Online User Representation Learning Across Heterogeneous Social Networks

Weiqing Wang
teresa.wang@monash.edu
Monash University

Hongzhi Yin*
h.yin1@uq.edu.au
The University of Queensland

Xingzhong Du
xingzhong.du@coles.com.au
Coles Advanced Analytics Centre

Wen Hua
w.hua@uq.edu.au
The University of Queensland

Yongjun Li
lyj@nwpt.edu.cn
Northwestern Polytechnical University

Quoc Viet Hung Nguyen
quocviethung1@gmail.com
Griffith University

ABSTRACT

Accurate user representation learning has been proven fundamental for many social media applications, including community detection, recommendation, etc. A major challenge lies in that, the available data in a single social network are usually very limited and sparse. In real life, many people are members of several social networks in the same time. Constrained by the features and design of each, any single social platform offers only a partial view of a user from a particular perspective. In this paper, we propose MV-URL, a **m**ulti-**v**iew **u**ser **r**epresentation **l**earning model to enhance user modeling by integrating the knowledge from various networks. Different from the traditional network embedding frameworks where either the whole framework is single-network based or each network involved is a homogeneous network, we focus on multiple social networks and each network in our task is a heterogeneous network. It's very challenging to effectively fuse knowledge in this setting as the fusion depends upon not only the varying relatedness of information sources, but also the target application tasks. MV-URL focuses on two tasks: user account linkage (i.e., to predict the missing true user account linkage across social media) and user attribute prediction. Extensive evaluations have been conducted on two real-world collections of linked social networks, and the experimental results show the superiority of MV-URL compared with existing state-of-art embedding methods. It can be learned online, and is trivially parallelizable. These qualities make it suitable for real world applications.

CCS CONCEPTS

• **Information systems → Collaborative and social computing systems and tools**; • **Human-centered computing → Collaborative and social computing**; • **Applied computing → Computer forensics**.

*This is the Corresponding Author

SIGIR '19, July 21–25, 2019, Paris, France

ACM ISBN 978-1-4503-6172-9/19/07…$15.00
https://doi.org/10.1145/3331184.3331258

KEYWORDS

user modelling, social networks, multiview, representation learning, heterogeneous networks

ACM Reference Format:
Weiqing Wang, Hongzhi Yin, Xingzhong Du, Wen Hua, Yongjun Li, and Quoc Viet Hung Nguyen. 2019. Online User Representation Learning Across Heterogeneous Social Networks. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19), July 21–25, 2019, Paris, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3331184.3331258

1 INTRODUCTION

The blossom of various social services has boosted many business intelligence by leveraging the big social data. In particular, people wonder how to gain a deeper and better understanding of each individual user from the vast amount of social data out there.

Most existing work focus on modeling users with their behavior data from a single social platform [13, 15, 32, 34]. Unfortunately, information of a user from a single platform are usually very limited and sparse (e.g., the sparsity of user-item behavior data on a single social network is usually higher than 99.9% [41]). The key to unleashing the true power of social media analysis is to link up all the data of the same user across different social platforms [17, 36, 41], which is an important topic covered in the broad learning task [37, 39, 42]. The benefit of integrating the information from multiple social platforms for user profiling are three-fold [17]. The first benefit is to complete users' information by offering different views of a user. Constrained by the features and design of a single platform, it can only offer a partial view of a user from a particular perspective. For example, many people use Facebook to communicate with their acquaintances, but post their real-time information on Twitter. The second advantage lies in that multiple networks can help in correcting the possible false or conflicting information provided by users on a single social platform. The last well-being is to integrate useful user information from those social platforms that have become less popular or even abandoned over time.

Data across multiple social networks can be very difficult to deal with due to their complex structures (containing heterogeneous links and nodes). Traditional machine learning algorithms face great challenges in handling such network data as they usually take feature vectors as the input. Thus, a general representation of users by fusing information from heterogeneous networks as feature vectors is desired for many user-related knowledge discovery on such complex-structured data, including community detection, recommendation, etc. Representing users with low dimension feature vectors is also known as user embedding. In this paper, we focus on effective user embedding by jointly learning users' related information across multiple social platforms.

