Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence

Locality Preserving Projection for Domain Adaptation with Multi-Objective Learning

Le Shu, Tianyang Ma, Longin Jan Latecki
Computer and Information Sciences, Temple University
Philadephia, PA, 19122, USA
{slevenshu,ma.tianyang}@gmail.com, latecki@temple.edu

Abstract

In many practical cases, we need to generalize a model trained in a source domain to a new target domain. However, the distribution of these two domains may differ very significantly, especially sometimes some crucial target features may not have support in the source domain. This paper proposes a novel locality preserving projection method for domain adaptation task, which can find a linear mapping preserving the 'intrinsic structure' for both source and target domains. We first construct two graphs encoding the neighborhood information for source and target domains separately. We then find linear projection coefficients which have the property of locality preserving for each graph. Instead of combing the two objective terms under compatibility assumption and requiring the user to decide the importance of each objective function, we propose a multi-objective formulation for this problem and solve it simultaneously using Pareto optimization. The Pareto frontier captures all possible good linear projection coefficients that are preferred by one or more objectives. The effectiveness of our approach is justified by both theoretical analysis and empirical results on real world data sets. The new feature representation shows better prediction accuracy as our experiments demonstrate.

Introduction

In recent years, domain adaptation has gained significant attention in many areas of applied machine learning, including bio-informatics, speech and language processing, computer vision and etc. In many supervised machine learning and data mining tasks, it is usually assumed that both the labeled and unlabeled data are sampled from the same distribution. However, in many real-world tasks, this assumption does not hold. For example, in temporal domains, the feature distribution may be different from that of the former features over time. In clinical studies of disease, the selected samples may not be representative enough and have selection bias. Given a new domain of interest, there may not be sufficient labeled data, and labeled data from a related domain need to be utilized. In these practical problems, given that the instances in the training and test domains may be drawn from different distributions, traditional supervised learning can not achieve good performance on the new domain. Domain adaptation algorithms are therefore designed to bridge the distribution gap between training (source) data and test (target) data.

Most domain adaptation algorithms seeks to eliminate the difference between source and target distributions. They can be mainly categorized into two classes. The first class of methods seeks to make source distribution close to target distribution by re-weighting (importance sampling) source domain data. Such methods include (Huang et al. 2006),(Jiang and Zhai 2007a),(Mansour, Mohri, and Rostamizadeh 2008). The second class of methods are based on feature mapping or feature representation, such as (Blitzer, McDonald, and Pereira 2006),(Fox and Gomes 2008),(Pan and Yang 2010). The assumption is that although source and target data have different distributions, either there exists some general features which have similar conditional distributions in both domains, or it is possible to transform the original feature space into a new feature space which is predictive for the target domain.

In this paper, we propose a novel feature representation transfer method. Given labeled data from source domain and unlabeled data from target domain, locality preserving projections are learned simultaneously on both domains through a multi-objective optimization framework.

There are two key innovations in our method. First, we adopt locality preserving projections, a linear feature transformation method, to solve domain adaptation problem. Locality preserving projections (LPP) are first proposed in (He and Niyogi 2003) as a dimension reduction method. Its key advantage compared to PCA and LDA is that it can discover the "intrinsic dimensionality" of the data, which could be much lower than the original feature space. It builds a graph incorporating neighborhood information of the data set and then computes a transformation which maps the data points to a subspace. The linear transformation optimally preserves local neighborhood information. Compared to other dimension reduction methods, higher classification accuracy can be achieved in the low dimensional space learned by LPP. Because of its good performance and simple implementation, there have been many works using LPP to solve different tasks where promising results are achieved. However, to the best of our knowledge, those methods do not attempt to solve the domain adaptation problem. In our work, in order to solve the domain adaptation problem, a discriminative low dimensional *common* space is discovered using LPP. LPP is learnt simultaneously on source and target domain. This promises that the source label can be transferred to target data in the learnt low dimensional common space.

