the learned transfer components. The prediction performance of the 3 algorithms on the synthetic dataset is shown in Table 4. Total number of features is 40. Source domain size is fixed at 300, and target domain size is increased from 100 to 300. The reduced number of dimensions is the same for all three methods, (d = 20). As can be seen in Table 4, our method outperforms both TCA and kernel PCA, especially as the target domain size increases.

3.2 Real World Datasets

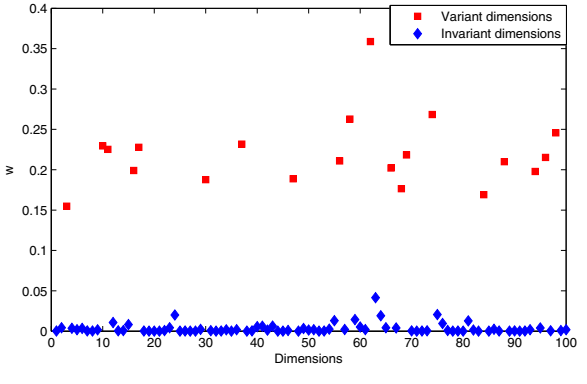
We tested our approach on two real world datasets: USPS handwritten digit images dataset and WIFI localization dataset [16]. Both datasets are commonly used in transfer learning tasks. USPS dataset contains images of size 16 x 16, totaling up 256 features, with pixel values ranging from 0 to 2. Many previous work states that, for the binary classification tasks on the USPS dataset, discriminating 4 from 7 [15] and 4 from 9 is particularly challenging [12][17].



(a) k = 5



(b) k = 10



(c) k = 20

Fig. 2. Weights, with respect to dimensions. Red dots illustrate variant features, blue dots illustrate invariant features. As can be seen in each subfigure, f-MMD weights for variant features are significantly higher than the invariant features.

Hence to make our task harder, for source domain we used digits 4 and 7, and for target domain we used digits 4 and 9. Labeling digit 4 as positive class and 7 and 9 as negative classes, the goal is to transfer the discriminative knowledge between 4 and 7 in the source domain to 4 and 9 in the target domain. Dimension size is determined by the output of f-MMD: Using a threshold, $\lambda = 0.1$, we removed all features with weights larger than λ. Denoting the number of remaining features with p, for f-MMD, linear SVM is trained with the reduced dataset of p-dimensions. For TCA, raw data is first mapped to the feature space, and dimensions with the highest p eigenvalues are used for classification.

WIFI localization dataset consists of wifi data collected in an indoor building at time points A and B. The goal is to predict the location where the wifi data is received at time point B, using the labeled data collected at time point A. This dataset has been used as a benchmark dataset for transfer learning applications, since source domain (data from time point A) and target domain (data from time point B) do not come from the same distribution [16]. For this dataset, we trained a ridge regression with ridge penalty 0.05 to predict the locations. Following [10] we used a training dataset of 621 instances, and varied the testing dataset size from 100 to 500. We compared our approach to KPCA and TCA, keeping the dimension size equal for all 3 methods.

Figure 3 (a) shows average absolute regression error results after applying TCA, KPCA and f-MMD respectively, along with the ridge regression results with no dimension reduction on the WIFI dataset. Figure 3 (b) shows the accuracy of linear SVM after applying TCA, KPCA and f-MMD on the USPS dataset. The exact performance results on WIFI localization dataset and USPS dataset are shown in Table 5 and Table 6 respectively. On both datasets, experiments show that our method significantly outperforms other feature representation methods: On the USPS handwritten digits dataset, for 550 samples, we obtain a 84% classification accuracy, while TCA and Kernel PCA achieves a mere 44% and 78.8% respectively. On the WIFI localization dataset, with the same number of dimensions and with 921 samples, average ridge regression error after dimensionality reduction with our algorithm is 88.6, while after TCA and Kernel PCA it is 103.37 and 92.4 respectively. This shows that our algorithm is a very promising method for domain adaptation.

