Linking Heterogeneous Input Features with Pivots for Domain Adaptation

Guangyou Zhou[1], **Tingting He**[1], **Wensheng Wu**[2], **and Xiaohua Tony Hu**[1]
[1] School of Computer, Central China Normal University, Wuhan 430079, China
[2] Computer Science Department, University of Southern California, Los Angeles, CA
{gyzhou,tthe,huxiaohua}@mail.ccnu.edu.cn wuwens@gmail.com

Abstract

Sentiment classification aims to automatically predict sentiment polarity (e.g., positive or negative) of user generated sentiment data (e.g., reviews, blogs). In real applications, these user generated sentiment data can span so many different domains that it is difficult to manually label training data for all of them. Hence, this paper studies the problem of domain adaptation for sentiment classification where a system trained using labeled reviews from a source domain is deployed to classify sentiments of reviews in a different target domain. In this paper, we propose to link heterogeneous input features with pivots via joint non-negative matrix factorization. This is achieved by learning the domain-specific information from different domains into unified topics, with the help of pivots across all domains. We conduct experiments on a benchmark composed of reviews of 4 types of Amazon products. Experimental results show that our proposed approach significantly outperforms the baseline method, and achieves an accuracy which is competitive with the state-of-the-art methods for sentiment classification adaptation.

1 Introduction

With the rise of social media (e.g., blogs, social networks, etc.), more and more user generated sentiment data have been shared on the web. They exist in the form of user reviews on shopping or opinion sites, in posts of blogs or customer feedbacks. This has created a surge of research in sentiment classification (or sentiment analysis), which aims to automatically determine the sentiment polarity (e.g., positive or negative) of user generated sentiment data (e.g., reviews, blogs).

Machine learning algorithms have been proved promising and widely used for sentiment classification [Pang *et al.*, 2002; Pang and Lee, 2008; Liu, 2012; Zhou *et al.*, 2014b; 2015]. Applications to many different domains have been presented, ranging from movie reviews [Pang *et al.*, 2002] and congressional floor debates [Thomas *et al.*, 2006] to product recommendations [Snyder and Barzilay, 2007; Blitzer *et al.*, 2007].

These user generated sentiment data can span so many different domains that it is difficult to manually label training data for all of them. In real applications, users may use different vocabularies (or domain-specific words) to express sentiment in different domains. For example, in the *Electronic* domain the words "durable" and "light" are used to express positive sentiment, whereas "expensive" and "short battery" often indicate negative sentiment. On the other hand, in the *Book* domain the words "exciting" and "thriller" are used to express positive sentiment, whereas the words "boring" and "lengthy" usually express negative sentiment. In all these examples, the source domain and the target domain have different input feature spaces - we refer to this problem setting as `Heterogeneous Input Features` (HIF). As a result, a classifier trained on one domain might not perform well on a different domain because it fails to learn the sentiment of domain-specific words. Thus domain adaptation is highly desirable to reduce the gap between domains and manually labeling cost.

Domain adaptation also poses new challenges from the algorithm perspective since multiple domains seem to be totally uncorrected with each other if their examples are in different input features. Here our key observation is that in many real applications, there might exist some correspondence among certain input dimensions of different domains. In the example of domain adaptation for sentiment classification, some domain-specific words are linked via the same sentiment polarity. The correspondence across different input features provides an important connection among multiple domains. In this paper, such correspondence is represented by `pivots`, which consists of tuples of input dimensions from multiple domains bearing the same correspondence.

Inspired by the above observation, we propose a novel learning framework, called Linking Heterogeneous Input Features via Pivots via Joint Non-negative Matrix Factorization (PJNMF). PJNMF assumes that there exists a set of domain-specific features for each of the domain, and there also exists a set of pivot features for the source and target domains. In PJNMF, documents in each domain are represented as a term-document matrix. The term-document matrix is then approximated as the product of two matrices: one matrix represents the domain-specific features as well as the pivot features, the second matrix denotes the document representation based on the latent features. An objective function

is defined to learn the mappings from the heterogeneous input features to a common space and the prediction model in this space. In this framework, to make full use of the pivot information, we enforce the input dimensions mapped to the same pivot be projected to the common space in a similar way. To demonstrate the effectiveness of the proposed approach, we conduct experiments on a benchmark composed of reviews of 4 types of Amazon products. Experimental results show that the proposed approach significantly outperforms several baselines, and achieves an accuracy which is competitive with best known method for sentiment classification adaptation.

The remainder of this paper is organized as follows. Section 2 introduces the related work. Section 3 describes our proposed linking heterogeneous input features via pivots for domain adaptation. Section 4 presents the experimental results. In section 5, we conclude with ideas for future research.

2 Related Work

Sentiment classification has gained widely interest in natural language processing (NLP) community, we point the readers to recent books [Pang and Lee, 2008; Liu, 2012] for an in-depth survey of literature on sentiment analysis. In this section, we briefly describe the related work of domain adaptation for sentiment classification.

In real applications, user generated sentiment data can span so many different domains that it is difficult to manually label training data for all of them. A classifier trained on one domain might not perform well on a different domain because it fails to learn the sentiment of unseen words. To address this problem, domain adaptation for sentiment classification has been widely studied in the literature. Blitzer et al. [2007] proposed a structural correspondence learning (SCL) algorithm to train a cross-domain sentiment classifier. SCL is motivated by a multi-task learning algorithm, alternating structural optimization (ASO), proposed by Ando and Zhang [2005]. Given labeled data from a source domain and unlabeled data from both source and target domains, SCL attempts to model the relationship between "pivot features" and "non-pivot features". Pan et al. [2010] proposed a spectral feature alignment (SFA) algorithm to align the domain-specific words from the source and target domains into meaningful clusters, with the help of domain-independent words as a bridge. In the way, the cluster can be used to reduce the gap between domain-specific words of two domains. In SCL and SFA. there is no constrain on the domain-specific words to prevent them from capturing the information from the pivot features. Neither is there a constrain that stops leakage of domain-specific words into the pivot features. In this paper, our PJNMF uses a regularization term on the pivots and domain-specific words, ensuring that the pivots capture only correspondence aspects and the domain-specific words capture only individual aspects. Besides, the formulation and optimization of PJNMF in this paper are different from that in [Blitzer *et al.*, 2007; Pan *et al.*, 2010].