To simultaneously learn LPP on both domains, we use a multi-objective learning framework, which is our second contribution. We first construct two graphs encoding the neighborhood information of source and target data. Intuitively, LPP needs to preserve local neighborhood information on both source and target data. Therefore, there are two objective functions to be optimized. A standard way to solve the above problem is to combine the two objective terms into a single objective with a trade-off parameter. The trade-off parameter is crucial, and can be obtained using cross-validation. However, in this work, we argue that such paradigm may not be suitable for domain adaptation task, which is simply because the labels of the target data are missing, so it is impossible to perform cross-validation. Therefore, we adopt the multi-objective learning framework. We use the classic Pareto optimization, which allows multiple objectives to compete with each other in deciding the optimal trade-off. More details are introduced in the methodology section.

The rest of paper is organized as follows: We first review the related work. And then, we describe how to formulate LPP for domain adaptation via multi-objective framework. We further show how to solve the multi-objective optimization by finding the Pareto Frontier via generalized eigendecomposition. After that, experimental results on real world data sets are described in detail. Finally, we draw some conclusions.

Related Work and Discussion

Domain adaptation have been extensively studied in many research areas (Pan and Yang 2010), (Kulis, Saenko, and Darrell 2011),(III 2007),(Chen, Weinberger, and Blitzer 2011),(Chen et al. 2012). In this paper, we mainly consider the methods which assume that there are no labeled data in target domain (unsupervised domain adaptation). In particular, we review feature representation domain adaptation methods.

(Blitzer, McDonald, and Pereira 2006) proposed a heuristic method for domain adaptation which is called structural correspondence learning (SCL). SCL uses labeled data from both domains to induce the correspondence among features. SCL identify some domain invariant "pivot" features first, the other features are represented using their relative co-occurrence count with all pivot features. After that, SCL computes a projection matrix through the low rank approximation of the matrix. In (Jiang and Zhai 2007b), the main idea is to select features that are generalizable across domains. The method uses a regularized logistic regression classifier. During training, it allows the generalizable features to be less regularized, compared with the domain-specific features. However, their method for finding the generalizable features assumes that there are multiple source domains. Pan et al. (Pan, Kwok, and Yang 2008) attempt to discover a latent feature representation across domains by minimizing the feature distribution difference, which is measured by the Maximum Mean Discrepancy statistic. The method solves a semi-definite programming (SDP) and directly gives the kernel matrix. In (Pan et al. 2011), an improved version is proposed, which is called "transfer component analysis". The method reduces the distance between domain distributions dramatically by projecting the data onto the learned transfer components. The algorithm learns a kernel function that can be applied on new data sets. Gong et al. proposes geodesic flow kernel (GFK) to solve domain adaptation problems. (Gong et al. 2012). The method embeds the source and target data into Grassmann manifolds and constructs geodesic flow between them to model domain shifts. GFK integrates an infinite number of subspaces that lie on the geodesic flow from the source subspace to the target one. and find new feature representations which is robust to changes of domains. In our work, we aim to learn a linear mapping matrix, which can preserve the local neighborhood structure of both source and target data. By learning the locality preserving projections on both source and target data simultaneously, we are able to discover a lower dimensional space which is domain independent.

Locality preserving projections (He and Niyogi 2003) has been applied to solve many machine learning tasks. For example, LPP is adopted in (Cai et al. 2007) to perform document indexing. In (He et al. 2005), LPP is used to tackle face recognition problem in computer vision. Most recently, (Gu et al. 2012) proposed a feature selection method which incorporates LPP.

In this paper, we use Pareto optimization to learn LPP simultaneously on source and target domain. In Pareto optimization theory, the Pareto frontier captures all possible good solutions without requiring the users to set the correct parameter. Pareto optimization has not been widely used for the reason that it is NP-hard problem to compute the Pareto frontier in most cases. Recently, (Davidson et al. 2013) show that by imposing orthogonal constraints and with some relaxation, the Pareto frontier of graph cut type objectives can be computed efficiently by solving a generalized eigendecomposition problem. In this paper, we follow the solution proposed in (Davidson et al. 2013). However, we aim to solve the domain adaptation problem, while (Davidson et al. 2013) aim to tackle the multi-view clustering problem.