Table 2. Prediction accuracy of SVM on synthetic dataset

#Samples	Invariant Features	All Features
400	86%	61%
450	86.7%	55.3%
500	86%	55.5%
550	87.2%	57.2%
600	87%	56.7%

Table 3. Prediction accuracy of logistic regression on synthetic dataset

#Samples	Invariant Features	All Features
400	82%	62%
450	84.7%	58%
500	83.5%	59.5%
550	85.2%	60.9%
600	84%	61.3%

Table 4. Linear SVM classification performance after f-MMD, TCA and KPCA on synthetic dataset

#Samples	f-MMD	TCA	KPCA
400	86%	81%	55%
500	86%	75%	48.5%
550	87.2%	67.2%	50.4%
600	87%	58%	51%

Table 5. Average absolute ridge regression error after f-MMD, TCA and KPCA on WIFI localization dataset

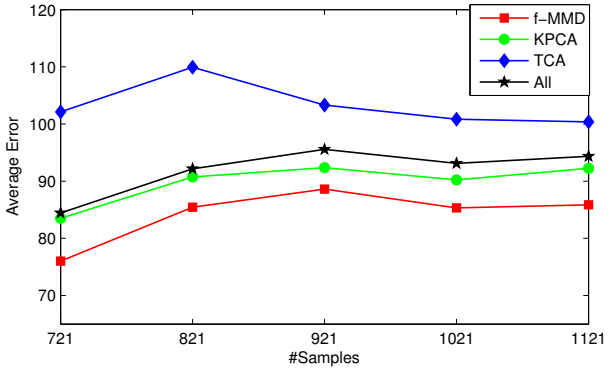
#Samples	f-MMD	TCA	KPCA
721	76.02	102.13	83.51
821	85.44	109.93	90.75
921	88.62	103.32	92.38
1021	85.32	100.84	90.23
1121	85.87	100.37	92.24

Table 6. Classification accuracy of linear SVM after f-MMD, TCA and KPCA on the USPS dataset

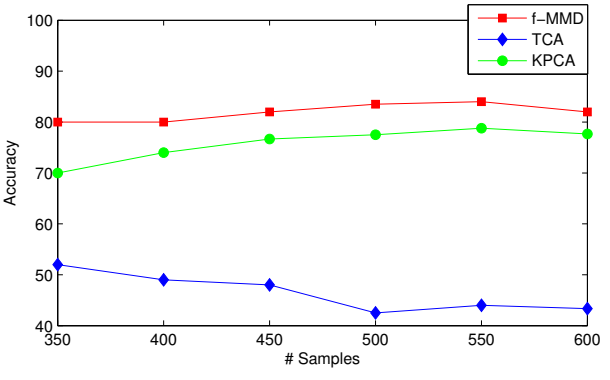
#Samples	f-MMD	TCA	KPCA
350	80	52	70
400	80	49	74
450	82	48	76.5
500	83.5	42	77.5
550	84	44	78.8
600	82	43	77.6

4 Related Work

Prior work in domain adaptation focuses on reducing the distance between source and target domains, either through re-weighting the instances (instance based approaches) or finding a common feature representation between two domains (feature based approaches). Instance based approaches, first estimate the weights



(a) WIFI



(b) USPS

Fig. 3. Comparison of f-MMD to benchmark methods. Figure 3 (a) shows average absolute ridge regression error of f-MMD, TCA and KPCA with respect to the sample size, as testing sample size is increased from 100 to 500. Results when no dimensionality reduction is applied is shown with black. Figure 3 (b) shows linear SVM prediction accuracy after f-MMD and TCA with respect to sample size as testing sample size is increased from 50 to 300.

corresponding to each instance in the source domain, prior model training. Instance weights are typically proportional to the distance between source and target density distributions. As a distance measure, Kullback-Leibler divergence [13], or Maximum Mean Discrepancy can be used [7]. Instance based approaches assume that there is a subset of instances with similar distributions in source and target domains, however such a subset may not exist when there are features that are variantly distributed across the two domains.

This work is mostly related to feature based approaches, which assume that the domains share a subset of features that come from similar distributions, and there are features that are variantly distributed across domains. The goal is to find a common feature representation of source and target domains. Among prior work in feature based approaches, Pan et al. first proposed Maximum Mean Discrepancy

Embedding (MMDE) [9] where the distance between distributions are measured with Maximum Mean Discrepancy. In MMDE, first, the kernel that minimizes distance between two distributions is found, and then kernel PCA is applied to the learned kernel. However this method requires an expensive SDP computation, and it is not feasible to be used on large datasets. Hence subsequently Pan et al. proposed Transfer Component Analysis (TCA), where the goal is to find a projection that minimizes distance between distributions [10]. TCA doesn't require an SDP computation, it is shown to be more efficient than MMDE in domain adaptation problems such as WIFI localization prediction [10].