We also note that our PJNMF is related to transfer common knowledge across domains via matrix factorization in [Li *et al.*, 2009]. However, Li et al. [2009] explored the matrix factorization and knowledge transfer by optimizing the two objective functions separately, e.g., maximizing the empirical likelihood on the source and target domains, respectively. Actually, these two objective functions are complementary to each other and optimizing them simultaneously can make the solution smoother and further improve the accuracy on the target domain. Recently, Zhou et al. [2014a] also explored PJNMF for community question retrieval. On the contrary, this paper focuses on domain adaptation for sentiment classification, giving rise to the different task description and model formulation.

Finally, other studies also address domain adaptation for sentiment classification from different perspectives. He et al. [2011] employed a joint sentiment-topic model for cross-domain sentiment classification; Bollegala et al. [2011] used a sentiment sensitive thesaurus to perform cross-domain sentiment classification; Glorot et al. [2011] and Chen et al. [2012] employed a deep learning approach to perform domain adaptation for large-scale sentiment classification; Li et al. [2013] proposed an active learning algorithm for cross-domain sentiment classification. In contrast, our paper conducts domain adaptation for sentiment classification from a new perspective via pivots transfer under a constraint framework.

3 Linking Heterogeneous Input Features with Pivots

In this section, we propose the learning scheme for domain adaptation with pivots, followed by the optimization algorithm and convergence analysis.

3.1 Learning Scheme

Given two domains X_s and X_t, where X_s and X_t are referred to a source domain and a target domain, respectively. Suppose we have a set of labeled sentiment documents as well as some unlabeled documents in a source domain X_s with size n_s, containing terms from a vocabulary $\mathcal{V}$ with size m. The documents in source domain X_s can be represented as a term-document matrix $\mathbf{X}_s = \{\mathbf{x}_1^{(s)}, \cdots, \mathbf{x}_{n_s}^{(s)}\} \in \mathbb{R}^{m \times n_s}$, where each element denotes the weight of the corresponding term, for example tf-idf used in this paper. The labels on sentiment documents can be described using a $n_s \times 2$ matrix $\mathbf{Y}_s$ where $\mathbf{Y}_s(i,1) = 1$ if the i-th document expresses positive sentiment, and $\mathbf{Y}_s(i,2) = 1$ for negative sentiment [Li *et al.*, 2009].

Similarly, suppose we have a set of unlabeled sentiment documents in a target domain X_t with size n_t, containing terms from a vocabulary $\mathcal{V}$ with size m. The documents in target domain X_t can also be represented as a term-document matrix $\mathbf{X}_t = \{\mathbf{x}_1^{(t)}, \cdots, \mathbf{x}_{n_t}^{(t)}\} \in \mathbb{R}^{m \times n_t}$, where each element denotes a tf-idf weight of the corresponding term. The task of cross-domain sentiment classification is to learn a robust classifier to predict the polarity of unseen sentiment documents from X_t. Note that we only consider one source domain and one target domain in this paper. However, our proposed algorithm is a general framework and can be easily adapted to multi-domain problems.

Without loss of generality, let $\bar{\mathbf{U}}_d = [\mathbf{U}_0, \mathbf{U}_d] \in \mathbb{R}^{m \times (k_0 + k_d)}$ be the term-topic matrix corresponding to domain d, where k_0 is the number of pivot topics, k_d is the

number of domain-specific topics corresponding to domain d, and $d \in \{s, t\}$. Term-topic matrix $\mathbf{U}_0$ can be represented as $\mathbf{U}_0 = [\mathbf{u}_1^{(0)}, \cdots, \mathbf{u}_{k_0}^{(0)}] \in \mathbb{R}^{m \times k_0}$, in which each column corresponds to a pivot topic. While the term-topic matrix $\mathbf{U}_d$ can be represented as $\mathbf{U}_d = [\mathbf{u}_1^{(d)}, \cdots, \mathbf{u}_k^{(d)}] \in \mathbb{R}^{m \times k_d}$. The total number of topics in domain d is $k_0 + k_d$. Let $\mathbf{V}_d^T = [\mathbf{v}_1^{(d)}, \cdots, \mathbf{v}_{n_d}^{(d)}] \in \mathbb{R}^{n_d \times (k_0 + k_d)}$ be the document-topic matrix corresponding to domain d, in which each column denotes the document representation in the topic space. We also denote $\mathbf{V}_d^T = [\mathbf{H}_d^T, \mathbf{L}_d^T]$, where $\mathbf{H}_d \in \mathbb{R}^{k_0 \times n_d}$ and $\mathbf{L}_d \in \mathbb{R}^{k_d \times n_d}$ correspond to the coefficients of shared topics $\mathbf{U}_0$ and domain-specific topics $\mathbf{U}_d$, respectively.

Given a sentiment document collection $\mathbf{X}_s$ in a source domain with unlabeled and labeled data, the document representation based on non-negative matrix factorization (NMF) [Lee and Seung, 2001] can be formulated as follows:

$$\mathcal{O}_s = \left\|\mathbf{X}_s - [\mathbf{U}_0, \mathbf{U}_s]\mathbf{V}_s\right\|_F^2 \tag{1}$$

where $\|\cdot\|_F^2$ is the Frobenius norm.

