User-guided Cross-domain Sentiment Classification

Arun Reddy Nelakurthi* Hanghang Tong* Ross Maciejewski* Nadya Bliss* Jingrui He*

Abstract
Sentiment analysis has been studied for decades, and it is widely used in many real applications such as media monitoring. In sentiment analysis, when addressing the problem of limited labeled data from the target domain, transfer learning, or domain adaptation, has been successfully applied, which borrows information from a relevant source domain with abundant labeled data to improve the prediction performance in the target domain. The key to transfer learning is how to model the relatedness among different domains. For sentiment analysis, a common practice is to assume similar sentiment polarity for the common keywords shared by different domains. However, existing methods largely overlooked the human factor, i.e., the users who expressed such sentiment. In this paper, we address this problem by explicitly modeling the human factor related to sentiment classification. In particular, we assume that the content generated by the same user across different domains is biased in the same way in terms of the sentiment polarity. In other words, optimistic/pessimistic users demonstrate consistent sentiment patterns, no matter what the context is. To this end, we propose a new graph-based approach named *U-Cross*, which models the relatedness of different domains via both the shared users and keywords. It is non-parametric and semi-supervised in nature. Furthermore, we also study the problem of shared user selection to prevent 'negative transfer'. In the experiments, we demonstrate the effectiveness of *U-Cross* by comparing it with existing state-of-the-art techniques on three real data sets.

Keywords: classification, transfer learning, user modeling

1 Introduction

Sentiment analysis, or opinion mining, is extremely useful in many real applications such as media monitoring, which allows us to gain an overview of public opinion on stocks, products, movies, politicians, or any other topic that is being discussed. For example, the Obama administration used sentiment analysis to gauge public response to campaign messages during the 2012 presidential election; nonprofit organizations, such as the American Cancer Society, have employed sentiment analysis to gauge feedback on their fundraising programs; and Expedia Canada was able to quickly identify and react to the fact that one of their television advertisements was considered to be annoying [1]. In sentiment analysis, when the target domain (e.g., review articles written in Chinese) has only limited amount of labeled data, and it is both costly and tedious to collect more labeled information, a common practice is to apply transfer learning, or domain adaptation, which borrows information from a relevant source domain with abundant labeled data (e.g., review articles written in English) to help improve the prediction performance in the target domain [19].

However, most existing transfer learning techniques for sentiment analysis largely overlooked an important factor, the human factor, which is usually associated with the degree of sentiment or opinion making [2, 12, 4]. In other words, users who are optimistic and positive tend to give high ratings, and vice versa. This bias can also be due to users associated with a company or brand usually post positive reviews for their products and negative reviews for their competitors. Therefore, the human behavior should be explicitly modeled in transfer learning to effectively leverage such information.

In this paper, we propose a new graph based transfer learning approach: User-guided Cross-domain sentiment classification (*U-Cross*). It constructs a user-example-feature tripartite graph, and imposes a set of constraints such that: (1) the sentiment of content generated by the same user is consistent; (2) label information is propagated from the source domain to the target domain via the common keywords; and (3) the subtle language differences between domains are identified by exploiting the label information (abundant from the source domain, and limited from the target domain). This approach is non-parametric and semi-supervised in nature. Furthermore, we address the problem of 'negative transfer' by excluding a set of common users across different domains with known inconsistent behaviors. To demonstrate the effectiveness of the proposed *U-Cross* approach, we test it on three different datasets of

*Arizona State University; Email:{anelakur, hanghang.tong, rmacieje, nadya.bliss, jingrui.he}@asu.edu

[1] http://www.marketingmag.ca/brands/expedia-ca-responds-to-angry-social-media-feedback-with-new-ads-99039

varied sizes, and compare it with state-of-the-art techniques on cross-domain sentiment classification. The major contributions of the paper are as follows.

1. A novel graph based framework for cross-domain sentiment classification, leveraging user-example-feature relationships.
2. A novel user selection approach to avoid negative transfer through soft-score reweighting, and to gauge the consistency of users across the source and target domains.
3. Extensive experimental analysis to demonstrate the effectiveness of *U-Cross* over state-of-the-art cross-domain sentiment classification approaches.

The rest of the paper is organized as follows. In Section 2, we briefly review the related work on transfer learning and cross-domain sentiment analysis. In Section 3, we introduce our proposed graph-based approach *U-Cross*, algorithm and proof of its convergence. A special case of the proposed approach is discussed in Section 4, which is equivalent to an existing method TRITER [5]. Then we demonstrate the effectiveness of *U-Cross* in Section 5 on multiple real datasets. Finally, we conclude the paper in Section 6.

2 Related Work

In this section, we briefly review the related work on transfer learning and cross-domain sentiment classification. Transfer learning has gained a lot of attention in the research community in the last decade [14]. Different supervised, unsupervised and semi-supervised methods have been proposed for a wide variety of applications such as image classification [21], WiFi-localization on time variant data [13], and web document classification [5, 12]. Recently transfer learning is successfully used to classify images by learning the classifier on related text data [16, 20]. Transfer learning is broadly classified into inductive, transductive and unsupervised transfer learning [14]. In this paper, we study the problem of transductive transfer learning also called as domain adaptation. In transductive transfer learning, the data distributions vary across the source and target domains, but the learning task, sentiment analysis in our case is same in both the domains.

In sentiment analysis, given a piece of written text, the task of sentiment classification involves categorizing the text into a specific sentiment polarity – positive or negative (or neutral) [9, 11]. Most of the previous research has shown that sentiment classification is sensitive to the domain from which training data is extracted. A classifier trained on Amazon reviews, can often perform poorly on test data from IMDB movie reviews. The reason being the words or the language constructs can significantly vary from one domain to the other. For e.g. the word "*faster*" in "*faster CPU*","*faster graphics*" in computer domain is positive; in the case of "*faster battery drain*" in case of electronics domain can be negative. So there is a need for transfer learning to model the knowledge in source domain to effectively use it in the target domain.

