# 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 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 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

*CIKM* '15, October 19-23, 2015, Melbourne, VIC, Australia
©2015 ACM ISBN 978-1-4503-3794-6/15/10$15.00
DOI: http://dx.doi.org/10.1145/2806416.2806643.

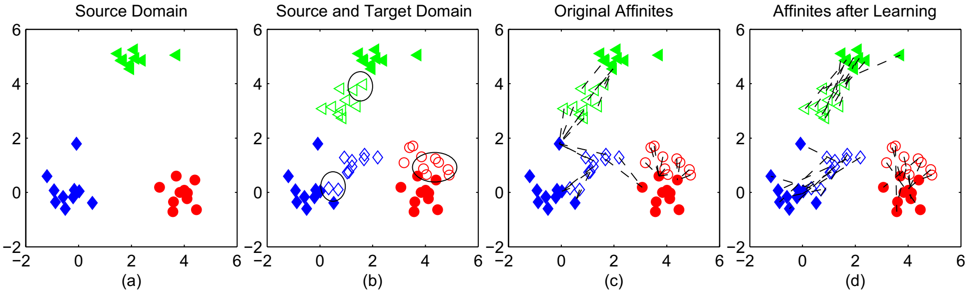

Figure 1: Schematic illustration of utilizing affinity learning for unsupervised domain adaptation. (a) Data points in source domain. Each color represents one class. (b) Data points in source and target domains. Solid shapes represent the data points in the source domain, hollow shapes represent the 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.

the true, underlying manifold structure in image retrieval. We utilize TPGD to learn joint geometric structure in the context of domain adaptation when training and testing are drawn from different distributions.

The rest of the paper is organized as follows. We first give a brief review of related works in Section 2. In Section 3, we describe the proposed affinity learning for domain adaptation task. In particular, we describe how to construct the transition probability matrix with similarity adjustment. We also show how to perform graph diffusion on a tensor product graph to obtain robust similarities. In Section 4, we present our experimental results on benchmark datasets and compare it to several state-of-the-art methods. Finally, we come to the conclusion is in Section 5.

## 2. RELATED WORK

Domain adaptation has been extensively studied in many research areas [13, 8, 2, 19]. Domain adaptation can be categorized into three types. The first type are self-labeling approaches, which include self-training [14] and co-training [2]. The second type of algorithms proposes to weight or select training instances to minimize the discrepancy distance [7, 9]. Our work belongs to the third type, which aims at finding ”good” feature representations to minimize domain divergence and classification error, such as [1, 5, 12]. In particular, for object recognition application in computer vision, many works have been proposed to learn new feature representation, such as [11, 16, 4]. Compared to existing approaches, our method focus on affinity learning to bridge the distribution gap between source and target domain.

While our work share some common components compared to graph-based semi-supervised method, such as [20] where graph is used to propagate labels, the key difference is that we aim to solve the domain adaptation problem and our goal is to use affinity learning to improve the noisy pairwise similarities due to domain shift. That motivates us to reweight the inter- and intra- domain edges, and use spectral embedding to obtain the low dimensional domain-invariant data representation.

There are also several works attempting to solve transfer learning in a transductive setting [15, 10, 3]. They apply label propagation to zero-shot and few-shot learning based on attribute graph or semantic graph. [17] exploits the mixture distribution to refine the classification labels. These work did not try to improve pair-wise similarities.

## 3. PROPOSED APPROACH

We assume that our data originate from two domains, Source (S) and Target (T). Source data $D_S = \{(x_S^1, y_S^1), (x_S^2, y_S^2), \cdots, (x_S^{N_S}, y_S^{N_S})\}$ is fully labeled, each pair $(x_S^i, y_S^i)$ lies in $R^d \times y$ 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 $D_T = \{(x_T^1, ?), (x_T^2, ?), \cdots, (x_T^{N_T}\ , ?)\}$, 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 $K$ eigenvectors to obtain a new feature representation of data points in source domain and target domain. After that, any classification approach can be adopted to predict labels for target data. In this work, we choose SVM classifier with linear kernel.

### 3.1 Cross-domain Graph Construction

The goal of this section is to construct the transition matrix of a graph $G$ whose nodes consist of data points in both source and target domain. We use $P_{SS}$ and $P_{TT}$ to denote the transition probability matrices of data points in the source and target domains respectively. $P_{ST}$ and $P_{TS}$ denote the transition probability matrix of data points across domains. We construct the overall transition matrix as follows:

$$P = \begin{bmatrix} \beta P_{SS} & (1-\beta)P_{ST} \\ (1-\beta)P_{TS} & \beta P_{TT} \end{bmatrix} \qquad (1)$$

where $\beta$ controls the relative importance of the intra- and the inter- domain transition probabilities and $\beta \in [0, 1]$. Empirically, $\beta$ can be set by solving $\frac{\beta}{1-\beta} = \frac{2N_S N_T}{N_S^2 + N_T^2}$, which calibrates the average inter- and intra- domain edge weights to be close.

$P_{SS}$ and $P_{TT}$ are row-wise normalized similarity matrices which are computed as:

$$P_{SS} = D_{SS}^{-1} A_{SS} \quad P_{TT} = D_{TT}^{-1} A_{TT} \qquad (2)$$

where $A_{SS}$ is the similarity matrix of data points in source domain and $A_{TT}$ encodes the similarities between data points in target domain. $D_{SS}$ and $D_{TT}$ are the diagonal matrices of the row sums of $A_{SS}$ and $A_{TT}$.