In a target domain, we transfer the pivot topics $\mathbf{U}_0$ from the source domain X_s to the target domain X_t as a bridge. Specially, given a set of unlabeled sentiment document collection $\mathbf{X}_t$ in a target domain, the sentiment classification problem based on non-negative matrix factorization can be formulated as follows:

$$\mathcal{O}_t = \left\|\mathbf{X}_t - [\mathbf{U}_0, \mathbf{U}_t]\mathbf{V}_t\right\|_F^2 \tag{2}$$

Once obtaining the representation of documents from the two domains, a natural avenue is to integrating the two objective functions seamlessly into one unified framework. However, this direct solution has a crucial drawback in segregating the pivot and individual feature spaces: there is no constraint on the individual feature spaces to prevent them from capturing basis vectors from the pivot spaces. Neither is there a constraint that stops leakage of individual basis vectors into the pivot space. Of those two issues, the first is important when we are interested in the individual aspects of the two domains. While the second issue is more important when looking from the point of view of transfer learning. This is because when the pivot matrix $\mathbf{U}_0$ captures some topics or basis vectors corresponding to the individual aspect of one domain and used later for modeling the other domain, this leakage degrades performance.

To circumvent the above problems, we propose to add a regularization on the pivot and domain-specific feature spaces which can avoid both of the above problems. It not only ensures that $\mathbf{U}_s$ and $\mathbf{U}_t$ capture only the individual basis vectors but also that $\mathbf{U}_0$ captures only the common basis vectors. With this regularization, our optimization cost function is to minimize the following form

$$\begin{aligned}\mathcal{O} = \lambda_s\left\|\mathbf{X}_s - [\mathbf{U}_0, \mathbf{U}_s]\mathbf{V}_s\right\|_F^2 + \lambda_t\left\|\mathbf{X}_t - [\mathbf{U}_0, \mathbf{U}_t]\mathbf{V}_t\right\|_F^2 \\ + R(\mathbf{U}_0, \mathbf{U}_s, \mathbf{U}_t) \\ s.t.\ \ \mathbf{U}_0, \mathbf{U}_s, \mathbf{U}_t, \mathbf{V}_s, \mathbf{V}_t \geq 0\end{aligned} \tag{3}$$

where $\lambda_s \triangleq \|\mathbf{X}_s\|_F^{-2}$, $\lambda_t \triangleq \|\mathbf{X}_t\|_F^{-2}$, and $R(\mathbf{U}_0, \mathbf{U}_s, \mathbf{U}_t)$ is a regularization term used to penalize the "similarity" between feature spaces spanned by matrices $\mathbf{U}_0$, $\mathbf{U}_s$ and $\mathbf{U}_t$.

When seeking feature spaces which do not capture the similar basis vectors and be complementary to each other, one way to formulate them is by considering them as mutually orthogonal. Note for example that if the feature spaces spanned by matrices $\mathbf{U}_0$, $\mathbf{U}_s$, are mutually orthogonal, we have $\mathbf{U}_0^T\mathbf{U}_s = \mathbf{0}$. To impose this constraint, we choose to minimize the sum-of-squares of entries of the matrix $\mathbf{U}_0^T\mathbf{U}_s$, i.e., $\|\mathbf{U}_0^T\mathbf{U}_s\|_F^2$ which uniformly optimizes each entry of $\mathbf{U}_0^T\mathbf{U}_s$. With this choice, the regularization term of equation (3) is given by

$$R(\mathbf{U}_0, \mathbf{U}_s, \mathbf{U}_t) = \alpha\|\mathbf{U}_0^T\mathbf{U}_s\|_F^2 + \beta\|\mathbf{U}_0^T\mathbf{U}_t\|_F^2 + \gamma\|\mathbf{U}_s^T\mathbf{U}_t\|_F^2 \tag{4}$$

where α, β and γ are the regularization parameters.

3.2 Optimization Algorithm

Note that the optimization problem in equation (3) is not convex in variables $\{\mathbf{U}_0, \mathbf{U}_s, \mathbf{U}_t, \mathbf{V}_s, \mathbf{V}_t\}$ together. However, when considering one variable at a time, the cost function turns out to be convex. For example, given $\{\mathbf{U}_s, \mathbf{U}_t, \mathbf{V}_s, \mathbf{V}_t\}$, the cost function is a convex function w.r.t. $\mathbf{U}_0$. Therefore, although we cannot expect to get a global minimum of the above problem, we shall develop an algorithm which is not only simple and efficient but its convergence can also be guaranteed.

The equation (3) is a constrained optimization problem due to the nonnegative constraints on the factorization matrices, and can be solved using the Lagrange multiplier method. Let $A^{u_0}(i,j)$ be the Lagrangian multiplier for constraints $\mathbf{U}_0(i,j) \geq 0$ and $\mathbf{A}^{u_0} = [A^{u_0}(i,j)]$, let Tr[$\cdot$] denote the trace of a matrix. Similarly if $\mathbf{A}^{u_s}$, $\mathbf{A}^{u_t}$, $\mathbf{A}^{v_s}$, $\mathbf{A}^{v_t}$ are the Lagrangian multiplier matrices for nonnegative constraints of matrices $\mathbf{U}_s$, $\mathbf{U}_t$, $\mathbf{V}_s$ and $\mathbf{V}_t$, then the above cost function in an unconstrained form (denoted by $\mathcal{L}$) can be written as below

$$\begin{aligned}\mathcal{L} = \lambda_s\|\mathbf{X}_s - [\mathbf{U}_0, \mathbf{U}_s]\mathbf{V}_s\|_F^2 + \lambda_t\|\mathbf{X}_t - [\mathbf{U}_0, \mathbf{U}_t]\mathbf{V}_t\|_F^2 \\ + D(\mathbf{U}_0, \mathbf{U}_s, \mathbf{U}_t, \mathbf{V}_s, \mathbf{V}_t)\end{aligned}$$

where $D(\mathbf{U}_0, \mathbf{U}_s, \mathbf{U}_t, \mathbf{V}_s, \mathbf{V}_t)$ is defined as

$$\begin{aligned}D(\mathbf{U}_0, \mathbf{U}_s, \mathbf{U}_t, \mathbf{V}_s, \mathbf{V}_t) \triangleq \alpha\|\mathbf{U}_0^T\mathbf{U}_s\|_F^2 + \beta\|\mathbf{U}_0^T\mathbf{U}_t\|_F^2 \\ +\gamma\|\mathbf{U}_s^T\mathbf{U}_t\|_F^2 + \mathrm{Tr}(\mathbf{A}^{u_0}\mathbf{U}_0^T) + \mathrm{Tr}(\mathbf{A}^{u_s}\mathbf{U}_s^T) + \mathrm{Tr}(\mathbf{A}^{u_t}\mathbf{U}_t^T) \\ +\mathrm{Tr}(\mathbf{A}^{v_s}\mathbf{V}_s^T) + \mathrm{Tr}(\mathbf{A}^{v_t}\mathbf{V}_t^T)\end{aligned}$$

