Transductive Domain Adaptation with Affinity Learning

Le Shu  
Computer and Information Sciences  
Temple University  
1925 N. 12 St. Philadelphia, PA  
le.shu@temple.edu

Longin Jan Latecki  
Computer and Information Sciences  
Temple University  
1925 N. 12 St. Philadelphia, PA  
latecki@temple.edu

ABSTRACT
We study the problem of domain adaptation, which aims to adapt the classifiers trained on a labeled source domain to an unlabeled target domain. We propose a novel method to solve domain adaptation task in a transductive setting. The proposed method bridges the distribution gap between source domain and target domain through affinity learning. It exploits the existence of a subset of data points in target domain which distribute similarly to the data points in the source domain. These data points act as the bridge that facilitates the data similarities propagation across domains. We also propose to control the relative importance of intra- and inter- domain similarities to boost the similarity propagation. In our approach, we first construct the similarity matrix which encodes both the intra- and inter- domain similarities. We then learn the true similarities among data points in joint manifold using graph diffusion. We demonstrate that with improved similarities between source and target data, spectral embedding provides a better data representation, which boosts the prediction accuracy. The effectiveness of our method is validated on standard benchmark datasets for visual object recognition (multi-category).

Categories and Subject Descriptors
H.4 [Information Systems Applications]: Miscellaneous;  
D.2.8 [Software Engineering]: Metrics—complexity measures, performance measures

Keywords
Domain Adaptation, Affinity Learning

1. INTRODUCTION
In recent years, domain adaptation has gained significant attention in many areas of applied machine learning, including bioinformatics, speech and language processing, computer vision etc. In these practical problems, given that the instances in the training and testing domains may be drawn from different distributions, traditional learning method cannot achieve good performance on the new domain. Domain adaptation algorithms are therefore designed to bridge the distribution gap between training (source) data and testing (target) data. Domain adaptation methods seek to eliminate the difference between source and target distributions.

In this paper, we propose a transductive method to explicitly improve intra- and inter- domain similarities. Our contribution is two-fold: first, we perform affinity learning via graph diffusion to bridge the distribution gap of source and target domain. The key idea is to exploit the existence of a subset of data points in the target domain which distributes similarly to the data points in the source domain. We denote this subset of data points as Bridge Points (BP). Through graph diffusion, we propagate the similarities between BP and other data points in the target domain, as well as the similarities between BP and data points in the source domain. Affinity learning is able to give robust pairwise similarities of data points, since all paths between all pairs of data points are considered. In this way, affinity learning can bridge distribution gap of source and target domain. As our experimental results clearly demonstrate, our assumption that part of the target data is similar to part of the source data is often satisfied by real world data sets. Figure 1 illustrates our motivation. Our second contribution is to adjust the intra- and inter- domain similarities. The intuition is that data points in the same domain are often more similar to each other than to those in different domain. In graph diffusion process, this makes the similarity propagation from data points in source domain to data points in target domain ineffective. Therefore, the proposed adjustment of the intra- and inter- domain similarities is a key step in making the affinity propagation successful. We balance the intra- and inter- domain edges by picking equal number of nearest neighbors in source and target domain for each data point and also re-weight intra- and inter- domain edges.

In summary, given the similarity matrix of source and target data, the procedure of our framework includes the following key steps: 1) Similarity Adjustment: re-weight intra- and inter- domain similarities. 2) Affinity Learning: iteratively learn similarities in joint geometric structure via Tensor Product Graph Diffusion (TPGD)[18]. 3) Spectral Embedding: apply spectral embedding on diffusion matrix to get a low-dimensional representation. In this paper, we use Tensor Product Graph Diffusion(TPGD) [18] to capture the joint manifold structure for the source and target domain. As demonstrated in [18], TPGD can robustly discover
3. PROPOSED APPROACH

We assume that our data originate from two domains, Source (S) and Target (T). Source data \( D_S = \{ (x_S^1, y_S^1), (x_S^2, y_S^2), \ldots, (x_S^N, y_S^N) \} \) is fully labeled, each pair \( (x_S^i, y_S^i) \) lies in \( \mathbb{R}^d \times y \) space. The source data are sampled from some distribution \( P_S(X, Y) \). The target data has equal dimension \( d \) as the source data but is sampled from a different distribution \( P_T(X, Y) \). We denote the target data as \( D_T = \{ (x_T^1, ?), (x_T^2, ?), \ldots, (x_T^N, ?) \} \), whose labels are unknown. Given \( D_S \) and \( D_T \), our goal is to infer the class labels of data points in \( D_T \).