Sentiment classification in a cross-domain set up is a well studied problem. For example, structural correspondence learning (SCL) generates a set of pivots using common features in both the source and target domains using mutual information and a set of classifiers on the common features [2]; spectral feature alignment (SFA) splits the feature space into domain independent features and domain specific features, then aligns the domain specific features into unified clusters by using domain independent features as a bridge through spectral feature clustering [12]; Transfer component analysis (TCA) utilizes both the shared and the mapped domain-specific topics to span a new shared feature space for knowledge transfer [8]; labeled-unlabeled-feature tripartite graph-based approach called TRITER was proposed to transfer sentiment knowledge from labeled examples in both the source and target domains to unlabeled examples in the target domain [5].

Prior research has shown that user information combined with linguistic features improved sentiment classification. Li et al. [7] proposed a user-item based topic model which can simultaneously utilize the textual topic and latent user-item factors for sentiment analysis; Tang et al. [18] incorporated user- and product-level information using vector space models into a neural network approach for document level sentiment classification. Motivated by prior work which demonstrated the usefulness of user information in single-domain sentiment classification, we propose *U-Cross* to explicitly model the user behaviors by borrowing information from the source domain to help construct the prediction model in the target domain. Tan et al. [17] used a factor-graph model for user labels in a transductive learning setting for a short-text sentiment classification task. It is likely that the user behavior can vary across the source and target domains, if not handled well it can lead to negative transfer of knowledge. Our work varies from Tan et al. [17] as we carefully model the user behavior based on the relatedness between the source and target domains, which prevents the 'negative transfer'.

3 User-Guided Transfer Learning

In this section, we propose a novel graph-based transfer learning approach, which takes into consideration the human factor by modeling the task relatedness via both the shared users and keywords from both the domains.

3.1 Notation Let $\mathcal{X}^S$ denote the set of examples from the source domain, i.e $\mathcal{X}^S = \{x_1^S, ..., x_m^S\} \subset \mathbb{R}^{ds}$, where m is the number of examples from the source domain, and ds the dimensionality of the feature space. Let $\mathcal{Y}^S$ denote the labels of these examples, i.e $\mathcal{Y}^S = \{y_1^S, ..., y_m^S\} \subset \{-1, 1\}^m$, where y_i^S is the class label of x_i^S, $1 \le i \le m$. Similarly for the target domain $\mathcal{X}^T$ denote the set of examples from the target domain, i.e $\mathcal{X}^T = \{x_1^T, ..., x_n^T\} \subset \mathbb{R}^{dt}$, where n is the number of examples from the target domain, and dt the dimensionality of the feature space. Let $\mathcal{Y}^T$ denote the labels of target domain examples, i.e $\mathcal{Y}^T = \{y_1^T, ..., y_{\epsilon n}^T\} \subset \{-1, 1\}^{\epsilon n}$, where y_i^T is the class label of x_i^T, $1 \le i \le \epsilon n$. Let $d = ds \cup dt$ be the combined feature space for the source and target domains. For convenience we represent the features in the shared feature space of size d. Let $\mathcal{U}$ denote the set of users who posted the content of examples both in the source and target domains, i.e $\mathcal{U} = \{u_1, ..., u_u\} \subset [0, 1]^u$, where u is the number of unique users from the source and target domain. Among the target domain examples only the first ϵn are labeled, and $\epsilon = 0$ corresponds to no labels from the target domain. Let $e = m+n$ the total number of examples in source and target domain combined. Further the examples are split into labeled examples $e = m + \epsilon n$ and unlabeled examples $eu = (1 - \epsilon)n$. Our goal is to find a sentiment classification function $f_{eu} \rightarrow \{y_{\epsilon n+1}^T, ..., y_n^T\}$ for all the unlabeled examples in the target domain $\mathcal{X}^T$ with a small error rate.

3.2 User-Example-Feature Tripartite Graph The tripartite graph consists of three different types of nodes: users, examples and keyword features extracted from examples of both the domains. Let $G^{(3)} = \{V^{(3)}, E^{(3)}\}$ denote the undirected tripartite graph, where $V^{(3)}$ is the set of nodes in the graph, and $E^{(3)}$ is the set of weighted edges. Users are connected to examples in the source and target domain, i.e. there exists an edge between every example and the user who posted the example. Moreover, it is also possible to have a set of users who have examples only in source domain or target domain but not in both. All the labeled and unlabeled example nodes are connected to corresponding feature nodes, i.e. there exists an edge between every labeled or unlabeled node to a feature node only if the feature has a positive weight associated with that example. The labeled and the unlabeled example nodes are not connected to each other. The edges between user nodes and examples have a weight $v_j \subset [0, 1]$. In the case of example and feature nodes, the edge weights can either be a real valued or binary values. To explain this with regards to the sentiment classification task and real data, examples correspond to Amazon reviews, features represent the n-gram keywords of each review and user is the one who wrote the review. Figure 1 shows the example of user-example-feature tripartite graph.

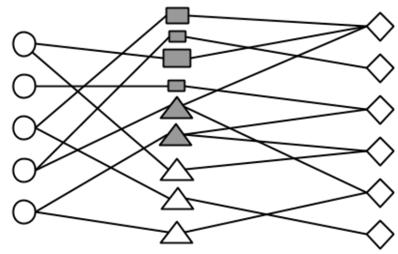

Figure 1: User-Example-Feature Tripartite Graph. Circles represent users; Squares are the source domain examples; Triangles are the target domain examples; Diamonds represent the keyword features. Different sizes of squares represent the reweighted source domain examples. Filled squares and triangles represent the labeled examples.