Many works have been proposed to embed the online social network data into a low-dimensional feature space. However, most existing social network embedding methods are proposed for homogeneous networks [11, 22, 30], which learn the feature vectors for user nodes merely based on the social connections among them. Recently, several research work have been proposed to embed the heterogeneous networks [1, 4, 33]. But different from our research task which focuses on multiple networks, they are single-network based. Technically, the research task in this paper is actually to embed users with "multiple heterogeneous networks". More specifically, we study multiple networks and every network involved is a heterogeneous network containing different links (e.g., User-User, User-Item, etc) and diverse nodes.

Qu et al. propose a multi-view network representation learning in [25], which integrates the information from multiple types of relationships between the *same group of nodes*. The structure of the studied network by Qu et al. is actually multiple homogeneous networks. Specifically, these networks are coupled by the same group of nodes, and the links and nodes inside each network are homogeneous. Thus, the research task in [25] is different from our task in this paper. The research work that is most related to our focused network structure is DIME (i.e., Deep alLgned autoencoder based eMbEdding), proposed in [39]. DIME is designed for multiple heterogeneous social networks. However, the aim of DIME is to transfer the user proximity information from a dense network to a sparse network, and a set of meta paths need to be *predefined* to represent the general correlation among users.

In this paper, we propose a general user representation learning framework to jointly and automatically learn and fuse users' related information across multiple heterogeneous social networks. It's very challenging to construct such a framework to automatically and effectively integrate relevant knowledge across various platforms as this integration depends upon not only the different relatedness of these information sources, but also the target application problems. In this paper, we focus on two application tasks: user account linkage and user attribute prediction. Most existing work on user account linkage [5, 17, 19] focus on *pair-wise* user linkage between *two* platforms and most existing work on users' attributes prediction [2, 23, 24] only leverage the data from single social platforms. Our proposed framework is able to achieve these two tasks with the information from multiple social platforms.

To model the various relatedness of these information sources, we design different schemes in integrating sources of information. We assume that all the social networks share the same semantic space by restraining that a content word has the same latent representations across different social networks. For the coupled graphs inside of one single social network, we assume that they have higher homogeneity compared with the graphs across social networks. Thus, we introduce different schemes to model the heterogeneity inside of one single social platform and across different social platforms. Specifically, to reconcile the heterogeneities of different latent spaces on single networks, we use a harmonious embedding matrix [4] to further embed the embeddings from one latent space to another latent space. On the other hand, a co-regularization scheme, which has been widely used in multi-view learning [9, 25, 27], is introduced to promote the collaboration of different networks.

As we need to learn a new harmonious embedding matrix for each extra coupled graph on one single social platform, which is really expensive and easy to lead to over fitting when training data is sparse in the real world, MV-URL constrains the harmonious matrix to be diagonal, to reduce the number of parameters to be learned. Different sampling methods are applied in MV-URL. Specifically, hierarchical sampling method is adopted to sample positive instances and bi-direction strategy is used to sample negative instances for each positive instance. A two-step optimization method (i.e., alternating optimization algorithm is adopted for the initialization of node vectors on single networks because of the convex objective functions, and a combination of Stochastic Gradient Descent and back-propagation algorithm is used to optimize the collaborations across social networks) is proposed to learn the parameters of MV-URL. To further ensure that the proposed model is suitable for real world applications, we also discuss how to parallelize the optimization and how to extend it to the streaming scenarios.

Extensive experiments have been conducted on two collections of real world linked networks. One collection is publicly available containing two linked networks while another one is contributed by this paper which contains three social networks sharing users. The evaluation results show that MV-URL outperforms state-of-the-art embedding approaches of both individual networks and multiple networks.

2 PROBLEM FORMULATION

As we aim at user representation, for each social network, we only model user-related information. Specifically, we focus on these information: users' social information (User-User graph) and users' preference information (User-Item graph and User-Word graph).

Definition 2.1. (**User-User Graph**) The User-User graph on a social network is defined as $G_u = (\mathcal{U}, \mathcal{E}_u, \mathcal{W}_u)$ where $\mathcal{U}$ are vertexes representing users, $\mathcal{E}_u$ are edges within G_u denoting the social relationships between users and $\mathcal{W}_u$ are edge weights.