In the rest of this section, we first describe how to construct the transition matrix \( P \) for source and target data jointly. We then iteratively learn the joint geometric structure and capture true similarities among data points in source and target domain. After we get the diffusion matrix, we compute the Laplacian graph and solve the smallest \( L \) eigenvalues. The source data are sampled from some distribution \( P_S(X, Y) \). The target data has equal dimension \( d \) as the source data but is sampled from a different distribution \( P_T(X, Y) \). We denote the target data as \( D_T = \{ (x_T^1, ?), (x_T^2, ?), \ldots, (x_T^N, ?) \} \), whose labels are unknown. Given \( D_S \) and \( D_T \), our goal is to infer the class labels of data points in \( D_T \).

In the rest of this section, we first describe how to construct the transition matrix \( P \) for source and target data jointly. We then iteratively learn the joint geometric structure and capture true similarities among data points in source and target domain. After we get the diffusion matrix, we compute the Laplacian graph and solve the smallest \( L \) eigenvalues. The source data are sampled from some distribution \( P_S(X, Y) \). The target data has equal dimension \( d \) as the source data but is sampled from a different distribution \( P_T(X, Y) \). We denote the target data as \( D_T = \{ (x_T^1, ?), (x_T^2, ?), \ldots, (x_T^N, ?) \} \), whose labels are unknown. Given \( D_S \) and \( D_T \), our goal is to infer the class labels of data points in \( D_T \).

In the rest of this section, we first describe how to construct the transition matrix \( P \) for source and target data jointly. We then iteratively learn the joint geometric structure and capture true similarities among data points in source and target domain. After we get the diffusion matrix, we compute the Laplacian graph and solve the smallest \( L \) eigenvalues. The source data are sampled from some distribution \( P_S(X, Y) \). The target data has equal dimension \( d \) as the source data but is sampled from a different distribution \( P_T(X, Y) \). We denote the target data as \( D_T = \{ (x_T^1, ?), (x_T^2, ?), \ldots, (x_T^N, ?) \} \), whose labels are unknown. Given \( D_S \) and \( D_T \), our goal is to infer the class labels of data points in \( D_T \).

**Figure 1:** Schematic illustration of utilizing affinity learning for unsupervised domain adaptation. (a) Data points in source domain. Each color represents one class. (b) Data points in source and target domains. Solid shapes represent the data points in the source domain, hollow shapes represent the data points in the target domain. Black circles mark the Bridge Points, which is subset of data instances in the target domain and have similar distribution as data instances in the source domain. (c) The lines connect points in the target domain to their nearest neighbors in the source domain with the original similarities. (d) The lines connect points in the target domain to their nearest neighbors in the source domain after the affinity learning.
domain and $A_{TT}$ encodes the similarities between data points in target domain. $D_S$ and $D_T$ are the diagonal matrices of the row sums of $A_S$ and $A_T$.

To compute $A_{SS}$, we take advantage of the available labels of source data, and define "closeness" in a supervised manner, i.e., nodes $i$ and $j$ are connected if $x_i$ and $x_j$ share the same label. The similarity matrix $A_{SS}$ is defined as follows:

$$A_{SS}(i,j) = \begin{cases} \frac{x_i^T x_j}{\|x_i\| \|x_j\|} & \text{if } x_i^S \text{ and } x_j^S \text{ share the same label} \\ 0 & \text{otherwise} \end{cases}$$

Here, we use $N_p$ to denote the $p$ nearest neighbors. In the source domain, the label information is embedded into the similarity matrix. While we compute the similarity matrix $A_{SS}$ with the cosine similarity measure, other similarity measures may also be applicable.

To compute $A_{TT}$, since the labels of target data are not available, we define "closeness" in an unsupervised manner, i.e., nodes $i$ and $j$ are connected if $i$ is among $p$ nearest neighbors of $j$ or $j$ is among $p$ nearest neighbors of $i$. Formally, we have:

$$A_{TT}(i,j) = \begin{cases} \frac{x_i^T x_j}{\|x_i\| \|x_j\|} & \text{if } x_i^T \in N_p(x_j^T) \text{ or } x_j^T \in N_p(x_i^T) \\ 0 & \text{otherwise} \end{cases}$$

where $N_p(x_i^T)$ is the set of $p$ nearest neighbors of $x_i^T$.