Problem Formulation

We assume that our data originate from two domains, Source (S) and Target (T). Source data is fully labeled, which is $(X_S, \mathbf{y_S}) = \{(x_S^1, y_S^1), (x_S^2, y_S^2), \cdots, (x_S^{n_s}, y_S^{n_s})\}$. Each pair of (x_S^i, y_S^i) lies in $R^d \times y$ space and samples from some distributions $P_S(X, Y)$. The target data has equal dimensionality d as source data, and is sampled from $P_T(X, Y)$. However we do not have any labels for the target domain data, i.e., $(X_T, ?) = \{(x_T^1, ?), (x_T^2, ?), \cdots, (x_T^{n_t}, ?)\}$. Given $(X_S, \mathbf{y_S})$ and $(X_T, ?)$, our goal is to learn linear projection coefficient $\mathbf{w} \in R^d$ such that the learned coefficients are discriminative for both domains.

If we consider $\mathbf{w}$ as coefficients in a linear projection function $y = \mathbf{w}^T x$, which maps data $x \in R^d$ to a continuous value y, then we think discriminative feature weights

$\mathbf{w}$ should has the property of locality preserving, i.e., if two data x^i and x^j are "close" then $\mathbf{w}^T x^i$ and $\mathbf{w}^T x^j$ should be as close as well. The same insight has been used in many existing approaches, where locality preserving property shows merits in solving other tasks such as dimension reduction, document indexing and feature selection (He and Niyogi 2003),(Cai et al. 2007),(He, Cai, and Niyogi 2005) .

In the rest of this section, we first describe how to construct the adjacency graphs for source and target data respectively. Given the two graphs, we show how to learn coefficients $\mathbf{w}$ simultaneously on both source and target data using a multi-objective optimization framework.

Graph Construction

Let A denote an adjacency graph, where each node represents a data point. We use A_S and A_T to denote the graph of source and target data respectively. When constructing A, an edge between nodes i and j exists if x_i and x_j are "close". The criteria for defining "close" can vary in different scenarios.

To construct A_T, since the labels of target data are not available, we define "close" in an unsupervised manner, i.e., nodes i and j are connected by an edge if i is among p nearest neighbors of j or j is among p nearest neighbors of i. Formally, we have:

$$A_T(i,j) = \begin{cases} \frac{x_T^j \cdot x_T^i}{\|x_T^j\| \cdot \|x_T^i\|} & \text{if } x_T^i \in N_p(x_T^j) \text{ or } x_T^j \in N_p(x_T^i) \\ 0 & \text{otherwise.} \end{cases} , \tag{1}$$

where $N_p(x_T^i)$ is the set of p nearest neighbors of x_T^i. Note that we compute the similarity matrix A_T with the cosine similarity measure. However, other similarity measures may be used.

For A_S, we take advantage of the available labels of source data, and define "close" in a supervised manner, i.e., nodes i and j are connected if x_i and x_j share the same label:

$$A_S(i,j) = \begin{cases} 1 & \text{if } x_S^i \text{ and } x_S^j \text{ share the same label} \\ 0 & \text{otherwise.} \end{cases} , \tag{2}$$

Note that unlike the weight computation for target data, the same weight 1 is used for all edges instead of computing cosine similarity (Cai et al. 2007).

Multi-Objective Optimization

Given data X and its adjacency graph A, we are trying to find a discriminative feature weight $\mathbf{w}$ which can preserve the local structure of data X. Here we assume that $\mathbf{w}$ projects data points in X to vector $\hat{\mathbf{y}}$, that is $\hat{\mathbf{y}} = X\mathbf{w}$, where X can either be X_S or X_T. We optimize $\mathbf{w}$ from a locality preserving view.

$$\begin{aligned} \mathbf{w} &= \arg\min_{\mathbf{w}} \frac{1}{2} \sum_{i,j=1}^{n} (\frac{\hat{y^i}}{\sqrt{d^i}} - \frac{\hat{y^j}}{\sqrt{d^j}})^2 A(i,j) \\ &= \arg\min_{\mathbf{w}} \frac{1}{2} \sum_{i,j=1}^{n} (\frac{\mathbf{w}^T x^i}{\sqrt{d^i}} - \frac{\mathbf{w}^T x^j}{\sqrt{d^j}})^2 A(i,j) \\ &= \arg\min_{\mathbf{w}} \mathbf{w}^T X D^{-1/2} L D^{-1/2} X^T \mathbf{w} \\ &= \arg\min_{\mathbf{w}} \mathbf{w}^T X \bar{L} X^T \mathbf{w} \end{aligned} \tag{3}$$

where $L = D - A$ is the graph Laplacian,and $d^i = \sum_j A(i,j)$ measures the local density around x^i. D is a diagonal matrix with $[d^1, d^2, \cdots, d^n]$ as its entries. The normalized graph Laplacian is denoted as $\bar{L} = D^{-1/2} L D^{-1/2}$.