Definition 2.2. (**User-Item Graph**) The User-Item graph on a social network is defined as a graph $G_{uv} = (\mathcal{U} \cup \mathcal{V}, \mathcal{E}_{uv}, \mathcal{W}_{uv})$ where $\mathcal{V}$ defines the items, $\mathcal{E}_{uv}$ are edges within G_{uv} denoting users' interaction information with items and $\mathcal{W}_{uv}$ correspond to edge weights.

Definition 2.3. (**User-Word Graph**) The User-Word graph on a social network is defined as a graph $G_{uc} = (\mathcal{U} \cup C, \mathcal{E}_{uc}, \mathcal{W}_{uc})$ where C represents content words, $\mathcal{E}_{uc}$ are edges within G_{uc} denoting users' interaction information with content words and $\mathcal{W}_{uc}$ correspond to edge weights.

Definition 2.4. (**Social Network, View**) A social network or view is denoted as $G = (G_u, G_{uv}, G_{uc}) = (\mathcal{U}, \mathcal{V}, C, \mathcal{E}_u, \mathcal{E}_{uv}, \mathcal{E}_{uc}, \mathcal{W}_u, \mathcal{W}_{uv}, \mathcal{W}_{uc})$ containing User-User graph, User-Item graph and User-Word graph, integrating both users' social information and preference information on this social network.

Figure 1 shows the structure of multiple linked heterogeneous networks that we study in this paper. A user may have accounts on different social networks, our aim in this paper is to learn robust low

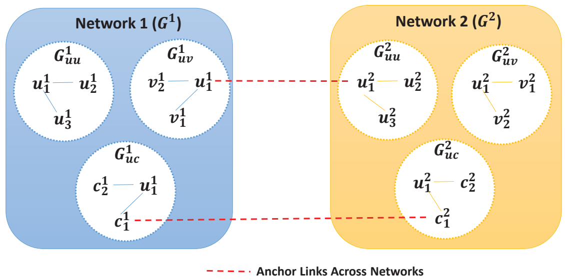


Figure 1: The Structure of Multiple Linked Heterogeneous Networks. a) Note that, in this paper, we assume that users and content words both might be shared across social networks, which is usually the case in the real world. We argued user sharing and it's also natural for the content sharing as most social media share the very similar semantic space. b) u_i^k, v_i^k and c_i^k represent the i^{th} user, item and content word on k^{th} social network (i.e., G^k), respectively

dimensional vector representations for users, which are consistent across different social networks.

Problem 1. ***(Multi-view User Embedding)*** *Given K ($K > 1$) social networks $\mathcal{G} = (G^1, G^2, \ldots, G^k, \ldots, G^K)$ where $G^k = (G_u^k, G_{uv}^k, G_{uc}^k) = (\mathcal{U}^k, \mathcal{V}^k, \mathcal{C}^k, \mathcal{E}_u^k, \mathcal{E}_{uv}^k, \mathcal{E}_{uc}^k, \mathcal{W}_u^k, \mathcal{W}_{uv}^k, \mathcal{W}_{uc}^k)$ is the k^{th} social network, and the linkage information across them (not all users are shared and the users can be shared by two or more networks), the problem of multi-view social network embedding is to learn a low dimensional vector representation for each user in $\mathcal{U} = \mathcal{U}^1 \cup \mathcal{U}^2 \cup \ldots \cup \mathcal{U}^K$.*

3 MULTI-VIEW USER EMBEDDING

In this paper, we follow the common symbolic notation, where upper case bold letters denote matrices, and lower case bold letters denote column vectors. To solve the Problem 1, our approach first mines the social information and preference information encoded in single views (Section 3.1.1). After that, we further integrate different views to vote for more robust user representations (Section 3.1.2). During voting, we automatically learn the voting weights of views through an attention based approach (Section 3.2).

The overall objective of our approach is summarized below:

$$O = O_{co} + O_{at} \tag{1}$$

where O_{co} is the objective function of the collaboration framework, in which we aim to learn the node proximities in individual social networks and meanwhile vote for the robust node representations. O_{at} is the goal function for weight learning of each social network.

3.1 Collaboration Embedding Framework

The aim of this framework is to capture the vertex proximity encoded in each social network and meanwhile integrate them to vote for the robust embeddings across social networks.