Similarly cross-domain transition probability matrices $P_{ST}$ and $P_{TS}$ are computed as follows:

$$P_{ST} = D_{ST}^{-1} A_{ST} \quad P_{TS} = D_{TS}^{-1} A_{TS}$$

where $A_{ST}$ denotes the cross-domain similarities. $D_{ST}$ and $D_{TS}$ are the diagonal matrices of the row sums of $A_{ST}$ and $A_{TS}$. $A_{ST}$ is computed as follows:

$$A_{ST}(i,j) = \begin{cases} \frac{x_i^T x_j}{\|x_i\| \|x_j\|} & \text{if } x_i^T \in N_p(x_j^T) \text{ or } x_j^T \in N_p(x_i^T) \\ 0 & \text{otherwise} \end{cases}$$

To summarize, we introduce two major differences in the transition matrix construction process which are tailored for unsupervised domain adaptation task: First, we add supervised information to the similarity matrix $A_{SS}$ to remove noisy entries in $A_{SS}$. Second, we control the relative importance of intra-domain and inter-domain transition probabilities. When building KNN connected graph, we balance the intra- and inter- domain edges by picking equal number of nearest neighbors in source and target domain for each data point. We also perform a reweighting on the intra- and inter-domain similarities. As demonstrated by our experiments in Section 4.1, these two steps greatly boost the performance of affinity learning which results in a better data representation, and therefore a higher prediction accuracy.

### 3.2 Diffusion Process on Tensor Product Graph

In this section we review tensor product graph diffusion process introduced in [18]. Given the edge (transition probability matrix) $P$, we define $Q^{(1)} = P$ and

$$Q^{(t+1)} = P Q^{(t)} P^T + I,$$

where $I$ is the identity matrix. We iterate (7) until convergence. Let us denote the limit matrix by $Q^* = \lim_{t \to \infty} Q^{(t)}$.

A closed form expression for $Q^*$ is as follows:

$$\lim_{t \to \infty} Q^{(t)} = Q^* = P^* = \frac{1}{\sum_{i=0}^{1} P^1} \text{vec}(I). \quad (8)$$

The proof of the convergence of (7) and closed form equation can be found in [18], where $P$ is the tensor product of $P$ with itself. Since $Q^* = P^*$, we obtain that the iterative algorithm on $Q$ defined by (7) yields the same similarities as the TPG diffusion process on $P$ for a sufficient number of iterations.

### 4. EMPIRICAL STUDY

In this section, we present our experimental results on visual object recognition tasks. We set $K$ and $p$ in our method through cross-validation based on classification error of data samples in source domains. We first compare to the baseline approach and evaluate the performance gain at each step and give detailed analysis. This provides clear insight about the merits of the proposed method. Our results on benchmark datasets are also favorable when compared to several state-of-the-art domain adaptation methods.

We perform experiments using 4 object recognition datasets, which includes: Amazon, Webcam, DSLR. These three datasets are first introduced in [16]. Additionally, we use Caltech-256 in [6] as the fourth dataset to further evaluate the proposed methods. Each dataset is treated as a domain and 10 common object categories are extracted. We downloaded the processed datasets with SURF features from [5]. We conduct each experiment using every pair of source and target dataset. We report the recognition accuracy on every pair of source and target dataset.

#### 4.1 Performance Analysis

As the baseline approach, we adopt the original features and train a linear SVM model on source domain. To illustrate the significance of performance gain using affinity learning to facilitate domain adaptation, we study four variants of our method. In the first variant, we apply spectral embedding directly to the original similarity matrix. In the second variant, we add intra- and inter- domain similarities before applying spectral embedding. In the third variant, we apply graph diffusion to the original similarity matrix before applying spectral embedding. In the final variant, we use all components together which is the proposed approach.

We compare the recognition accuracy of the baseline approach and the 4 variants in Table 1. We can see that low-dimensional feature representation obtained by spectral embedding can preserve most of the information for each dataset, whose accuracy is comparable to the baseline, but with no improvement. If we adjust intra- and inter- domain similarities and apply spectral embedding, the average recognition accuracy improves $7.5\%$ compared to that of the baseline. If we apply affinity learning and spectral embedding together, the average recognition accuracy improves $4.6\%$ compared to that of the baseline. If we combine adjusting intra- and inter- domain similarities and affinity learning through graph diffusion, the performance improves $10.0\%$ compared to that of the baseline method. Overall, these results demonstrate that adjusting intra- and inter-domain similarities can facilitate the affinity learning, and affinity learning can provide more reliable affinities for data points in joint manifold.
Table 1: Recognition Accuracy on 8 pairs of datasets, where: C : Caltech, A : Amazon, W : Webcam, D : DSLR. SE: spectral embedding; SA: similarity adjustment; TPGD: tensor product graph diffusion; Our Method = SA+TPGD+SE.