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:

$$A_{SS}(i,j) = \begin{cases} \frac{x_S^j \cdot x_S^i}{\|x_S^j\| \cdot \|x_S^i\|} & \text{if } x_S^i \text{ and } x_S^j \text{ share the same label} \\ & \text{ and } x_S^i \in N_p(x_S^j) \text{ or } x_S^j \in N_p(x_S^i) \\ 0 & \text{otherwise.} \end{cases}, \tag{3}$$

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:

$$A_{TT}(i,j) = \begin{cases} \frac{x_T^j \cdot x_T^i}{\|x_T^j\| \cdot \|x_T^i\|} & \text{if } x_T^i \in N_p(x_T^j) \text{ or } x_T^j \in N_p(x_T^i) \\ 0 & \text{otherwise.} \end{cases}, \tag{4}$$

where $N_p(x_T^i)$ is the set of $p$ nearest neighbors of $x_T^i$.

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

$$P_{ST} = D_{ST}^{-1} A_{ST} \quad P_{TS} = D_{TS}^{-1} A_{TS} \tag{5}$$

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:

$$A_{ST}(i,j) = \begin{cases} \frac{x_S^i \cdot x_T^j}{\|x_S^i\| \cdot \|x_T^j\|} & \text{if } x_S^i \in N_p(x_T^j) \text{ or } x_T^j \in N_p(x_S^i) \\ 0 & \text{otherwise.} \end{cases}, \tag{6}$$

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

$$Q^{(t+1)} = P\ Q^{(t)}\ P^T + I, \tag{7}$$

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:

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

The proof of the convergence of (7) and closed form equation can be found in [18], where $\mathbb{P}$ 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 $\mathbb{P}$ 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 reweighting 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 put all components together which is the proposed approach.

We compare the recognition accuracy of the baseline approach and the 4 variants in Tabel 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.

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

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

| % | A-D | A-W | D-A | D-W | W-A | W-C | C-D | C-W |
|---|---|---|---|---|---|---|---|---|
| Baseline | 40.8 | 41.7 | 32.3 | 73.6 | 34.1 | 29.9 | 42.0 | 40.7 |
| TCA[12] | 36.3 | 27.8 | 28.7 | 82.0 | 24.2 | 22.5 | 45.2 | 32.5 |
| KMM[7] | 42.7 | 42.4 | 36.0 | 83.0 | 31.9 | 29.0 | 53.5 | 45.8 |
| GFK[5] | 42.7 | 40.7 | 36.2 | 76.3 | 31.8 | 30.9 | 43.3 | 44.7 |
| LandMark[4] | 47.1 | 46.1 | 33.4 | 78.0 | **40.2** | 35.4 | **57.3** | 49.5 |
| Our Method | **50.3** | **49.0** | **40.5** | **92.2** | 39.7 | **36.8** | 55.4 | **52.9** |

## 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

[1] John Blitzer, Ryan T. McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In *EMNLP*, 2006.

[2] Minmin Chen, Kilian Q. Weinberger, and John Blitzer. Co-training for domain adaptation. In *NIPS*, pages 2456–2464, 2011.

[3] Yanwei Fu, Timothy M. Hospedales, Tao Xiang, Zhen-Yong Fu, and Shaogang Gong. Transductive multi-view embedding for zero-shot recognition and annotation. In *ECCV*, pages 584–599, 2014.

[4] Boqing Gong, Kristen Grauman, and Fei Sha. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *ICML*, pages 222–230, 2013.

[5] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, pages 2066–2073, 2012.

[6] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.

[7] Jiayuan Huang, Alexander J. Smola, Arthur Gretton, Karsten M. Borgwardt, and Bernhard Schölkopf. Correcting sample selection bias by unlabeled data. In *NIPS*, pages 601–608, 2006.

[8] Hal Daumé III. Frustratingly easy domain adaptation. In *ACL*, 2007.

[9] Jing Jiang and ChengXiang Zhai. Instance weighting for domain adaptation in nlp. In *ACL*, 2007.

[10] Yu-Gang Jiang, Jun Wang, Shih-Fu Chang, and Chong-Wah Ngo. Domain adaptive semantic diffusion for large scale context-based video annotation. In *ICCV*, pages 1420–1427, 2009.

[11] Brian Kulis, Kate Saenko, and Trevor Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *CVPR*, 2011.

[12] Sinno Jialin Pan, Ivor W. Tsang, James T. Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. In *IJCAI*, pages 1187–1192, 2009.

[13] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, 22(10):1345–1359, 2010.

[14] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y. Ng. Self-taught learning: transfer learning from unlabeled data. In *ICML*, 2007.

[15] Marcus Rohrbach, Sandra Ebert, and Bernt Schiele. Transfer learning in a transductive setting. In *NIPS*, pages 46–54, 2013.

[16] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, 2010.

[17] Dikan Xing, Wenyuan Dai, Gui-Rong Xue, and Yong Yu. Bridged refinement for transfer learning. In *PKDD*, pages 324–335, 2007.

[18] Xingwei Yang, Lakshman Prasad, and Longin Jan Latecki. Affinity learning with diffusion on tensor product graph. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(1):28–38, 2013.

[19] Chao Zhang, Lei Zhang, Wei Fan, and Jieping Ye. Generalization bounds for representative domain adaptation. *CoRR*, abs/1401.0376, 2014.

[20] Xiaojin Zhu, Zoubin Ghahramani, and John D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, 2003.