3.1.1 Embedding on Single Networks. MV-URL aims at maintaining the proximities between vertexes directly connected by edges on each social network. The probability of linkage measured in latent space is quantified by the following equation:

$$p(\boldsymbol{u}_i, \boldsymbol{u}_j) = \frac{1}{1 + exp\{-\boldsymbol{u}_i^T \boldsymbol{u}_j\}} \tag{2}$$

where $\boldsymbol{u}_i$ and $\boldsymbol{u}_j$ are column vectors of embedding for vertexes indexed by i and j in the User-User graph G_u, respectively.

There are multiple graphs on each single social network, which are coupled together with shared user set $\mathcal{U}$. There are embedding methods available for individual graph [11, 30, 31]. However, it is not straightforward to extend them to coupled graphs. The major challenge is imposed by heterogeneous characteristics of multiple graphs, which would result in heterogeneous latent spaces. As a result, embeddings of different graphs cannot be directly matched.

MV-URL embeds the embeddings from one latent space to another by harmonious embedding matrices [33]. However, it's very challenging to effectively learn harmonious embedding matrices because of the data sparsity problem. More specifically, there are multiple graphs (i.e., this paper focuses on three graphs on each social network but MV-URL aims at solving the problem of embedding across multiple heterogeneous networks and thus is expected to be capable of easily extending to multiple graphs), and thus we need to learn *multiple* harmonious embedding matrix. This can easily lead to over-fitting problem especially when data is sparse, which is usually the case on social networks. To deal with this challenge, MV-URL constrains the matrix to be *diagonal*.

Specifically, MV-URL enables the transformation between social spaces (i.e., User-User graph) and preference spaces (i.e., User-Item graph and User-Word graph) by multiplying two diagonal matrices: $diag(m_1^{uv}, \ldots, m_d^{uv})$ and $diag(m_1^{uc}, \ldots, m_d^{uc})$, where d is the dimension of latent features. Here, we take the transformation from User-User graph to User-Item graph for an example. The probability of linkage between vertexes of User-Item graph G_{uv} can be quantified as follows:

$$p(\boldsymbol{u}_i, \boldsymbol{v}_j) = \frac{1}{1 + exp\{-\boldsymbol{u}_i^T diag(m_1^{uv}, \ldots, m_d^{uv}) \boldsymbol{v}_j\}} \tag{3}$$

The loss function on each individual graph is as Equation 4 (e.g., take G_u as an example).

$$O(G_u) = -\sum_{(i,j)\in\mathcal{E}_u} (w_u)_{ij} \times log(p(\boldsymbol{u}_i, \boldsymbol{u}_j)) - \sum_{(h,k)\notin\mathcal{E}_u} log(1 - p(\boldsymbol{u}_h . \boldsymbol{u}_k)) \tag{4}$$

To ensure the scalability of MV-URL on large-scale and sparse social networks, negative sampling scheme [25, 29, 33] is adopted. In this paper, we adopt the *bi-direction* strategy to sample negative instances for each positive instance. For example, for an edge $(i, j) \in \mathcal{E}_{uv}^k$, we first fix user u_i and randomly sample $\lceil \frac{N}{2} \rceil$ negative items, and then fix item v_j and randomly sample $\lfloor \frac{N}{2} \rfloor$ negative users. The loss functions for G_{uv}, and G_{uc} are all defined in this way. To enable the complementary information inside each graph affecting each other, we jointly optimize these loss functions for each individual graph as follows:

$$\begin{aligned} O(G^k) = \\ &- \sum_{(i,j)\in\mathcal{E}_u^k} (w_u^k)_{ij} \times [log(p(\boldsymbol{u}_i^k, \boldsymbol{u}_j^k)) + \sum_{(i'_n,j'_n)\notin\mathcal{E}_u^k}^{N} log(1 - p(\boldsymbol{u}_{i'_n}^k . \boldsymbol{u}_{j'_n}^k))] \\ &- \sum_{(i,j)\in\mathcal{E}_{uv}^k} (w_{uv}^k)_{ij} \times [log(p(\boldsymbol{u}_i^k, \boldsymbol{v}_j^k)) + \sum_{(i'_n,j'_n)\notin\mathcal{E}_{uv}^k}^{N} log(1 - p(\boldsymbol{u}_{i'_n}^k . \boldsymbol{v}_{j'_n}^k))] \\ &- \sum_{(i,j)\in\mathcal{E}_{uc}^k} (w_{uc}^k)_{ij} \times [log(p(\boldsymbol{u}_i^k, \boldsymbol{c}_j)) + \sum_{(i'_n,j'_n)\notin\mathcal{E}_{uc}^k}^{N} log(1 - p(\boldsymbol{u}_{i'_n}^k . \boldsymbol{c}_{j'_n}))] \end{aligned} \tag{5}$$