Optimize $\mathbf{U}_0$ given $\{\mathbf{U}_s, \mathbf{U}_t, \mathbf{V}_s, \mathbf{V}_t\}$

The first derivative of the cost function $\mathcal{L}$ with respect to matrix $\mathbf{U}_0$ is given by

$$\begin{aligned}\nabla_{\mathbf{U}_0}\mathcal{L} = 2\big[\lambda_s(\mathbf{X}_s^{(r)} - \mathbf{X}_s)\mathbf{H}_s^T + \lambda_t(\mathbf{X}_t^{(r)} - \mathbf{X}_t)\mathbf{L}_t^T \\ + (\alpha\mathbf{U}_s\mathbf{U}_s^T + \beta\mathbf{U}_t\mathbf{U}_t^T)\mathbf{U}_0 + \mathbf{A}^{u_0}\big]\end{aligned} \tag{5}$$

where $\mathbf{X}_s^{(r)} \triangleq [\mathbf{U}_0^{(r)}, \mathbf{U}_s^{(r)}]\mathbf{H}_s^{(r)}$ and $\mathbf{X}_t^{(r)} \triangleq [\mathbf{U}_0^{(r)}, \mathbf{U}_t^{(r)}]\mathbf{H}_t^{(r)}$. Using Karush-Kuhn-Tucker (KKT) conditions $\mathbf{A}^{u_0}(i,j)\mathbf{U}_0(i,j) = 0$ with the expression of the gradient $\nabla\mathcal{L}$, for any stationary point, we get the following

$$\begin{aligned}\big[\lambda_s(\mathbf{X}_s^{(r)} - \mathbf{X}_s)\mathbf{H}_s^T + \lambda_t(\mathbf{X}_t^{(r)} - \mathbf{X}_t)\mathbf{H}_t^T + \\ (\alpha\mathbf{U}_s\mathbf{U}_s^T + \beta\mathbf{U}_t\mathbf{U}_t^T)\mathbf{U}_0\big](i,j)\mathbf{U}_0(i,j) = 0\end{aligned} \tag{6}$$

which leads to the following update equation

$$\mathbf{U}_0 \leftarrow \mathbf{U}_0 \frac{[\lambda_s \mathbf{X}_s \mathbf{H}_s^T + \lambda_t \mathbf{X}_t \mathbf{H}_t^T]}{[\lambda_s \mathbf{X}_s^{(r)} \mathbf{H}_s^T + \lambda_t \mathbf{X}_t^{(r)} \mathbf{H}_t + (\alpha \mathbf{U}_s \mathbf{U}_s^T + \beta \mathbf{U}_t \mathbf{U}_t^T)\mathbf{U}_0]} \quad (7)$$

where $\frac{[\cdot]}{[\cdot]}$ is element-wise division.

Optimize $\mathbf{U}_s$ given $\{\mathbf{U}_0, \mathbf{U}_t, \mathbf{V}_s, \mathbf{V}_t\}$

The first derivative of the cost function $\mathcal{L}$ with respect to matrix $\mathbf{U}_s$ is given by

$$\nabla_{\mathbf{U}_s} \mathcal{L} = 2[\lambda_s(\mathbf{X}_s^{(r)} - \mathbf{X}_s)\mathbf{L}_s^T + (\alpha \mathbf{U}_0 \mathbf{U}_0^T + \gamma \mathbf{U}_t \mathbf{U}_t^T)\mathbf{U}_s + \mathbf{A}^{u_s}]$$

Similar as above, using KKT conditions $\mathbf{A}^{u_s}(i,j)\mathbf{U}_s(i,j) = 0$ with the expression of the gradient $\nabla\mathcal{L}$, for any stationary point, we get the following update equation

$$\mathbf{U}_s \leftarrow \mathbf{U}_s \frac{[\lambda_s \mathbf{X}_s \mathbf{L}_s^T]}{\left[\lambda_s \mathbf{X}_s^{(r)} \mathbf{L}_s^T + (\alpha \mathbf{U}_0 \mathbf{U}_0^T + \gamma \mathbf{U}_t \mathbf{U}_t^T)\mathbf{U}_s\right]} \quad (8)$$

Similarly, optimizing for $\mathbf{U}_t$ given $\{\mathbf{U}_0, \mathbf{U}_s, \mathbf{V}_s, \mathbf{V}_t\}$, we get the following update equation

$$\mathbf{U}_t \leftarrow \mathbf{U}_t \frac{[\lambda_t \mathbf{X}_t \mathbf{H}_t^T]}{\left[\lambda_t \mathbf{X}_t^{(r)} \mathbf{H}_t^T + (\beta \mathbf{U}_0 \mathbf{U}_0^T + \gamma \mathbf{U}_s \mathbf{U}_s^T)\mathbf{U}_t\right]} \quad (9)$$

Update equations for matrices $\mathbf{V}_s$, $\mathbf{V}_t$ and $\mathbf{W}_s$, are similar to standard NMF and are given by

$$\mathbf{V}_s \leftarrow \mathbf{V}_s \frac{[\bar{\mathbf{U}}_s^T \mathbf{X}_s]}{\left[\bar{\mathbf{U}}_s^T \bar{\mathbf{U}}_s \mathbf{V}_s\right]}, \quad \mathbf{V}_t \leftarrow \mathbf{V}_t \frac{[\bar{\mathbf{U}}_t^T \mathbf{X}_t]}{\left[\bar{\mathbf{U}}_t^T \bar{\mathbf{U}}_t \mathbf{V}_t\right]} \quad (10)$$