Given the tripartite graph $G^{(3)}$ we define the symmetric affinity matrix $\mathbf{A}^{(3)}$ of size $(u + e + d)$. The first u nodes correspond to the users, the next e nodes correspond to examples and the last d nodes represent keyword features extracted from examples. Considering m examples from the source domain and n examples from the target domain, the e examples consists of $el = m+\epsilon n$ labeled examples followed by $eu = n - \epsilon n$ unlabeled examples. The affinity matrix has the following structure:

$$\mathbf{A}^{(3)} = \begin{bmatrix} \mathbf{0}_{u\times u} & \mathbf{A}_{u\times e}^{(1,2)} & \mathbf{0}_{u\times d} \\ \mathbf{A}_{e\times u}^{(2,1)} & \mathbf{0}_{e\times e} & \mathbf{A}_{e\times d}^{(2,3)} \\ \mathbf{0}_{d\times u} & \mathbf{A}_{d\times e}^{(3,2)} & \mathbf{0}_{d\times d} \end{bmatrix}$$

where $\mathbf{0}_{a\times b}$ is an $a \times b$ zero matrix, $\mathbf{A}_{u\times e}$ is a non-zero user-example affinity matrix, $\mathbf{A}_{e\times d}$ is a non-zero example-keyword affinity matrix. $\mathbf{A}_{u\times e}$ and $\mathbf{A}_{e\times d}$ are the submatrices of the affinity matrix $\mathbf{A}^{(3)}$. The matrix $\mathbf{A}^{(3)}$ is symmetric matrix such that $\mathbf{A}_{i,j} = \mathbf{A}_{j,i}$ where $\mathbf{A}_{i,j}$ is a submatrix of $\mathbf{A}^{(3)}$. We also define a diagonal matrix $\mathbf{D}^{(3)}$ of size $(u + e + d)$ with a diagonal element $\mathbf{D}_i^{(3)} = \sum_{j=1}^{u+e+d} \mathbf{A}_{i,j}^{(3)}, i = 1, ..., u+e+d$, where $\mathbf{A}_{i,j}^{(3)}$ denote the element in the i^{th} row and j^{th} column of $\mathbf{A}^{(3)}$. The diagonal matrix has the following structure:

$$\mathbf{D}^{(3)} = \begin{bmatrix} \mathbf{D}_{u\times u}^{(1,1)} & \mathbf{0}_{u\times e} & \mathbf{0}_{u\times d} \\ \mathbf{0}_{e\times u} & \mathbf{D}_{e\times e}^{(2,2)} & \mathbf{0}_{e\times d} \\ \mathbf{0}_{d\times u} & \mathbf{0}_{d\times e} & \mathbf{D}_{d\times d}^{(3,3)} \end{bmatrix}$$

where $\mathbf{D}^{(1,1)}$, $\mathbf{D}^{(2,2)}$ and $\mathbf{D}^{(3,3)}$ are submatrices of diagonal matrix $\mathbf{D}^{(3)}$ which equals row sums of affinity submatrices $\mathbf{A}^{(1,2)}$, $(\mathbf{A}^{(1,2)})^T + \mathbf{A}^{(2,3)}$ and $(\mathbf{A}^{(2,3)})^T$ respectively. Finally we define $\mathbf{S}^{(3)} = (\mathbf{D}^{(3)})^{-1/2}\mathbf{A}^{(3)}(\mathbf{D}^{(3)})^{-1/2}$. Similar to $\mathbf{A}^{(3)}$, $\mathbf{S}^{(3)}$ is a symmetric matrix with non-negative elements $\mathbf{S}_{i,j}^{(3)}$ such that the sub matrices $\mathbf{S}^{(1,2)} = (\mathbf{S}^{(2,1)})^T$ and $\mathbf{S}^{(2,3)} = (\mathbf{S}^{(3,2)})^T$.

3.3 Objective Function The goal of building a tripartite graph is to learn the sentiment classification function on unlabeled target domain data. We define four functions f_u, f_{el}, f_{eu} and f_d that take values on users, labeled examples from the source and target domains, unlabeled examples from the target domain and feature nodes respectively, and define f as: $f = [(f_u)^T, (f_{el})^T, (f_{eu})^T, (f_d)^T]^T$. We also define four column vectors y_u, y_{el}, y_{eu} and y_f of size u, el, eu and d respectively. We merge all the column vectors into a single column vector $y = [(y_u)^T, (y_{el})^T, (y_{eu})^T, (y_d)^T]^T$.

Example function f_{el} and column vector y_{el} are comprised of the first $m + \epsilon n$ values for labeled examples, similarly function f_{eu} and column vector y_{eu} are comprised of $n - \epsilon n$ values for unlabeled examples. Vectors y_u, y_{el}, y_{eu} and y_d represent the prior knowledge of users, labeled examples, unlabeled examples and features respectively. If we do not have any prior knowledge we set the vectors to zero. The vector y_{el} is set to sentiment labels $\{-1, 1\}$ corresponding to the labeled examples.

In regular supervised learning problems the training data and test data are usually from the same distribution, but in a situation when training data and test data are from different distributions, it is called covariate shift. In transfer learning tasks, distribution of data in the source domain varies with distribution of data in the target domain. In such scenarios reweighting training data $w(x) = p_{test}(x)/p_{train}(x)$ to fit test data distribution often resulted in increased classification performance,[1, 6, 15]. We used the reweighting technique as suggested in Sugiyama et al. [15] to reweight the source domain examples based on entire set of examples from the target domain.