The objective function in (3) incurs a heavy penalty if neighboring points x^i and x^j are mapped far away. Intuitively, to minimize (3) is to find $\mathbf{w}$ which can ensure that if x^i and x^j are "close" then $\mathbf{w}^T x^i$ and $\mathbf{w}^T x^j$ are close as well.

In order to optimize $\mathbf{w}$ on source and target graph simultaneously, it is clear that the optimization must involve two objective terms: $\mathbf{w}^T X_S \bar{L_S} X_S^T \mathbf{w}$ and $\mathbf{w}^T X_T \bar{L_T} X_T^T \mathbf{w}$. When combining two objective terms, a common practice is to convert two objective terms into a single objective term, by adding up two terms and using a parameter to control the trade-off, i.e.,

$$\mathbf{w} = \arg\min_{\mathbf{w}} \{\mathbf{w}^T X_S \bar{L_S} X_S^T \mathbf{w} + \alpha \mathbf{w}^T X_T \bar{L_T} X_T^T \mathbf{w}\} \tag{4}$$

The parameter α controls the trade-off between source and target graph. Therefore, it is critical to find a "good" parameter to guarantee that a feature coefficient is obtained by solving (4). A standard way to find the "good" parameter is through cross-validation. However, we argue that such paradigm may not be suitable for the unsupervised domain adaptation task, because the target data labels are unavailable, which makes it impossible to perform the cross-validation.

In our approach, instead of converging two separate objective terms into a single objective by introducing a trade-off parameter, we aim to directly solve the following multi-objective optimization, which is one of our main contributions.

$$\mathbf{w} = \arg\min_{\mathbf{w}} \{\mathbf{w}^T X_S \bar{L_S} X_S^T \mathbf{w}, \mathbf{w}^T X_T \bar{L_T} X_T^T \mathbf{w}\} \tag{5}$$

We add the following constraints where the last two constraints exclude the solution with eigenvalue 0.

$$\Omega \doteq \{\mathbf{w} \in R \mid \mathbf{w}^T \mathbf{w} = 1, X_S \mathbf{w} \perp D_S^{1/2} \mathbf{1}, X_T \mathbf{w} \perp D_T^{1/2} \mathbf{1}\} \tag{6}$$

To solve the above multi-objective optimization problem, we aim to find the Pareto frontier (Davidson et al. 2013). Before we introduce the concept of Pareto frontier, we first define Pareto improvement.

Pareto Improvement: We set $f_S(\mathbf{w}) = \mathbf{w}^T X_S \bar{L_S} X_S^T \mathbf{w}$ and $f_T(\mathbf{w}) = \mathbf{w}^T X_T \bar{L_T} X_T^T \mathbf{w}$. Given two coefficients $\mathbf{w}$ and $\mathbf{w}'$, we say $\mathbf{w}$ is a Pareto improvements over $\mathbf{w}'$ if and only if one of the following two conditions holds:

$$f_S(\mathbf{w}) < f_S(\mathbf{w}') \wedge f_T(\mathbf{w}) \leq f_T(\mathbf{w}')$$

or

$$f_S(\mathbf{w}) \leq f_S(\mathbf{w}') \wedge f_T(\mathbf{w}) < f_T(\mathbf{w}')$$

When $\mathbf{w}$ is a Pareto improvement over $\mathbf{w}'$, we say $\mathbf{w}$ is better than $\mathbf{w}'$.

Pareto frontier $\hat{P}$ refers to the optimal set of solutions, which satisfy the following three properties:

1. any $\mathbf{w}$ in $\hat{P}$ is better than that not in $\hat{P}$;
2. any two $\mathbf{w}$ in $\hat{P}$ are equally good;
3. for any $\mathbf{w}$ in $\hat{P}$, it is impossible to reduce the cost on one objective function without increasing its cost on the other objective function.