We note that multiplicative update rules given by equations (7)∼(10) are obtained by extending the updates of standard NMF [Lee and Seung, 2001]. There are alternative ways of optimizing the objective function $\mathcal{L}$ such as alternating least squares and the active set method [Kim and Park, 2008] or the projected gradients approach [Lin, 2007], which often have better convergence behavior. Nonetheless, the multiplicative updates derived in this paper have reasonably fast convergence behavior as shown empirically in the experiments. Besides, we also note that a good initialization of matrices $\mathbf{U}_0$, $\mathbf{U}_s$, $\mathbf{U}_t$, $\mathbf{V}_s$ and $\mathbf{V}_t$ may lead to quicker convergence of the proposed algorithm. Several methods have been proposed for NMF to address this issue [Langville *et al.*, 2006; Boutsidis and Gallopoulos, 2008]. However, it is difficult to use them for initializing matrix $\mathbf{U}_0$ and the corresponding coefficient matrix $\mathbf{V}_s$ and $\mathbf{V}_t$. Other matrices such as $\mathbf{U}_s$ and $\mathbf{U}_t$ also depend on $\mathbf{U}_0$ given the data. Therefore, we initialize these matrices by random nonnegative values.

With the optimization results of $\{\mathbf{U}_0, \mathbf{U}_t\}$, a linear classifier is trained on the transformed labeled data of the source domain. Support Vector Machines (SVM) being known to perform well on sentiment classification [Pang *et al.*, 2002], we use a linear SVM with squared hinge loss. This classifier is eventually tested on the target domain.

3.3 Convergence Analysis

Without loss of generality, we only show the convergence of multiplicative update given by equation (8). We first introduce the definition of the auxiliary function as follows.

Definition 1. *$\mathcal{F}(\mathbf{X}, \mathbf{X}')$ is an auxiliary function for $\mathcal{L}(\mathbf{X})$ if $\mathcal{L}(\mathbf{X}) \leq \mathcal{F}(\mathbf{X}, \mathbf{X}')$ and equality holds if and only if $\mathcal{L}(\mathbf{X}) = \mathcal{F}(\mathbf{X}, \mathbf{X})$.*

Lemma 1. *[Lee and Seung, 2001] If $\mathcal{F}$ is an auxiliary function for $\mathcal{L}$, $\mathcal{L}$ is non-increasing under the update*

$$\mathbf{X}^{(r+1)} = \arg\min_{\mathbf{X}} \mathcal{F}(\mathbf{X}, \mathbf{X}^{(r)})$$

Proof. By Definition 1, $\mathcal{L}(\mathbf{X}^{(r+1)}) \leq \mathcal{F}(\mathbf{X}^{(r+1)}, \mathbf{X}^{(r)}) \leq \mathcal{F}(\mathbf{X}^{(r)}, \mathbf{X}^{(r)}) = \mathcal{L}(\mathbf{X}^{(r)})$ □

Lemma 2. *Let $\mathcal{L}(\mathbf{U}_s^{(r)})$ denote the sum of all terms in $\mathcal{L}$ that contain $\mathbf{U}_s^{(r)}$, the following function is an auxiliary function for $\mathcal{L}(\mathbf{U}_s^{(r)})$*

$$\mathcal{F}(\mathbf{U}_s, \mathbf{U}_s^{(r)}) = \mathcal{L}(\mathbf{U}_s^{(r)}) + (\mathbf{U}_s - \mathbf{U}_s^{(r)}) \bigtriangledown \mathcal{L}(\mathbf{U}_s^{(r)}) + \frac{1}{2}(\mathbf{U}_s - \mathbf{U}_s^{(r)})^2 \mathcal{S}(\mathbf{U}_s^{(r)})$$

$$\mathcal{S}(\mathbf{U}_s^{(r)}) = \frac{\left[\lambda_s \mathbf{X}_s^{(r)} \mathbf{L}_s^T + (\alpha \mathbf{U}_0 \mathbf{U}_0^T + \gamma \mathbf{U}_t \mathbf{U}_t^T)\mathbf{U}_s^{(r)}\right]}{[\mathbf{U}_s^{(r)}]}$$

Following the similar convergence analysis as shown above, we can prove the updating rules for the rest variables.

Theorem 1. *The objective function $\mathcal{L}(\mathbf{U}_0, \mathbf{U}_s, \mathbf{U}_t, \mathbf{V}_s, \mathbf{V}_t)$ is non-increasing under the alternating multiplicative update rules of equations (7)∼(10).*

Proof. Since we are minimizing $\mathcal{L}(\mathbf{U}_s)$ using the auxiliary function $\mathcal{F}(\mathbf{U}_s, \mathbf{U}_s^{(r)})$. Therefore, evaluating $\nabla\mathcal{F}(\mathbf{U}_s, \mathbf{U}_s^{(r)}) = 0$ and utilizing the results of Lemma 1 and Lemma 2, we get the following update equation

$$\mathbf{U}_s^{(r+1)} = \mathbf{U}_s^{(r)} - [\nabla\mathcal{L}(\mathbf{U}_s^{(r)})/\mathcal{L}(\mathbf{U}_s^{(r)})] \quad (11)$$

Noting that $\nabla\mathcal{L}(\mathbf{U}_s^{(r)})$

$$= [\lambda_s(\mathbf{X}_s^{(r)} - \mathbf{X}_s)\mathbf{H}_s^T + \alpha \mathbf{U}_0 \mathbf{U}_0^T \mathbf{U}_s^{(r)} + \gamma \mathbf{U}_t \mathbf{U}_t^T \mathbf{U}_s^{(r)}] \quad (12)$$

and substituting $\mathcal{S}(\mathbf{U}_s^{(r)})$ from Lemma 2, we get the desired update of equation (8). □