where N is the number of negative data instances sampled for each positive data instance, $\boldsymbol{u}_i^k$ and $\boldsymbol{v}_i^k$ are the embeddings for i^{th} user and item on k^{th} platform respectively, and $\boldsymbol{c}_j$ is the latent vector for j^{th} content word. In this Equation, the regularization norms are omitted and we will illustrate them in the following part.

Note that, in this paper, we assume that all social platforms share same content space while different user and item spaces. This is reasonable as one single content word generally has the same semantic meaning even on different social platforms while different platforms focus on varying aspects for the same users or items [41] (e.g., Facebook reflect users' interaction with acquaintances in daily life while Youtube focus on users' preferences on videos).

3.1.2 Collaboration Embedding. By minimizing the Equation 5 on each platform, the view-specific representations $\{U^k\}_{k=1}^K$ are able to preserve the structure information encoded in individual platforms. In this part, we focus on promoting the collaboration of different platforms to learn a robust representations for users. We introduce the co-regularization scheme [25] to achieve this.

The problem setting in [25] is that all users have to be shared across all views , but this is usually not the case in real world. A user might have accounts on one single social platform or a subset of social platforms. These two types of users are modeled in different ways in MV-URL. Assume that there are S users shared across a subset of social platforms, We use a set $\mathcal{U}^s$ to define these shared users. For each shared user u, the subset of social platforms having u's account is defined as $\mathcal{S}_u$. Note that, $|\mathcal{S}_u|$ is not necessarily equal to K. We introduce the following regularization term for each shared user u:

$$R_u^{co} = \sum_{k \in \mathcal{S}_u} \lambda_u^k \parallel \boldsymbol{u}^k - \boldsymbol{u} \parallel_2^2 \tag{6}$$

where λ_u^k is the weight of platform k in voting for the robust representation for the shared user u, $\boldsymbol{u}$ is the robust representation for u, and $\boldsymbol{u}^k$ is the representation for u on k^{th} social platform. Intuitively, by learning proper weights λ_u^k for each shared user u, MV-URL focuses on the most informative views in learning u's embedding. We will introduce how MV-URL automatically learns such weights in Section 3.2. By minimizing Equation 6, representations on each social platform will vote for the robust representations based on Equation 7. This vote process is quite intuitive. Naturally, the robust embeddings are calculated as the weighted combinations of the view-specific representations with coefficients as the voting weights of each view.

$$\boldsymbol{u} = \sum_{k \in \mathcal{S}_u} \lambda_u^k \boldsymbol{u}^k \tag{7}$$

The final objective of our collaboration embedding framework can be summarized below:

$$O_{co} = \sum_{k=1}^{K} O(G^k) + R \tag{8}$$

$$R = \alpha R^u + \beta \sum_{k=1}^{K} \sum_{n=1}^{N_v^k} \parallel \boldsymbol{v}_n^k \parallel_2^2 + \gamma \sum_{n=1}^{N_c} \parallel \boldsymbol{c}_n \parallel_2^2 + \delta \sum_{k=1}^{K} \parallel \boldsymbol{M}_{uv}^k \parallel_2^2 + \epsilon \sum_{k=1}^{K} \parallel \boldsymbol{M}_{uc}^k \parallel_2^2 \tag{9}$$

$$R^u = \sum_{u \in \mathcal{U}^s} R_u^{co} + \sum_{k=1}^{K} \sum_{n=1}^{N_{\bar{s}}^k} \parallel \boldsymbol{u}_n^k \parallel_2^2 \tag{10}$$

where α, β, γ, δ and $\epsilon \in R$ are regularization coefficients, N_u^k and N_v^k are the numbers of users and items in k^{th} social platform respectively, N_c is the number of content words, and $N_{\bar{s}}^k$ is the number of users who have accounts on k^{th} platform only and are not shared across platforms.