Therefore, the Pareto frontier is a complete set of equally "good" solutions that are superior to any other possible solutions. Despite this good property of Pareto frontier, computing Pareto frontier is unfortunately NP-hard in most cases. However, (Davidson et al. 2013) show that if a multi-objective optimization problem has graph-cut objective terms, then its approximated Pareto frontier can be solved efficiently with a generalized eigendecomposition problem.

Computing the Pareto Frontier via Generalized Eigendecomposition

For the optimization problem defined in formula (5), its Pareto frontier contains infinite number of solutions. In order to make the computation efficient, we made an approximation to original optimization problem, by introducing additional constraints to narrow down the search space. Particularly, we aim to find a subset of solutions in Pareto frontier which is distinctive enough. Therefore, we apply an mutually orthogonal constraint, which is defined as:

$$\hat{\Omega} \doteq \{\mathbf{w} \in \Omega \mid \forall \mathbf{w} \neq \mathbf{w}', X_S\mathbf{w} \perp D_S^{1/2}\mathbf{1}, X_T\mathbf{w} \perp D_T^{1/2}\mathbf{1}\} \tag{7}$$

Under an assumption that the null space of $X_S\bar{L_S}X_S^T$ and $X_T\bar{L_T}X_T^T$ do not overlap, the optimization turns into solving a generalized Hermitian definite pencil problem (Demmel et al. 2000). Then $\hat{\Omega}$ is the set of N eigenvectors of the generalized eigenvalue problem (Golub and Van Loan 1996).

$$X_S\bar{L_S}X_S^T\mathbf{w} = \lambda X_T\bar{L_T}X_T^T\mathbf{w} \tag{8}$$

However, in order to get a stable solution of the above eigenproblem, $X_T\bar{L_T}X_T^T$ is required to be non-singular (Golub and Van Loan 1996). Since in our applications, this does not always hold, in order to make the computation numerically stable, we adopt the SVD decomposition described as below.

SVD decomposition

Suppose we have the SVD decomposition of X_T as $X_T = U\Sigma V^T$. If we let $\bar{X_T} = U^TX_T = \Sigma V^T$ and multiply U^T to both sides of the equation, we can rewrite Eq. (8) as :

$$\begin{aligned} U^TX_S\bar{L_S}X_S^T\mathbf{w} &= \lambda U^TX_T\bar{L_T}{X_T}^T\mathbf{w} \\ &= \lambda\bar{X_T}\bar{L_T}{X_T}^T\mathbf{w} \end{aligned} \tag{9}$$

If we let $\mathbf{w} = U\mathbf{b}$, then we have:

$$\begin{aligned} U^TX_SL_SX_S^TU\mathbf{b} &= \lambda\bar{X_T}\bar{L_T}{X_T}^TU\mathbf{b} \\ &= \lambda\bar{X_T}\bar{L_T}{\bar{X_T}}^T\mathbf{b} \end{aligned} \tag{10}$$

Let $\bar{X_S} = U^TX_S$, then we rewrite Eq. (10) as:

$$\bar{X_S}\bar{L_S}{\bar{X_S}}^T\mathbf{b} = \lambda\bar{X_T}\bar{L_T}{\bar{X_T}}^T\mathbf{b} \tag{11}$$

whose optimal solution for $\mathbf{b}^*$'s can be still solved as the generalized eigenvalue problem. It is easy to check that $\bar{X_T}\bar{L_T}{\bar{X_T}}^T$ have a larger chance to be nonsingular so that the above eigen-problem has a stable closed form.

After we obtain $\mathbf{b}^*$, then $\mathbf{w}^*$ is obtained by solving a set of linear equations $\mathbf{w}^* = U\mathbf{b}^*$. The above function consists of $N-2$ orthogonal cuts in $\hat{\Omega}$. We further compute the Pareto frontier using the Algorithm 1.