3.4 Algorithm Complexity

In this section, we analyze the computational complexity of the learning algorithm described in equations (7)∼(10). Besides expressing the complexity of the algorithm using big O notation, we also count the number of arithmetic operations to provide more details about the run time. Computational complexity of learning matrix $\mathbf{U}_0$ is $O(m \times n \times k_0)$ per iteration where $n = \max(n_s, n_t)$. Similarly, for each iteration, learning matrices $\mathbf{U}_s$ and $\mathbf{U}_t$ takes $O(m \times n_s \times k_s)$ and $O(m \times n_t \times k_t)$, respectively. Learning matrices $\mathbf{V}_s$ and $\mathbf{V}_t$ takes $O(m \times n_s \times (k_0 + k_s))$ and $O(m \times n_t \times (k_0 + k_t))$ operations per iteration. Therefore, overall complexity of the algorithm, dominated by computation of matrices $\mathbf{V}_s$ and $\mathbf{V}_t$, is $O(m \times n \times k)$ where $k = \max(k_0 + k_s, k_0 + k_t)$. This is also the order of complexity of NMF algorithm [Lee and

Table 1: Average results (accuracy±standard deviation) for cross-domain sentiment classification on the Amazon product benchmark of 4 domains. The bold formate indicates the best results, † indicates that the difference between the results of our proposed approach PJNMF and SDA is significant with $p < 0.05$ under a McNemar paired test for labeling disagreements, and ‡ indicates the mildly significant with $p < 0.08$.

Task	baseline	SCL	MCT	SFA	SDA	CODA	PJNMF
BD	76.41 ± 0.31	78.68 ± 0.26	78.92 ± 0.23	80.58 ± 0.18	81.12 ± 0.17	80.64 ± 0.16	$^{\ddagger}\mathbf{81.85} \pm 0.17$
ED	71.95 ± 0.19	75.51 ± 0.27	72.67 ± 0.35	76.02 ± 0.12	76.63 ± 0.25	76.10 ± 0.23	$^{\ddagger}\mathbf{77.35} \pm 0.20$
KD	73.35 ± 0.20	76.88 ± 0.29	74.05 ± 0.28	76.55 ± 0.16	76.85 ± 0.28	76.62 ± 0.21	$^{\dagger}\mathbf{78.62} \pm 0.28$
DB	73.86 ± 0.24	78.27 ± 0.18	75.67 ± 0.30	77.58 ± 0.23	78.22 ± 0.33	77.83 ± 0.17	$^{\dagger}\mathbf{79.27} \pm 0.25$
EB	72.14 ± 0.26	75.06 ± 0.21	72.90 ± 0.27	75.38 ± 0.27	75.50 ± 0.19	75.46 ± 0.25	$^{\ddagger}\mathbf{76.30} \pm 0.22$
KB	71.25 ± 0.18	73.08 ± 0.24	74.01 ± 0.31	74.15 ± 0.34	74.47 ± 0.25	75.41 ± 0.22	$^{\dagger}\mathbf{75.87} \pm 0.23$
BE	71.75 ± 0.32	75.21 ± 0.18	75.62 ± 0.26	75.35 ± 0.26	75.77 ± 0.27	$\mathbf{76.34} \pm 0.18$	$^{\ddagger}76.28 \pm 0.27$
DE	72.38 ± 0.20	75.95 ± 0.25	76.82 ± 0.34	77.13 ± 0.23	77.65 ± 0.22	$\mathbf{77.94} \pm 0.20$	$^{\ddagger}77.86 \pm 0.24$
KE	83.35 ± 0.13	85.18 ± 0.15	84.24 ± 0.25	85.01 ± 0.23	84.65 ± 0.34	84.50 ± 0.32	$^{\dagger}\mathbf{85.92} \pm 0.32$
BK	74.44 ± 0.30	77.06 ± 0.21	78.31 ± 0.22	78.28 ± 0.25	78.54 ± 0.23	78.35 ± 0.26	$^{\ddagger}\mathbf{79.15} \pm 0.29$
DK	75.11 ± 0.33	78.96 ± 0.19	80.57 ± 0.24	80.35 ± 0.29	80.77 ± 0.31	80.65 ± 0.24	$^{\ddagger}\mathbf{81.26} \pm 0.33$
EK	85.11 ± 0.13	85.08 ± 0.16	85.33 ± 0.26	85.91 ± 0.19	$\mathbf{87.25} \pm 0.20$	86.08 ± 0.27	86.37 ± 0.21
avg.	75.09 ± 0.23	77.91 ± 0.20	77.43 ± 0.28	78.52 ± 0.23	78.95 ± 0.25	78.83 ± 0.23	$^{\ddagger}\mathbf{79.68} \pm 0.25$

Seung, 2001] for each iteration. Note that, for implementation efficiency, when computing matrices such as $\bar{\mathbf{U}}_s\bar{\mathbf{U}}_s^T\mathbf{V}_s$, we should compute $\bar{\mathbf{U}}_s\bar{\mathbf{U}}_s^T$ first and then multiply it to matrix $\mathbf{V}_s$. Similarly, when computing $\mathbf{U}_s\mathbf{U}_s^T\mathbf{U}_0$, we should compute $\mathbf{U}_s^T\mathbf{U}_0$ first and pre-multiply by $\mathbf{U}_s$. The main idea is to group the matrix multiplications by their inner-products first rather than their outer products.