We propose to minimize the following objective function with respect to f.

$$\begin{aligned} Q_1(f) &= \frac{1}{2}\sum_i^e w_i \sum_j^u v_j \mathbf{A}^{(3)}_{i,j}\Big(\frac{f_i}{\sqrt{\mathbf{D}^{(3)}_i}} - \frac{f_j}{\sqrt{\mathbf{D}^{(3)}_j}}\Big)^2 \\ &+ \frac{1}{2}\sum_i^e w_i \sum_j^d \mathbf{A}^{(3)}_{i,j}\Big(\frac{f_i}{\sqrt{\mathbf{D}^{(3)}_i}} - \frac{f_j}{\sqrt{\mathbf{D}^{(3)}_j}}\Big)^2 \\ &+ \mu \sum_k^{u+e+d} (f_k - y_k)^2 \\ &= f^T(I_{(u+e+d)\times(u+e+d)} - \mathbf{S}^{(3)})f + \mu||f - y||^2 \end{aligned}$$

where w_i is the example reweighting parameter to reduce the covariate shift between the source and target domain examples, $w_i = 1$ for the target domain examples, v_j is the user soft-score weight to ensure user consistency across the source and target domains, μ is a small positive parameter and I is the identity matrix. The objective function has three terms. The first and second terms in the equation measures the label smoothness of the function f w.r.t users with labeled examples and keywords with labeled examples respectively. The second term represents the consistency of the function f with label information and prior knowledge.

3.4 User Soft-score Weights Our proposed approach utilizes user behavior from labeled examples in computing the sentiment of the posts from the target domain. It is very likely that the sentiment labeling behavior of a user might vary across the source and target domains. For example, it is possible that a certain user has more positive reviews in the source domain and more negative reviews in the target domain. Such users degrade the performance of the classifier due to inconsistency in user behavior across the source and target domains. In extreme cases such inconsistency might lead to negative transfer learning.

In our approach we handle this issue by assigning non-negative soft-weights $v_{cu} \in [0, 1]$ to the set of common users $cu \in \mathcal{U}_c$ and $\mathcal{U}_c \subseteq \mathcal{U}$ from the source and target domains. We use the labeled examples from the source and target domains along with their keywords and sentiment labels to assign a soft-score to each shared user. The user soft-score weight calculation mechanism for each shared user across domains is as follows:

$$v_{cu} = \sum_i^{el^S} \sum_j^{el^T} sim(\mathbf{x}_i, \mathbf{x}_j) * y_i * y_j \tag{3.1}$$

where el^S and el^T represent the set of labeled examples for the user u in the source and target domains respectively, $\mathbf{x}_i \in \mathcal{X}^S$ and $\mathbf{x}_j \in \mathcal{X}^T$ represent the feature vectors for the examples in the source and target domains, $sim(\mathbf{x}_i, \mathbf{x}_j)$ is the cosine similarity between the feature vectors $\mathbf{x}_i$ and $\mathbf{x}_j$, finally, y_i and y_j are the corresponding sentiment labels for the examples i and j. In order to avoid negative transfer due to inconsistent user behavior across domains, the approach assigns smaller weights to inconsistent users. From the eq (3.1), the more consistent users have a positive value and more inconsistent users have a negative value. As the edge weights are always positive, we scale the user weights v_{cu} from $[-1, 1]$ to $[0, 1]$.

3.5 U-Cross Algorithm To minimize Q_1, we first set $f_{el} = y_{el}$, which requires the outputs of the classification function to be consistent with the known labels in the source and target domains, and then solve for f_{eu}, f_u, and f_d from the following lemma.

Lemma 3.1. *If $f_{el} = y_{el}$, Q_1 is minimized at:*

$$f^*_{eu} = [\alpha(\mathbf{S}^{(1,2)}_{eu})^T \mathcal{P} + \alpha(\mathbf{S}^{(1,2)}_{eu})^T \mathcal{P} + R] \\ [I - \alpha^2(\mathbf{S}^{(1,2)}_{eu})^T \mathbf{S}^{(1,2)}_{eu} - \alpha^2 \mathbf{S}^{(2,3)}_{eu} (\mathbf{S}^{(2,3)}_{eu})^T]^{-1} \tag{3.2}$$

$$f^*_u = \mathcal{P} + \alpha \mathbf{S}^{(1,2)}_{el} f^*_{eu} \tag{3.3}$$

$$f^*_d = \mathcal{Q} + \alpha(\mathbf{S}^{(2,3)}_{el})^T f^*_{eu} \tag{3.4}$$

where $\alpha = \frac{1}{1+\mu}$, $\mathcal{P} = \alpha \mathbf{S}^{(1,2)}_{el} y_{el} + (1-\alpha) y_u$, $\mathcal{Q} = \alpha(\mathbf{S}^{(2,3)}_{el})^T y_{el} + (1-\alpha) y_d$ *and* $\mathcal{R} = (1-\alpha) y_{eu}$

Proof. Please refer to the supplementary file ∎

From the equations (3.2), (3.3) and (3.4) computing f^*_u, f^*_{eu} and f^*_d requires solving matrix inversions which is a computationally intensive operation given the large size of the unlabeled examples and keyword features. To address this issue we consider the following iteration steps obtained after minimizing Q_1 to compute the optimal solution.

$$f_{eu}(t+1) = (1-\alpha) y_{eu} - \alpha((\mathbf{S}^{(1,2)}_{eu})^T f_u(t) + \mathbf{S}^{(2,3)}_{eu} f_d(t)) \tag{3.5}$$

$$f_u(t+1) = (1-\alpha) y_u - \alpha(\mathbf{S}^{(1,2)}_{el} y_L + \mathbf{S}^{(1,2)}_{eu} f_{eu}(t)) \tag{3.6}$$

$$f_d(t+1) = (1-\alpha) y_d - \alpha((\mathbf{S}^{(2,3)}_{el})^T y_{el} + (\mathbf{S}^{(2,3)}_{eu}))^T f_{eu}(t)) \tag{3.7}$$

where t is the number of iterations. The following theorem guarantees the convergence of these iteration steps:

Theorem 3.1. *When t goes to infinity, $f_{eu}(t)$ converges to f^*_{eu}, $f_u(t)$ converges to f^*_u and $f_d(t)$ converges to f^*_d.*

Proof. Please refer to the supplementary file ∎

Based on the above discussion, we present the *U-Cross* algorithm in Algorithm 1. Our algorithm *U-Cross* takes as input a set of m labeled examples as an example-keyword sparse binary matrix from the source domain, set of n examples as an example-keyword sparse binary matrix from the target domain among which a small subset $n\epsilon$ are labeled examples and the set of users $\mathcal{U}$ who authored the examples from the source and target domains. The algorithm outputs the labels of all the unlabeled examples from the target domain.