5 Conclusion

In this paper, we proposed a novel and an extremely efficient method for domain adaptation. Unlike previous feature based approaches, rather than finding a projection of the feature space to maximize the similarity between source and target domains, we identify the features whose distribution vary between the two domains. In our experiments, we showed that knowing which features are variant and incorporating this knowledge to the prediction task significantly improves prediction performance, a novel finding in domain adaptation. We showed that our method significantly outperforms other comparable feature based methods on benchmark datasets. In the future, our goal is to extend this work to select the kernels that are differently distributed across domains. We are also intrigued to see how we can incorporate variant features to further increase prediction accuracy in domain adaptation.

References

1. Bach, F.R., Jordan, M.: Kernel Independent Component Analysis. Journal of Machine Learning Research 3, 1–48 (2002)
2. Blitzer J., McDonald R., Pereira F.: Domain Adaptation with Structural Correspondence Learning. In: EMNLP 2006: 2006 Conference on Empirical Methods in Natural Language Processing, pp. 120–128 (2006)
3. Borgwardt, K., Gretton, A., Rasch, M., Kriegel, H.P., Schölkopf, B., Smola, A.J.: Integrating structured biological data by kernel maximum mean discrepancy. Bioinformatics 22(14), 49–57 (2006)
4. Grant, M., Boyd, S.: Graph implementations for nonsmooth convex programs. In: Blondel, V., Boyd, S., Kimura, H. (eds.) Recent Advances in Learning and Control (a tribute to M. Vidyasagar). LNCIS, pp. 95–110. Springer, Heidelberg (2008)
5. Grant, M., Boyd, S.: CVX: Matlab software for disciplined convex programming, version 1.21 (April 2011), `http://cvxr.com/cvx`
6. Gretton, A., Borgwardt, K.M., Rasch, M., Schölkopf, B., Smola, A.: A Kernel Method for the Two-Sample-Problem. In: Advances in Neural Information Processing Systems. Proceedings of the 2006 Conference, vol. 19, pp. 513–520. MIT Press, Cambridge (2007)
7. Huang, J., Smola, A.J., Gretton, A., Borgwardt, K.M., Schölkopf, B.: Correcting sample selection bias by unlabeled data. Advances in Neural Information Processing Systems 19, 601–608 (2007)

8. Margolis, A.: Literature Review of Domain Adaptation with Unlabeled Data
9. Pan, S.J., Kwok, J.T., Yang, Q.: Transfer Learning via Dimensionality Reduction. In: AAAI, pp. 677–682 (2008)
10. Pan, S.J., Tsang, I.W., Kwok J.T., Yang, Q.: Domain Adaptation via Transfer Component Analysis. In: Proceedings of IJCAI 2009, pp. 1187–1192 (2009)
11. Pan, S.J., Yang, Q.: IEEE Transactions on Knowledge and Data Engineering 22(10), 1345–1359 (2010)
12. Remus, S., Tomasi, C.: Semi-Supervised Fisher Linear Discriminant (SFLD). In: ICASSP, pp. 1862–1865 (2010)
13. Sugiyama, M., Nakajima, S., Kashima, H., Bnau, P.V., Kawanabe, M.: Direct Importance Estimation with Model Selection and Its Application to Covariate Shift Adaptation. In: NIPS (2007)
14. Vandenberghe, L., Boyd, S.: Semidefinite Programming. SIAM Review 38(1), 49–95 (1996)
15. Xu, Z., Jin, R., Lyu, M.R., King, I.: Discriminative Semi-Supervised Feature Selection via Manifold Regularization. In: IJCAI, pp. 1303–1308 (2009)
16. Yang, Q., Pan, S.J., Zheng, V.W.: Estimating Location Using Wi-Fi. IEEE Intelligent Systems 23(1), 8–13 (2008)
17. Zeng, H., Cheung, Y.: Feature Selection for Local Learning Based Clustering. In: Theeramunkong, T., Kijsirikul, B., Cercone, N., Ho, T.-B. (eds.) PAKDD 2009. LNCS, vol. 5476, pp. 414–425. Springer, Heidelberg (2009)