4 Experiments

4.1 Data Set

Domain adaptation for sentiment classification has been widely studied in the NLP community. A large majority experiments are performed on the benchmark made of reviews of Amazon products gathered by Blitzer et al. [2007]. This data set contains 4 different domains: Book (B), DVDs (D), Electronics (E) and Kitchen (K). For simplicity and comparability, we follow the convention of [Blitzer *et al.*, 2007; Pan *et al.*, 2010; Glorot *et al.*, 2011] and only consider the binary classification problem whether a review is positive (higher than 3 stars) or negative (3 stars or lower). There are 1000 positive and 1000 negative reviews for each domain, as well as approximately 4,000 unlabeled reviews (varying slightly between domains). The positive and negative reviews are also exactly balanced. Following the literature [Pan *et al.*, 2010], we can construct 12 cross-domain sentiment classification tasks: DB, EB, KB, KE, DE, BE, BD, KD, ED, BK, DK, EK, where the word before corresponds with the source domain and the word after corresponds with the target domain. To be fair to other algorithms that we compare to, we use the raw bag-of-words unigram/bigram features as their input and pre-process with tf-idf. Table 2 presents the statistics of the data set.

Table 2: Amazon review statistics. This table depicts the number of training, testing and unlabeled reviews for each domain, as well as the portion of negative training reviews of the data set.

Domain	#Train	#Test	#Unlab.	% Neg.
Books	1600	400	4465	50%
DVDs	1600	400	5945	50%
Electronics	1600	400	5681	50%
Kitchen	1600	400	3586	50%

4.2 Compared Methods

As *baseline*, we train a linear SVM on the raw bag-of-words representation of the labeled *source* and test it on *target*. In the original paper regarding the benchmark data set, Blitzer et al. [2007] adapt Structural Correspondence Learning (SCL) for sentiment analysis. Li and Zong [2008] propose the Multi-label Consensus Training (MCT) approach which combines several base classifiers trained with SCL. Pan et al. [2010] first use a Spectral Feature Alignment (SFA) algorithm to align words from different source and target domains to help bridge the gap between them. Glorot et al. [2011] first employ stacked Denoising Auto-encoders (SDA) to extract meaningful representation for domain adaptation. Chen et al. [2011] propose a state-of-the-art domain adaptation algorithm called CODA, which is based on sample and feature selection, applied to tf-idf features. For SCL, SDA and CODA, we use implementations provided by the authors. For SFA and MCT, we re-implement them based on the original papers. The above methods serve as comparisons in our empirical evaluation. All hyper-parameters are set by 5-fold cross validation on the source training set[1].

Table 1 shows the accuracy of classification results for all methods and for all source-target domain pairs. We can check that all compared methods achieve the similar performance with the results reported in the original papers. From Table 1, we can see that our proposed approach outperforms all

[1]We keep the default value of some of the parameters in SCL and SFA, e.g., the number of stop-words removed and stemming parameters − as they were already tuned for this benchmark set by the authors.

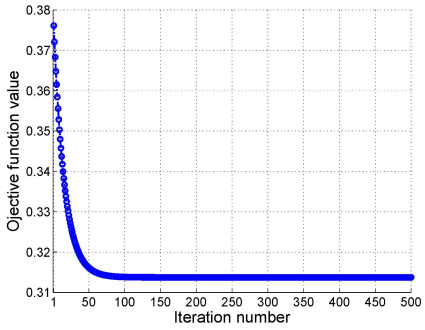

Figure 1: Convergence curve of PJNMF.

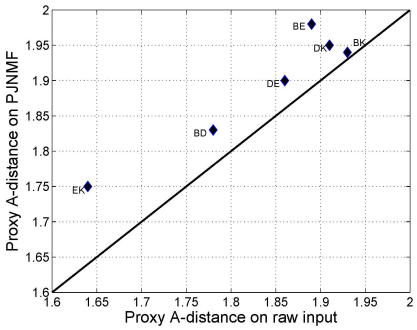

Figure 2: Proxy A-distance between domains of the Amazon benchmark for the 6 different paris.

other six comparison methods in general. The `baseline` performs poorly on all the 12 tasks, while the other five domain adaptation methods, SCL, MCT, SFA, SDA and CODA, consistently outperform the `baseline` method across all the 12 tasks, which demonstrates that the transferred knowledge from the source domain to the target domain is useful for sentiment classification. Nevertheless, the improvements achieved by these five methods over the `baseline` are much smaller than the proposed approach. Surprisingly, we note that the deep learning based method (SDA) does not achieve the satisfactory results with ours, the reason may be that unsupervised feature learning typically only model the syntactic context of words but ignore the sentiment of text. This is problematic for sentiment analysis as they usually map words with similar context but opposite sentiment polarity, such as *good* and *bad*, to neighboring word vectors [Tang *et al.*, 2014]. We also conduct significance tests for our proposed approach and SDA using a McNemar paired test for labeling disagreements [Gillick and Cox, 1989]. In general, the average result on the 12 source-target domain pairs indicates that the difference between our proposed approach PJNMF and SDA is mildly significant with $p < 0.08$.

4.3 Convergence Behavior

In subsection 3.3, we have shown that the multiplicative updates given by equations (7)$\sim$(10) are convergent. Here, we empirically show the convergence behavior of PJNMF. Figure 1 shows the convergence curve of PJNMF on the training data set. From the figure, y-axis is the value of objective function and x-axis denotes the iteration number. We can see that the multiplicative updates for PJNMF converge very fast, usually within 200 iterations.

4.4 Further Analysis

In this subsection, we would like to see how similarity the two domains are from each other. Ben-David et al. [2007] showed the A-distance as a measure of how different between the two domains. They hypothesized that it should be difficult to discriminate between the source and target domains in order to have a good transfer between them. In practice, computing the exact A-distance is impossible and one has to compute a proxy. Similar to [Glorot *et al.*, 2011], the proxy for the A-distance is then defined as $2(1 - 2\epsilon)$, where ϵ is the generalization error of a linear SVM classifier trained on the binary classification problem to distinguish inputs between the two domains. Figure 2 presents the results for each pair of domains. Surprisingly, the distance is increased with the help of pivot features. We explain this effect through the fact that PJNMF can ensure that the pivots capture only correspondence aspects and the domain-specific words capture only individual aspects.