As an initial data processing step, we construct the affinity matrix $\mathbf{A}^{(3)}$ from the user-example and example-keyword affinity matrices. And then compute the degree matrix $\mathbf{D}^{(3)}$ and normalized symmetric matrix $\mathbf{S}^{(3)}$. As a preprocessing step, we calculate the covariate shift parameter weights w_i as discussed in Section 3.3 to reweight all the source domain examples.

Algorithm 1: *U-Cross* Algorithm

Input: Set of m labeled examples from source domain $\mathcal{X}^S$ and their labels $\mathcal{Y}^S$; set of n examples from target domain $\mathcal{X}^T$ and labels for first ϵn examples $\mathcal{Y}^T$; Users who authored the examples $\mathcal{U}$; the number of iterations t.

Output: Labels of all unlabeled examples in $\mathcal{X}^T$

1 Calculate the soft-score weights v_u for all the shared users according to eq (3.1). Set weights of user-user adjacency matrix $\mathbf{A}_u$ from v_u.

2 Set labeling function f_{el} to given labels y_{el}, $f_{el} = y_{el}$; Set initial user information y_u, unlabeled values y_{eu} and feature values y_d to zero if their prior values are not available. Initialize the corresponding functions $f_u(0)$, $f_{eu}(0)$ and $f_d(0)$ to y_u, y_{eu} and y_d respectively.

3 **for** $i \leftarrow 1$ ***to*** t **do**

4   Calculate $f_u(i)$ and $f_d(i)$ according to eq (3.6) and eq (3.7).

5   Calculate $f_{eu}(i)$ according to eq (3.5) and using the functions $f_u(i)$ and $f_d(i)$ calculated in previous step.

6 **end**

7 **for** $i \leftarrow (\epsilon n + 1)$ ***to*** n **do**

8   If $f_{eu}(t)$ at $x^T_i > 0$ then set $y^T_i = 1$ else set $y^T_i = -1$

9 **end**

In Step 1, we calculate the soft-score weights for all the shared users across the source and target domains to ensure consistency in sentiment labeling behaviors. As the only known prior values are the labels from the source domains, we initialize the function for labeled examples f_{el} to the known set of labels y_{el} and initialize the rest of the prior values and corresponding functions to 0. In Step 2, we learn the functions for users, unlabeled examples and keywords by label propagation using gradient method over t iterations. The functions are updated using the eq (3.6), eq (3.7) and eq (3.5). Finally, in the last step the sign of the function value for each unlabeled example is set as the sentiment label.

Based on the notation in Section 3.1, the following lemma shows the computational complexity of *U-Cross*:

Lemma 3.2. *The computational complexity of the* U-Cross *is given by* $\mathcal{O}\big(t(n+m)(u+d) + (p_{\max})^2 * (d_{\max})^2 * u\big)$, *where t is the number of* U-Cross *iterations, $p_{\max}$ is the maximum number of posts generated by a user, and $d_{\max}$ is the maximum number of keywords in a post.*

Proof. Please refer to the supplementary file ∎

From this lemma, we can see that *U-Cross* scales linearly with respect to the problem size (e.g., the number of examples in the source domain and the target domain, the size of the combined vocabulary space). Therefore, it can be naturally applied to large datasets.

4 Case Study

In this section we discuss how an existing method named TRITER [5] can be seen as a special case of *U-Cross*. TRITER uses both the keywords-labeled-unlabeled examples tripartite graph and a labeled-unlabeled examples bipartite graph to model the relationship between the source and the target domains, using high weights for examples from the target domain. However, in scenarios where a target domain example is mapped to both positive and negative examples from the source domain, the inclusion of bipartite graph could even harm the performance. Therefore, we ignore the bipartite graph (i.e., setting the corresponding weight to 0), and use a reweighting scheme to connect examples from the source domain and the target domain. More specifically, using the same notation as in the previous section, the objective function of TRITER can be written as follows.

$$\begin{aligned} Q_2(f) &= \frac{1}{2}\sum_{i}^{e} w_i \sum_{j}^{d} A_{i,j}^{(3)}\left(\frac{f_i}{\sqrt{D_i^{(3)}}} - \frac{f_j}{\sqrt{D_j^{(3)}}}\right)^2 \\ &+ \mu\sum_{i}^{e+d}(f_k - y_k)^2 - \gamma\sum_{l}^{e}(f_l^T f_{u(l)})^2 \\ &= f^T(I_{(e+d)\times(e+d)} - S^{(3)})f + \mu||f-y||^2 \\ &+ \beta||f_U||^2 - \gamma(||f_{el}^T U_{el}^T f_U||^2 + ||f_{eu}^T U_{eu}^T f_{eu}||^2) \end{aligned}$$

where μ, β and γ are positive parameters, w_i is the instance weight for labeled and unlabeled nodes and I is an identity matrix. U_{el} and U_{eu} are matrices of size $u \times m$ and $u \times n$ respectively that map users to labeled and unlabeled examples. Matrices U_{el} and U_{eu} can be compared to $A_{el}^{(1,2)}$ and $A_{eu}^{(1,2)}$ matrices in the tripartite graph mentioned in previous section. The extension includes adding a regularizer on user behavior function and also on the user-example interaction. The last equation in minimization function Q_2 captures the interaction between users and different labeled and unlabeled examples in the graph which needs to be maximized.