Algorithm 1: Locality Preserving Projection for Domain Adaption with Multi-Objective Learning

```
input  : Data Matrix: $X_S, X_T$, Label: $y_S$
output: The set of Pareto optimal weights: $\hat{P}$
Compute the normalized graph Laplacians $\bar{L_S}$, $\bar{L_T}$, and
  compute the SVD decomposition for $X_S = USV$.
Solve the generalized eigenvalue problem:
  $U^TX_S\bar{L_S}X_S^TU\mathbf{b} = \lambda\bar{X_T}\bar{L_T}{\bar{X_T}}^T\mathbf{b}$.
Let $\mathbf{w} = U^T\mathbf{b}$, Normalize all $\mathbf{w}$'s such that $\mathbf{w}^T\mathbf{w} = 1$.
Let $\hat{P}$ be the set of all the $\mathbf{w}$, excluding the two
  associated with eigenvalue 0 and $\infty$.
for all $\mathbf{w}$ in $\hat{P}$ do
    for all $\mathbf{w}'$ in $\hat{P}$ do
        if $\mathbf{w}$ is a Pareto improvement over $\mathbf{w}'$ then
            remove $\mathbf{w}'$ from $\hat{P}$;
            continue;
        if $\mathbf{w}'$ is a Pareto improvement over $\mathbf{w}$ then
            remove $\mathbf{w}$ from $\hat{P}$;
            break;
```

Approximation Bound for Our Algorithms

As described above, we compute the orthogonal Pareto frontier as an approximation to the Pareto frontier. Here we create an upper bound on how far a point in the Pareto frontier can be to the orthogonal Pareto frontier. Let $\hat{\Omega} = \{\hat{\mathbf{b}}_i\}_{i=1}^{N-2}$ and $\hat{B} = (\hat{\mathbf{b}}_1, \cdots, \hat{\mathbf{b}}_{N-2})$. Any $\mathbf{b} \in \Omega$ can be represented by a linear combination of $\hat{\mathbf{b}}_i$'s: $\mathbf{b} = \hat{B}\mathbf{a}$, where $\mathbf{a} = (a_1, a_2, \cdots, a_{N-2})^T$. According to (Davidson et al. 2013), we can derive a lower-bound for $||\mathbf{a}||$.

$$||\mathbf{a}||^2 \geq 1/\sigma_{max}^2(\hat{B}) \tag{12}$$

where $1/\sigma_{max}^2(\hat{B})$ is the largest singular value of $\hat{B}$. The larger $1/\sigma_{max}^2(\hat{B})$ is, the closer the two costs on the Pareto frontier and orthogonal Pareto frontier. This effectively bounds the difference between the costs of the cuts on the Pareto frontier and those on the orthogonal Pareto frontier.

Empirical Study

In this section, results of our analysis of locality preserving projection for domain adaptation with multi-objective learning are presented. First, the data sets and the experiment settings used in this analysis are briefly described. Second, we analyze the classification accuracy for different domain adaptation algorithms on real world data sets. We aim to answer the following questions: (1) how does our algorithms perform on data sets with different distributions on training and test? (2) how does it compare to other domain adaptation algorithms?

Data description and experiment setup

The data set we evaluate first is the USPS handwritten digit database (Hull 1994). We extract two data sets from 9298 16x16 handwritten digit data sets. The data set 'USPS1' is constructed as follows: the source domain contains all the handwritten digits of '1's which are labeled '+1', all the handwritten digits of '8's which are labeled '-1'. The target domain includes all handwritten digits

’7’s and ’3’s with no labels. The data set ’USPS2’ is constructed in a similar way as that for ’USPS1’. The source domain contains all handwritten digits ’7’s with label ’+1’ and ’8’s with label ’-1’. The target domain includes all handwritten digits ’2’s and ’3’s with no labels.

We then evaluate our algorithm on the 14 tumor data sets which were published by Ramaswamy et al. (Statnikov et al. 2005), and we downloaded them in the preprocessing version from Statnikov (Statnikov et al. 2005). The data sets contain 14 different human tumor types and 12 normal types. Each type of tumor have only 10s order of subjects and 15009 genes. We extract three transfer learning data sets by coupling normal and tumor samples from the same tissue type together. The details of each data sets are as follows. For ’Bladder-Uterus’, the source domain contains all normal and disease samples with labels extracted from bladder tissue. The target domain includes all normal and disease samples without labels extracted from uterus tissue. For ’Prostate-Uterus’, the source domain contains all normal and disease samples with labels extracted from prostate tissue. The target domain includes all normal and disease samples without labels extracted from uterus tissue. For ’Uterus-Pancreas’, the source domain contains all normal and disease samples with labels extracted from uterus tissue. The target domain includes all normal and disease samples without labels extracted from pancreas tissue. We aim to predict whether a sample in the target domain is normal or disease given that samples in the source domain with labels.