5 Conclusion

This paper proposes a joint non-negative matrix factorization framework by linking heterogeneous input features with pivots for domain adaptation, where different domains have heterogeneous input spaces. This is achieved by learning the domain-specific information from different domains into unified topics, with the help of pivots across all domains. An objective function is defined to learn the mappings from the heterogeneous input features to a common space, and then construcct a single prediction model in this space. To demonstrate the effectiveness of the proposed approach, we conduct experiments on a benchmark composed of reviews of 4 types of Amazon products. Experimental results show that our proposed approach significantly outperforms the baseline method, and achieves an accuracy which is competitive with the state-of-the-art methods for sentiment classification adaptation. There are some ways in which this research could be continued. First, since each word has its sentiment polarity (e.g., positive, negative), a natural avenue for future research is to encode the domain-independent lexical prior knowledge into our framework. Second, we will try to investigate the scalability of the proposed approach on a larger industrial-strength data set of 22 domains [Glorot *et al.*, 2011].

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 61303180 and No. 61272332), the Beijing Natural Science Foundation (No. 4144087), and the Major Project of National Social Science Found (No. 12&2D223), and the Fundamental Research Funds for the Central Universities (No. CCNU15ZD003). We thank the anonymous reviewers for their insightful comments.

References

[Ando and Zhang, 2005] Rie Kubota Ando and Tong Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *J. Mach. Learn. Res.*, 6:1817–1853, December 2005.

[Ben-David *et al.*, 2007] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *NIPS*, Cambridge, MA, 2007. MIT Press.

[Blitzer *et al.*, 2007] John Blitzer, Ryan McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In *Proceedings of the EMNLP*, EMNLP '07, pages 120–128, 2007.

[Bollegala *et al.*, 2011] Danushka Bollegala, David Weir, and John Carroll. Using multiple sources to construct a sentiment sensitive thesaurus for cross-domain sentiment classification. In *Proceedings of the ACL*, HLT '11, pages 132–141, 2011.

[Boutsidis and Gallopoulos, 2008] C. Boutsidis and E. Gallopoulos. Svd based initialization: a head start for nonnegative matrix factorization. *Pattern Recognition*, 41(4):1350–1362, 2008.

[Chen *et al.*, 2011] Minmin Chen, Kilian Weinberger, and Yixin Chen. Automatic feature decomposition for single view co-training. In *Proceedings of the ICML*, ICML '11, pages 953–960, 2011.

[Chen *et al.*, 2012] Minmin Chen, Zhixiang Xu, Kilian Weinberger, and Fei Sha. Marginalized denoising autoencoders for domain adaptation. In *Proceedings of the ICML*, ICML '12, pages 767–774. July 2012.

[Gillick and Cox, 1989] L. Gillick and S. Cox. Some statistical issues in the comparison of speech recoginition algorithms. In *ICASSP*, 1989.

[Glorot *et al.*, 2011] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the ICML*, 2011.

[He *et al.*, 2011] Yulan He, Chenghua Lin, and Harith Alani. Automatically extracting polarity-bearing topics for cross-domain sentiment classification. In *Proceedings of the ACL*, HLT '11, pages 123–131, 2011.

[Kim and Park, 2008] H. Kim and H. Park. Non-negative matrix factorization based on alternating non-negativity constrained least squares and active set method. *SIAM J Matrix Anal Appl*, 30(2):713–730, 2008.

[Langville *et al.*, 2006] A. Langville, C. Meyer, R. Albright, J. Cox, and D. Duling. Initializations for the nonnegative matrix factorization. In *KDD*, 2006.

[Lee and Seung, 2001] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562. MIT Press, 2001.

[Li and Zong, 2008] Shoushan Li and Chengqing Zong. Multi-domain adaption for sentiment classification: Using multiple classifier combining classification. In *Proceedings of the NLPKE*, NLPKE '08. 2008.

[Li *et al.*, 2009] Tao Li, Vikas Sindhwani, Chris H. Q. Ding, and Yi Zhang 0005. Knowledge transformation for cross-domain sentiment classification. In *SIGIR*, pages 716–717. ACM, 2009.

[Li *et al.*, 2013] Shoushan Li, Yunxia Xue, Zhongqing Wang, and Guodong Zhou. Active learning for cross-domain sentiment classification. In *Proceedings of the IJCAI*, IJCAI'13, pages 2127–2133, 2013.

[Lin, 2007] C. Lin. Projected gradient methods for nonnegative matrix factorization. *Neural Comput*, 19(10), 2007.

[Liu, 2012] B. Liu. Sentiment analysis and opinion mining. *Morgan & Claypool Publishers*, 2012.

[Pan *et al.*, 2010] Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the WWW*, WWW '10, pages 751–760, 2010.

[Pang and Lee, 2008] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135, January 2008.

[Pang *et al.*, 2002] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In *EMNLP*, pages 79–86, 2002.

[Snyder and Barzilay, 2007] Benjamin Snyder and Regina Barzilay. Multiple aspect ranking using the good grief algorithm. In *Proceedings of the NAACL*, pages 300–307, April 2007.

[Tang *et al.*, 2014] Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the ACL*, pages 1555–1565, June 2014.

[Thomas *et al.*, 2006] Matt Thomas, Bo Pang, and Lillian Lee. Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In *Proceedings of the EMNLP*, EMNLP '06, pages 327–335, 2006.

[Zhou *et al.*, 2014a] Guangyou Zhou, Yubo Chen, Daojian Zeng, and Jun Zhao. Group non-negative matrix factorization with natural categories for question retrieval in community question answer archives. In *Proceedings of COLING 2014*, pages 89–98, Dublin, Ireland, 2014.

[Zhou *et al.*, 2014b] Guangyou Zhou, Jun Zhao, and Daojian Zeng. Sentiment classification with graph co-regularization. In *Proceedings of COLING 2014*, pages 1331–1340, Dublin, Ireland, 2014.

[Zhou *et al.*, 2015] Guangyou Zhou, Yin Zhou, Xiyue Guo, Xinhui Tu, and Tingting He. Cross-domain sentiment classification via topical correspondence transfer. *Neurocomputing*, 159:298–305, 2015.