3.2 View Weight Learning Model

In this part, we present an attention-based method to automatically compute the weights for each social platform in voting the robust node representations for the shared users. The weight learning should be both task-specific and user-specific. First, the weights learning depends on application tasks. For example, one user may prefer Facebook over Twitter for communicating with daily acquaintance while use Twitter more frequently instead for sharing real-time status. On the other hand, the weight learning should be personalized [41] as users usually play different roles and get different levels of influence from their neighbors in different networks.

We focus on two applications: user attribute prediction and user linkage prediction. We treat the prediction of one user attribute as a multi-label classification task. For user account linkage prediction, we focus on this scenario: given $\mathcal{G} = \{G^1, G^2, \ldots, G^K\}$ where $K \geq 2$ and the user account linkage information inside of $\mathcal{G}$ are known and another social network G^{K+1} which shares users with $\mathcal{G}$ but the linkage information between $\mathcal{G}$ and G^{K+1} are partially given, the task is to predict the unknown user linkage information between $\mathcal{G}$ and G^{K+1}. In other words, we already know the user linkage information on two or more existing social networks (i.e., $\mathcal{G}$), and then we receive a new social network which also share some users with the existing networks but the linkage information is only partially known and to be predicted for the remaining.

To make weight learning user aware, personalized softmax units are used to define the weight of view k for user u_i as follows:

$$\lambda_i^k = \frac{exp(z_k \cdot \boldsymbol{u}_i^C)}{\sum_{k'=k_1}^{k_n} exp(z_{k'} \cdot \boldsymbol{u}_i^C)} \tag{11}$$

where $\boldsymbol{z}_k$ is a feature vector of platform k, describing which users will consider platform k as informative, and $\boldsymbol{u}_i^C$ is the concatenation of all the platform-specific representations of user u_i. An observation is that for users who have similar embeddings on each platform, they will have similar weights over each platform, which allows us to leverage the learned embeddings in each social platform.

To make the weight learning task-aware, the weights of social networks for each user are automatically learned with the back-propagation algorithm based on the predictive error in different tasks. Assume that the set of labeled users is $\mathcal{L}$. For user linkage prediction, $\mathcal{L}$ is the collection of linked user pairs between $\mathcal{G}$ and G^{K+1} and the loss function is defined as Equation 12, where $\boldsymbol{u}_i$ is the robust representation of user u_i from $\mathcal{G}$ and $\boldsymbol{u}_j$ is the view-specific representation of user u_j in G^{K+1}. For user attribute prediction, $\mathcal{L}$ is the set of users with labeled attribute values and the loss function is defined as Equation 13. In this equation, $\boldsymbol{u}_i$ is the robust representation for user u_i, and $\boldsymbol{l}_i$ is the attribute value label vector of u_i, in which the dimension j is set to 1 if u_i's attribute is j^{th} value and set as 0 otherwise, and $\boldsymbol{w}$ is the parameter set of the classifier.

$$O_{at}^{link} = -\sum_{(u_i, u_j) \in \mathcal{L}} [log(p(\boldsymbol{u}_i, \boldsymbol{u}_j)) + \sum_{(u_{i'_n}, u_{j'_n}) \notin \mathcal{L}}^{N} log(1 - p(\boldsymbol{u}_{i'_n}, \boldsymbol{u}_{j'_n}))] \tag{12}$$

$$O_{at}^{attr} = \sum_{u_i \in \mathcal{L}} \parallel \boldsymbol{w} \boldsymbol{u}_i - \boldsymbol{l}_i \parallel_2^2 \tag{13}$$

3.3 Model Optimization

Alternating optimization algorithm is adopted for the initialization of view-specific node representations on each social platform. Then, MV-URL optimizes the collaborations across different social platforms with a combination of the well known Mini-Batch Stochastic

Algorithm 1: Optimization Algorithm

Input: K social networks $G = (G^1, G^2, \ldots, G^K)$, a set of labeled data $\mathcal{L}$, number of samples M, number of negative samples N;
Output: Robust latent representations for all the users on G;