At last, we evaluate our algorithm on the Lung tumor and brain tumor data sets downloaded from (Statnikov et al. 2005). The source domain for ’Lung1’ contains all samples in ’Adeno’ and ’Squamous’. The target domain for ’Lung1’ contains all samples in ’CIOD’ and ’SMCL’. In the same way, the source domain for ’Lung2’ contains all samples in ’Adeno’ and ’SMCL’. The target domain for ’Lung2’ contains all samples in ’CIOD’ and ’Squamous’. The source domain for ’Brain’ contains all samples in ’Medulloblastoma’ and ’Malignant glima’. The target domain for ’Brain’ contains all samples in ’AT/RT’ and ’PNET’. In this part, we aim to adapt the feature space between source domain and target domain and aim to separate two types of cancers.

The details of each data sets are listed in Table 1. It is easy to observe that there are several data sets with extremely small sample size and high dimensional feature space. And for the USPS data sets, the distribution for some features in the training data sets is significantly different from that in the test data sets. We want to see how our algorithms perform on all these various kinds of transfer learning data sets. To make comparisons, we implemented several state-of-art domain adaptation algorithms. GFK embeds the datasets into Grassmann manifolds and constructs geodesic flows between them to model domain shifts (Gong et al. 2012). TCA discovers a latent feature representation across domains by learning some transfer components in reproducing Kernel Hilbert space using maximum mean discrepancy. Our baseline method ’Original’ use the original features without learning a new representation for adaptation. We use 1-nearest neighbor classifier to do the classification and report the classification accuracy for each data set.

Experiment Results

The results are summarized in Table 2. From the table, it is easy to observe that our algorithm can achieve better classification on almost all data sets. Most importantly, our approach is more reliable in terms of performance than its competitors when the training and test data sets differ significantly.

For the gene expression data sets, which have very few of samples and high dimension of genes, our approach can find a linear projection which can enhance the classification accuracy. For the USPS data sets, there are quite a lot of features which have different distribution across the source and target domains. The new feature representation computed by GFK failed to adapt the source domain to the target domain.

Conclusion and Future Work

In this paper, we explore the locality preserving projection for domain adaptation with multi-objective learning. We propose multi-objective formulation for domain adaptation. The search space of our objective is the joint numerical range of two graphs. We find a relaxed mutually orthogonal optimal sets by using Pareto optimizations. The effectiveness of our approach is evaluated on the benchmark data sets with comparison to the state-of-the-art algorithms. The pragmatic benefits of our approach over existing domain adaptation algorithms are: 1) the users do not need to specify the tradeoff parameters; 2) the training and test data sets do not need to be similar to each other. Our algorithm can find the new feature representation which can effectively preserve the local structure.

Acknowledgments

This work was supported by National Science Foundation under Grants OIA-1027897 and IIS-1302164.

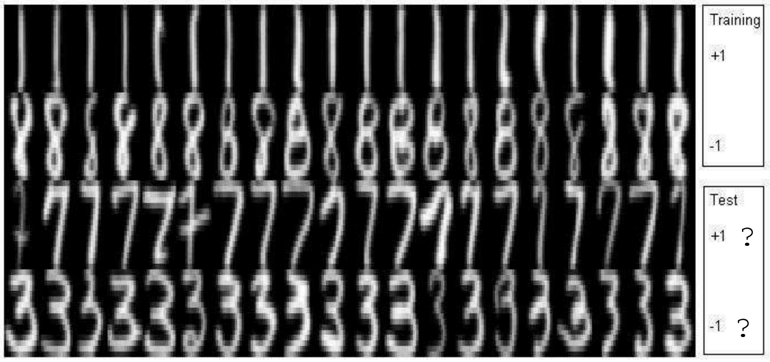

Figure 1: The training and test data sets for USPS handwritten digit: the first two rows represent the training data with labels, the third and fourth rows represent test data without labels.