Comparing all the terms in Q_1 and Q_2, we can see that both equations are similar. By setting $\beta = 1$ and $\gamma = 2$, it is possible to rewrite equation Q_2 in terms of Q_1 with minimal difference. The major difference between *U-Cross* and TRITER is that TRITER does not model user behavior. From the objective function Q_2, it can be seen that TRITER is a special case of *U-Cross* without user behavior. Therefore, *U-Cross* is expected to perform better that TRITER since it explicitly models the human factor.

5 Results

In this section we report the experimental results. We first introduce three real-world cross-domain sentiment datasets related to product reviews. Then we compare different user soft-weight scoring approaches. Finally, we compare the *U-Cross* with other state-of-the-art methods to demonstrate its effectiveness.

5.1 Data Sets To compare our transfer learning approach *U-Cross* we perform experiments on three different real-world datasets. User reviews from three different product review websites are used for the sentiment classification task. Table 1 describes the dataset statistics. The dataset details are as follows:

1. Amazon product reviews[2]: The dataset is a part of Stanford Network Analysis Project [10] and includes amazon product reviews from 28 different product categories. For experimental evaluation, we created six different datasets with varying common user frequency from *office products*, *software*, *toy games*, *video games*, *electronics*, *amazon videos*, *kitchen*, *movies* and *music* product categories.
2. Yelp reviews[3]: The data set is from Yelp Data set challenge and includes user reviews from *restaurants* and *shopping* domains.
3. CIAO dataset[4]: The dataset is crawled from the CIAO website and consists of consumer reviews of *books* and *beauty* products.

Several preprocessing steps were taken before experiments. Words were converted to lower cases and then stemmed to terms. All the stop words, punctuation and symbols are removed. Binary feature vector as a bag of words on n-grams $n = \{1, 2, 3\}$ was extracted for each review. Also, we dropped those users with less than three reviews and more than hundred reviews in the source and target domains to ensure consistent and unbiased user contribution. Features with document (product reviews) frequency less than 10 are also dropped. Table 1 reports the size of the feature vector on the entire vocabulary space for the source and target domains combined. Moreover as explained in Section 3.5, the source domain examples are reweighted to reduce the covariate shift across the domains.

[2]http://www.amazon.com
[3]http://www.yelp.com/dataset_challenge
[4]http://www.ciao.co.uk

	Source domain	Target domain	Examples	Examples	Unique Users	Source Users	Target Users	Common Users	Features
Amazon	office products	software	2225	3968	1362	588	832	58	9069
	toy games	video Games	5760	16700	3937	1153	3127	343	26982
	electronics	amazon Video	41340	72976	21102	9632	12770	1300	144669
	electronics	kitchen	44518	56296	15126	9279	8952	3105	118418
	amazon videos	music	73448	478414	78842	11538	74149	6845	1704032
	amazon videos	movies	73177	533418	82665	17799	79677	14811	1908543
Yelp	restaurants	shopping	5502	19338	622	510	345	310	95094
CIAO	beauty	books	3524	6734	339	217	195	73	64429

Table 1 Dataset statistics

5.2 User Selection As prior research demonstrated that by using user information along with linguistic features improved the performance of sentiment classifiers, we employ a robust user selection approach proposed in Section 3.4 to assign soft-score weights to all the common set of users $\mathcal{U}_c$ in the source and the target domains. In order to avoid negative transfer due to inconsistent user behavior across domains, the approach assigns larger weights to more consistent users with similar product labeling behavior across domains and smaller weights to inconsistent users.

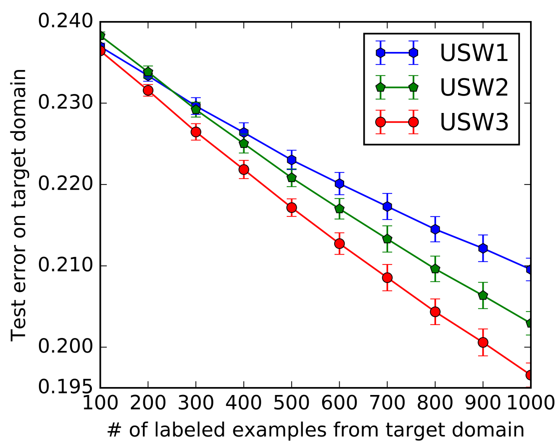

Figure 2: Comparison of user selection approaches USW1, USW2 and USW3 described in Section 5.2. The x-axis represents # of labeled examples from the target domain and the y-axis represents test error on the target domain data.

Let $\mathcal{U}^S$ and $\mathcal{U}^T$ be the set of unique users in the source and target domains respectively. To evaluate the effectiveness of our user selection approach, we consider the following variations of the soft-score user weights:

1. **USW1**: The baseline approach that assigns unit weights to all the users $\mathcal{U}$.

$$v_u = 1, \forall u \in \{\mathcal{U}^S \cup \mathcal{U}^T\}$$

2. **USW2**: Set all the user weights for shared users across domains as per the proposed approach and the rest to 0.

$$v_u = \begin{cases} v_{cu}, & \forall u \in \mathcal{U}_c \\ 0, & \forall u \in\in \{\mathcal{U}^S \cup \mathcal{U}^T\}\backslash\mathcal{U}_c \end{cases}$$

3. **USW3**: Set all the user weights for shared users across domains as per the proposed approach and the non-shared users in the target domain to 1.

$$v_u = \begin{cases} v_{cu}, & \forall u \in \mathcal{U}_c \\ 0, & \forall u \in \mathcal{U}^S\backslash\mathcal{U}_c \\ 1, & \forall u \in \mathcal{U}^T\backslash\mathcal{U}_c \end{cases}$$

Figure 2 compares the performance of different user soft-score weighting approaches USW1, USW2 and USW3 on the Amazon reviews data set (*electronics* $\rightarrow$ *amazon videos*). It can be observed that USW3 performs the best compared to USW1 and USW2. In congruence with previous findings, leveraging the knowledge from associations between users and examples USW3 performed better compared to not using the associations between users and examples USW2. The approach with equal weights to all the users (consistent and inconsistent) performed the worst because of the negative transfer effect often associated with transfer learning. In all the following experiments, unless specified otherwise, the user soft-score weights in *U-Cross* refers to USW3.