1 $k = 1$;
2 Initialize $\boldsymbol{C}$ randomly;
3 **while** $k <= K$ **do**
4 Initialize $\boldsymbol{U}^k$ and $\boldsymbol{V}^k$ randomly;
5 **while** *not converge* **do**
6 Fixing $\boldsymbol{V}^k$ and $\boldsymbol{C}$, find the optimal $\boldsymbol{U}^k$ with gradient descent w.r.t. Equ. 5;
7 Fixing $\boldsymbol{U}^k$ and $\boldsymbol{C}$, find the optimal $\boldsymbol{V}^k$ with gradient descent w.r.t. Equ. 5;
8 Fixing $\boldsymbol{U}^k$ and $\boldsymbol{V}^k$, find the optimal $\boldsymbol{C}$ with gradient descent w.r.t. Equ. 5;
9 **end**
10 **end**
11 **while** *not converge* **do**
12 $iter = 0$;
13 **while** $iter<M$ **do**
14 Randomly select a view, denoted as k;
15 Randomly select a graph from G^k;
16 Sample an edge with Equation 14 from the selected graph and sample N negative edges with bi-direction strategy;
17 Update $\boldsymbol{U}^k$, $\boldsymbol{V}^k$ or $\boldsymbol{C}$ w.r.t. Equ. 1;
18 $iter = iter + 1$;
19 **end**
20 Optimize the parameters of the softmax unit with gradient decent w.r.t. Equ. 12 or 13;
21 Update the weights of social platforms for each user according to Equ. 11;
22 Compute robust user representations across social platforms $\boldsymbol{U}$ according to Equ. 7;
23 **end**

Gradient Descent (SGD) [25] and the back-propagation algorithm. The overall optimization algorithm is summarized in Algorithm 1.

Minimizing the loss function for each social platform in Equation 5 is a convex optimization problem. Thus, we can employ alternating optimization algorithms for pre-training the node vectors (i.e., users, items and words) on each social platform (Lines from 1 to 10). This pre-training is for not propagating negative influences to other social platforms during the optimization for the collaborations across related social media.

After initialization of vectors on each social platform, we optimize the collaborations across platforms with a combination of Mini-Batch SGD and back-propagation algorithm (Lines from 11 to 23). For SGD (Lines from 12 to 19), we fix the weights of social platforms for each user and control the learning rate using *AdaGrad* [7]. Hierarchical sampling strategy is used to sample positive edges (Line 14 to 18). Specifically, we first sample a view randomly and then sample a positive edge (i, j) from this view according to the probability defined in Equation 14. For this positive edge, we set corresponding negative edges following existing work [25, 29, 30], and update the view-specific representations and shared word representations with respect to Equation 1. For the back-propagation algorithm (Line 20 to Line 22), by fixing the entities' representations on each social platform, we compute the parameter vectors of social platforms (i.e., $\boldsymbol{z}_k$) with the labeled data using back propagation algorithm (Line 20), and update the voting weights of social platforms (i.e., λ_u^k) for different users (Line 21). Finally, different view-specific user representations will be integrated to vote for the robust representations $\boldsymbol{U}$ based on the learned weights (Line 22).

$$p(i, j) = \frac{w_{ij}}{\sum_{(i', j') \in \mathcal{E}} w_{i'j'}} \tag{14}$$

3.4 Further Discussion

There are two major steps in Algorithm 1: independent training on each social platform and joint training across networks. The computation of first step is naturally parallelizable (i.e., from Line 3 to Line 10). For the second step, inspired by [7], asynchronous stochastic gradient descent (ASGD) can be used to parallelize the model optimization procedure on GPUs to accelerate the training process. This is based on the observation that social graphs are usually very sparse and when different threads sample different positive instances for model updating on a sparse dataset, the elements of the sampled instances in different threads seldom overlap, i.e., the latent vectors of the elements (e.g., users, items and words) usually do not conflict across different threads. It has been shown experimentally in [7] that, on social networks, the speed up ratio is nearly linear to the number of threads and there is almost no loss of performance in running models with ASGD. MV-URL is implemented with ASGD for the evaluation in this paper.

Studying MV-URL under streaming scenarios is interesting as it offers a solution to online embedding of *heterogeneous streams*. In streaming scenario, the optimization should be implemented without knowledge of entire graphs. For Algorithm 1, we discuss how to extend the part between Line 11 and Line 23 to streaming scenario, as the initialization of MV-URL between Line 1 and Line 10 is usually done offline. One change is that, batch-based stochastic gradient decent is used to update the weight of each social media from Line 20 to Line 22. For the collaboration embedding part (i.e., between Line 13 and Line 19), we discuss two possible scenarios. If the window size is large and there are data from various social platforms in a window, we still use the hierarchy sampling strategy in Algorithm 1 to update the embeddings online. This is to ensure that every view and every graph will get updated if the data distribution is unbalanced. If the window size is small and a window only contains the data from one single social platform, we adopt the existing edge-based sampling strategy for single graphs [11, 30].