Table 1: Summary of Data Sets

Datasets	Training	Testing	Features
	Pos vs Neg	Pos vs Neg	
$Lung_1$	20 : 6	17 : 21	12600
$Lung_2$	21 : 6	17 : 20	12600
$Brain$	7 : 14	14 : 15	10367
$USPS1$	664 : 731	1858 : 645	256
$USPS2$	644 : 731	542: 645	256
$Bladder - Uterus$	11 : 7	11 : 6	15009
$Prostate - Uterus$	14 : 9	11 : 6	15009
$Uterus - Pancreas$	11 : 6	11 : 10	15009

Datasets	Original	TCA	GFK	Our method
$Lung_1$	0.5263	0.6053	0.6053	**0.8158**
$Lung_2$	0.7027	0.6406	0.6406	**0.8919**
$Brain$	0.8966	0.8621	0.8966	**0.9655**
$USPS1$	0.8554	0.5610	0.6117	**0.9036**
$USPS2$	0.8569	0.7787	0.7889	**0.8833**
$Bladder - Uterus$	0.6534	0.6191	**0.7110**	0.7059
$Prostate - Uterus$	0.7059	0.7647	0.7647	**0.8235**
$Uterus - Pancreas$	0.7143	0.7143	0.7619	**0.8095**

Table 2: Performance comparison of classification accuracy of different domain adaptation algorithms on different datasets. The best results of each data set are highlighted in bold.

References

Blitzer, J.; McDonald, R. T.; and Pereira, F. 2006. Domain adaptation with structural correspondence learning. In *EMNLP*, 120–128.

Cai, D.; He, X.; Zhang, W. V.; and Han, J. 2007. Regularized locality preserving indexing via spectral regression. In *CIKM*, 741–750.

Chen, M.; Xu, Z. E.; Weinberger, K. Q.; and Sha, F. 2012. Marginalized denoising autoencoders for domain adaptation.

Chen, M.; Weinberger, K. Q.; and Blitzer, J. 2011. Co-training for domain adaptation. In *NIPS*, 2456–2464.

Davidson, I.; Qian, B.; Wang, X.; and Ye, J. 2013. Multi-objective multi-view spectral clustering via pareto optimization. In *SDM*, 234–242.

Demmel, J.; Dongarra, J.; Ruhe, A.; and van der Vorst, H. 2000. *Templates for the solution of algebraic eigenvalue problems: a practical guide*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics.

Fox, D., and Gomes, C. P., eds. 2008. *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*. AAAI Press.

Golub, G. H., and Van Loan, C. F. 1996. *Matrix computations (3rd ed.)*. Baltimore, MD, USA: Johns Hopkins University Press.

Gong, B.; Shi, Y.; Sha, F.; and Grauman, K. 2012. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, 2066–2073.

Gu, Q.; Danilevsky, M.; Li, Z.; and Han, J. 2012. Locality preserving feature learning. In *AISTATS*, 477–485.

He, X., and Niyogi, P. 2003. Locality preserving projections. In *NIPS*.

He, X.; Yan, S.; Hu, Y.; Niyogi, P.; and Zhang, H. 2005. Face recognition using laplacianfaces. *IEEE Trans. Pattern Anal. Mach. Intell.* 27(3):328–340.

He, X.; Cai, D.; and Niyogi, P. 2005. Laplacian score for feature selection. In *NIPS*.

Huang, J.; Smola, A. J.; Gretton, A.; Borgwardt, K. M.; and Schölkopf, B. 2006. Correcting sample selection bias by unlabeled data. In *NIPS*, 601–608.

Hull, J. J. 1994. A database for handwritten text recognition research. *IEEE Trans. Pattern Anal. Mach. Intell.* 16(5):550–554.

III, H. D. 2007. Frustratingly easy domain adaptation. In *ACL*.

Jiang, J., and Zhai, C. 2007a. Instance weighting for domain adaptation in nlp. In *ACL*.

Jiang, J., and Zhai, C. 2007b. A two-stage approach to domain adaptation for statistical classifiers. In *CIKM*, 401–410.

Kulis, B.; Saenko, K.; and Darrell, T. 2011. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *CVPR*, 1785–1792.

Mansour, Y.; Mohri, M.; and Rostamizadeh, A. 2008. Domain adaptation with multiple sources. In *NIPS*, 1041–1048.

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

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

Pan, S. J.; Kwok, J. T.; and Yang, Q. 2008. Transfer learning via dimensionality reduction. In *AAAI*, 677–682.

Statnikov, A. R.; Aliferis, C. F.; Tsamardinos, I.; Hardin, D. P.; and Levy, S. 2005. A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. *Bioinformatics* 21(5):631–643.