<table>
<thead>
<tr>
<th></th>
<th>C-D</th>
<th>A-D</th>
<th>A-W</th>
<th>D-A</th>
<th>D-W</th>
<th>W-A</th>
<th>W-C</th>
<th>C-D</th>
<th>C-W</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>40.8</td>
<td>41.7</td>
<td>32.3</td>
<td>73.6</td>
<td>34.1</td>
<td>29.9</td>
<td>42.0</td>
<td>40.7</td>
<td>41.9</td>
<td></td>
</tr>
<tr>
<td>SE</td>
<td>40.1</td>
<td>40.7</td>
<td>34.4</td>
<td>66.4</td>
<td>35.7</td>
<td>28.7</td>
<td>47.1</td>
<td>43.7</td>
<td>42.1</td>
<td></td>
</tr>
<tr>
<td>SA + SE</td>
<td>45.8</td>
<td>44.4</td>
<td>41.1</td>
<td>89.5</td>
<td>38.7</td>
<td>35.0</td>
<td>52.8</td>
<td>49.1</td>
<td>49.6</td>
<td></td>
</tr>
<tr>
<td>TPGD + SE</td>
<td>43.9</td>
<td>41.0</td>
<td>34.8</td>
<td>74.9</td>
<td>39.8</td>
<td>35.2</td>
<td>54.1</td>
<td>48.5</td>
<td>46.5</td>
<td></td>
</tr>
<tr>
<td>Our Method</td>
<td>50.3</td>
<td>49.0</td>
<td>40.5</td>
<td>92.2</td>
<td>39.7</td>
<td>36.8</td>
<td>55.4</td>
<td>52.9</td>
<td>52.1</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Recognition Accuracy on 8 pairs of datasets, where C : Caltech, A : Amazon, W : Webcam, D : DSLR.

<table>
<thead>
<tr>
<th></th>
<th>C-D</th>
<th>A-D</th>
<th>A-W</th>
<th>D-A</th>
<th>D-W</th>
<th>W-A</th>
<th>W-C</th>
<th>C-D</th>
<th>C-W</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>40.8</td>
<td>41.7</td>
<td>32.3</td>
<td>73.6</td>
<td>34.1</td>
<td>29.9</td>
<td>42.0</td>
<td>40.7</td>
<td>41.9</td>
<td></td>
</tr>
<tr>
<td>TCA[12]</td>
<td>36.3</td>
<td>27.8</td>
<td>28.7</td>
<td>82.0</td>
<td>24.2</td>
<td>22.5</td>
<td>45.2</td>
<td>32.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KMM[7]</td>
<td>42.7</td>
<td>42.4</td>
<td>36.0</td>
<td>83.0</td>
<td>31.9</td>
<td>29.0</td>
<td>53.5</td>
<td>45.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GFK[5]</td>
<td>42.7</td>
<td>40.7</td>
<td>36.2</td>
<td>78.3</td>
<td>31.8</td>
<td>30.9</td>
<td>43.3</td>
<td>44.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LandMark[4]</td>
<td>47.1</td>
<td>46.1</td>
<td>33.4</td>
<td>78.0</td>
<td>40.2</td>
<td>35.4</td>
<td>57.3</td>
<td>49.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Our Method</td>
<td>50.3</td>
<td>49.0</td>
<td>40.5</td>
<td>92.2</td>
<td>39.7</td>
<td>36.8</td>
<td>55.4</td>
<td>52.9</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.2 Comparison with State-of-the-art Methods

We compare the proposed method to several state-of-the-art methods: KMM [7], TCA [12], GFK [5], LandMark [4]. Table 2 summarizes accuracy of object recognition on 8 pairs of source and target domains obtained from the four datasets. For the compared methods, most results are quoted from [4], except for D-A and D-W which we generated using the code downloaded from authors’ websites. The average recognition accuracy of our method improves 3.7% when compared to that of the second best method ‘LandMark’. Our method performs the best on 6 out of 8 pairs of domains.

5. CONCLUSIONS

We propose a novel transductive domain adaptation method. Empirical results clearly demonstrate that it outperforms state-of-the-art methods.

6. ACKNOWLEDGMENTS

This work was in part supported by NSF under Grants OIA-1027897 and IIS-1302164.

7. REFERENCES