5.3 Empirical Analysis To show the robustness of *U-Cross* approach we run the experiments by resampling the labeled examples from the target domain. We report the results with confidence scores from 20 runs. The target domain examples are carefully chosen to minimize the class- and user-bias. By class bias we choose target domain examples with fairly equal proportions of positive and negative class labels. Also we ensure that the chosen labeled target examples maximize the set of common users across the source and target domains. We compare our *U-Cross* approach with other state-of-the-art methods and report the test error on the unlabeled examples from the target domain. The methods to be compared include: SCL [3], where for each data set, 2000 pivot features are selected from the source and target domains; TCA [8], which utilizes both the shared and the mapped domain-specific topics to span a new shared feature space for knowledge

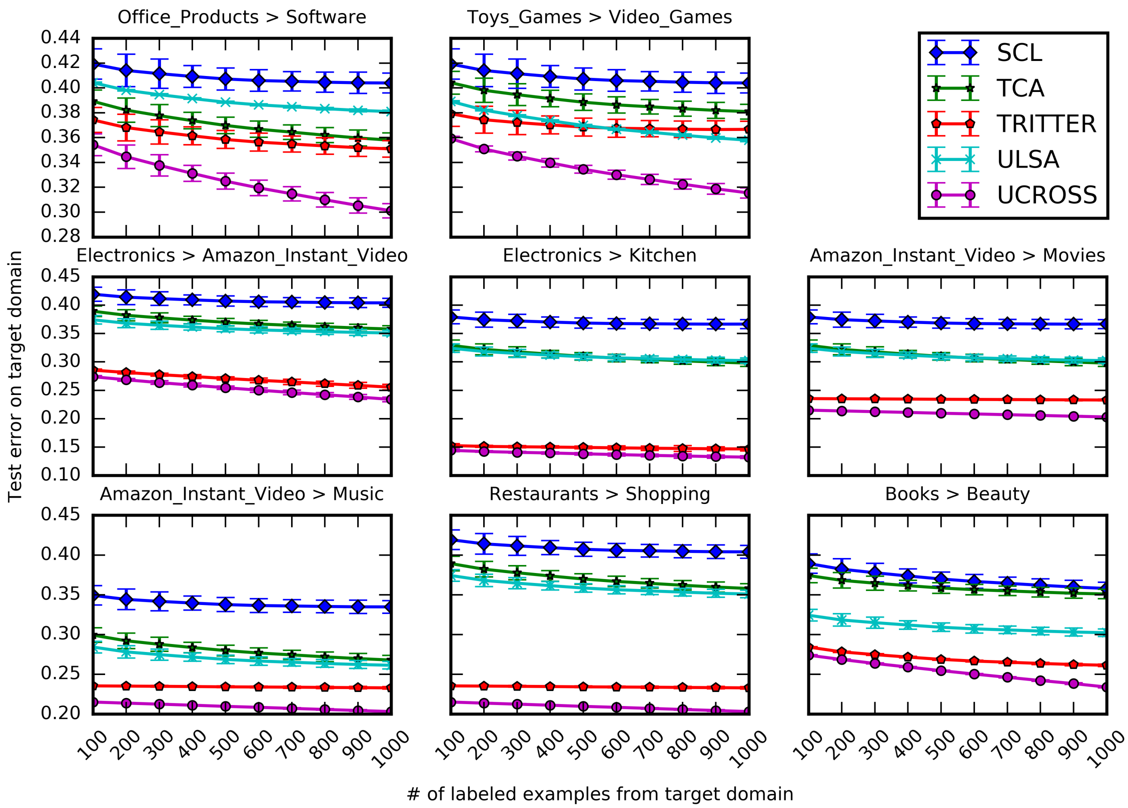

Figure 3: Performance evaluation on various datasets. In each subfigure, the title represents the source and target domains, the x-axis represents # of labeled examples from the target domain and the y-axis represents test error with error bars on the target domain over 20 runs. In each run the labeled data from target domain is resampled.

Source domain	Target domain	SCL	TCA	TRITER	ULSA	U-Cross
Office Products	Software	0.414 ± 0.007	0.358 ± 0.005	0.349 ± 0.006	0.381 ± 0.001	**0.301** ± **0.005**
Toy Games	Video Games	0.430 ± 0.006	0.381 ± 0.005	0.367 ± 0.006	0.358 ± 0.001	**0.315** ± **0.004**
Electronics	Amazon Videos	0.424 ± 0.005	0.351 ± 0.003	0.255 ± 0.001	0.351 ± 0.003	**0.234** ± **0.001**
Electronics	Kitchen	0.367 ± 0.003	0.298 ± 0.004	0.147 ± 0.001	0.302 ± 0.004	**0.132** ± **0.001**
Amazon Videos	Music	0.367 ± 0.004	0.298 ± 0.003	0.233 ± 0.001	0.302 ± 0.002	**0.203** ± **0.001**
Amazon Videos	Movies	0.335 ± 0.003	0.268 ± 0.004	0.221 ± 0.001	0.261 ± 0.003	**0.196** ± **0.001**
Restaurants	Shopping	0.404 ± 0.006	0.358 ± 0.004	0.228 ± 0.001	0.351 ± 0.004	**0.201** ± **0.001**
Beauty	Books	0.358 ± 0.005	0.351 ± 0.004	0.261 ± 0.001	0.302 ± 0.003	**0.234** ± **0.001**

Table 2 Performance comparison of *U-Cross* with other methods.

transfer; TRITER which leverages labeled-unlabeled-keywords to propagate sentiment information from labeled examples to unlabeled examples; and ULSA [17], which performs user-level sentiment analysis incorporating social networks with user-user relationship parameter $\lambda_k = 0$.

The parameter selection for *U-Cross* is performed through 10-fold cross validation on labeled examples from the source and target domains on different datasets. Labeled examples from target domain are randomly sampled over 20 runs to ensure robust parameter selection. A total of 1000 examples are sampled from target domain for parameter selection. From the empirical results, setting the regularization parameter to $\alpha = 0.1$ resulted in best performance. So, we have set the regularization parameter of *U-Cross* to $\alpha = 0.1$ for comparison with other state-of-the-art methods.

The experimental results are summarized in Table 2. Figure 3 compares the performance of *U-Cross* on different datasets from Table 1. In each figure, the x-axis represents number of labeled examples from the target domain and the y-axis represents test error with error bars on the target domain over 20 runs. First, our *U-Cross* approach outperforms all other methods in terms of test error on unlabeled examples from target domain in all the datasets. This validates the effectiveness of leveraging user information for cross-domain sentiment classification of user reviews. Second, the variation is significant in datasets with large user network which shows that user behavior plays significant role in large scale sentiment classification tasks.

6 Conclusion

In this paper we proposed *U-Cross*, a novel graph-based transfer learning approach that explicitly models the human factor for cross-domain sentiment analysis. In *U-Cross*, we used the user-example-features tripartite graph to propagate sentiment information from labeled examples, users and keyword features to the unlabeled examples. Based on the tripartite graph, we proposed an effective optimization algorithm, which is guaranteed to converge to the global optimum. Also, from the time complexity analysis of the algorithm we showed that *U-Cross* scales linearly with respect to the problem size (e.g., the number of examples in the source domain and the target domain, the size of the combined vocabulary space). We also showed how a previously proposed approach TRITER is a special case of our non-parametric approach *U-Cross*. We also proposed an effective approach to choose common users across the source and target domains to avoid negative transfer. Empirical comparison with other state-of-the-art transfer learning based sentiment classification approaches showed that explicitly modeling the user behaviors leads to improved performance. The *U-Cross* approach is generalizable and scalable to multiple sources easily.

Acknowledgements

This work is supported by NSF research grants IIS-1552654 and IIS-1651203, ONR research grant N00014-15-1-2821, an IBM Faculty Award, DTRA under the grant number HDTRA1-16-0017, Army Research Office under the contract number W911NF-16-1-0168, National Institutes of Health under the grant number R01LM011986, the U.S. Department of Homeland Securitys VACCINE Center under Award Number 2009-ST-061-CI0001, Region II University Transportation Center under the project number 49997-33 25 and a Baidu gift. The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agencies or the government.

References

[1] S. Bickel, M. Brückner, and T. Scheffer. Discriminative learning under covariate shift. *JMLR*, 10:2137–2155, 2009.

[2] J. Blitzer, M. Dredze, F. Pereira, et al. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*, volume 7, pages 440–447, 2007.

[3] J. Blitzer, S. Kakade, and D. P. Foster. Domain adaptation with coupled subspaces. In *AI Statistics*, pages 173–181, 2011.

[4] X. Glorot, A. Bordes, and Y. Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *ICML*, pages 513–520, 2011.

[5] J. He, Y. Liu, and R. Lawrence. Graph-based transfer learning. In *CIKM*, pages 937–946. ACM, 2009.

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

[7] F. Li, S. Wang, S. Liu, and M. Zhang. Suit: A supervised user-item based topic model for sentiment analysis. In *AAAI*, 2014.

[8] L. Li, X. Jin, and M. Long. Topic correlation analysis for cross-domain text classification. In *AAAI*, 2012.

[9] B. Liu. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167, 2012.

[10] J. McAuley and J. Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *ACM RecSys*, pages 165–172. ACM, 2013.

[11] T. Nasukawa and J. Yi. Sentiment analysis: Capturing favorability using natural language processing. In *CIKM*, pages 70–77. ACM, 2003.

[12] S. J. Pan, X. Ni, J.-T. Sun, Q. Yang, and Z. Chen. Cross-domain sentiment classification via spectral feature alignment. In *WWW*, pages 751–760. ACM, 2010.

[13] S. J. Pan, D. Shen, Q. Yang, and J. T. Kwok. Transferring localization models across space. In *AAAI*, pages 1383–1388, 2008.

[14] S. J. Pan and Q. Yang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE*, 22(10):1345–1359, 2010.

[15] M. Sugiyama, S. Nakajima, H. Kashima, P. V. Buenau, and M. Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *NIPS*, pages 1433–1440, 2008.

[16] B. Tan, Y. Song, E. Zhong, and Q. Yang. Transitive transfer learning. In *SIGKDD*, pages 1155–1164. ACM, 10 Aug. 2015.

[17] C. Tan, L. Lee, J. Tang, L. Jiang, M. Zhou, and P. Li. User-level sentiment analysis incorporating social networks. In *KDD*, pages 1397–1405. ACM, 2011.

[18] D. Tang, B. Qin, and T. Liu. Learning semantic representations of users and products for document level sentiment classification. In *ACL*, pages 1014–1023, Beijing, China, July 2015. ACL.

[19] X. Wan. Co-training for cross-lingual sentiment classification. In *ACL and AFNLP*, ACL '09, pages 235–243, Stroudsburg, PA, USA, 2009. ACL.

[20] Y. Zhou and J. He. Crowdsourcing via tensor augmentation and completion. In *IJCAI*, pages 2435–2441, 2016.

[21] Y. Zhu, Y. Chen, Z. Lu, S. J. Pan, G.-R. Xue, Y. Yu, and Q. Yang. Heterogeneous transfer learning for image classification. In *AAAI*, 2011.