	Property	Twitter	Foursquare
Node	Users	5,223	5,392
	Items	122,047	38,921
	Words	123,534	28,411
Link	User-User	164,920	76,972
	User-Item	615,515	48,756
	User-Word	12,209,481	292,561

Table 1: Statistics of Graphs on TW-FS

4 EVALUATION

The experiment is to evaluate the effectiveness of the latent features learned by MV-URL on two tasks: user account linkage across social platforms and user attribute prediction.

4.1 Experimental Setups

4.1.1 Datasets and Graph Construction. In this paper, two collections of real-world linked heterogeneous networks are used: TW-FS-FB and TW-FS. TW-FS has been widely used in previous research work [36, 38, 40] while TW-FS-FB is a new collection of three-linked heterogeneous networks. There are three real-world social networks involved in these two collections of datasets: Twitter (TW), Foursquare (FS) and Facebook (FB).

TW-FS-FB contains the related information of a group of users on these three social networks. There are 2,990 users in total on this dataset. 2,328 users have records on both Foursquare and Twitter and 1,755 of them have activities on all three platforms. TW-FS

	Property	Twitter	Foursquare	Facebook
Node	Users	2,502	2,990	2,041
	Items	14,517	39,364	3487
	Words	20,028	12,591	20,341
Link	User-User	61,991	180,247	14,347
	User-Item	48,103	39,364	80,613
	User-Word	767,994	194,768	1,812,013

Table 2: Statistics of Graphs on TW-FS-FB

contains the activities of another group of users on Foursquare and Twitter [38]. The number of shared users is 3, 388.

We construct User-User graph, User-Item graph and User-Word graph for each social network on both datasets. For Facebook, we extract "Likes" information for each user and the items in "Likes" are used to construct User-Item graph. For both Twitter and Foursquare, the check-in locations are used as the items in User-Item graph. Users' comments and post information on all three social platforms are used to extract the words in User-Word graph. More specifically, we first extract all the words based on the stop words listed in "NLTK 3.3" (Natural Language Toolkit)[1], which is a leading platform for building Python programs to work with human language data, then remove the words which have been used less than 20 times, and at last remove the words whose length is less than 3.

We assume that, for User-User graphs, all the edges are equally weighted. On Facebook, all edges in User-Item graph are equally weighted. On Twitter and Foursquare, the weight of e_{uv} in the User-Item graphs is computed as $1 + log(N_{uv})$, where N_{uv} is the number of check-ins associated with user u and item v. Similarly, the weight of an edge e_{uc} in User-Word graphs is computed as $1 + log(N_{uc})$, where N_{uc} is the numbers of activities associated with u and word c. Specifically, N_{uc} is the number of items that are linked with u in User-Item graph and have content containing the word c. The statistics of constructed graphs on the two collections of networks are presented in Table 1 and Table 2 respectively.

4.1.2 Comparison Methods. We compare MV-URL with the following four graph embedding methods:

LINE and **Node2Vec**: Both LINE [30] and Node2Vec [11] are designed for *single networks*. To feed our datasets to these two models, we merge the edges of different social networks into a unified network and then embed the unified network with these models. We follow the rules that the items are not shared while both users and content words are shared when do the merging for both LINE and Node2Vec.

MVE: MVE [25] treats each social network as a single graph and uses a single-view based embedding method to encode the nodes on each social network. Thus, we first merge the *User-User Graph*, *User-Item Graph* and *User-Word Graph* on each social network into one single graph. As a result, we have three merged graphs on TW-FS-FB dataset and two merged graphs on TW-FS dataset. Then MVE is used to embed the merged graphs by treating each merged graph on a social network as a single view.

EOE: According to the defined input of EOE [33], we first merge the *User-User Graphs*, *User-Item Graphs* and *User-Word Graphs* on different social networks into a unified *User-User Graph*, *User-Item Graph* and *User-Word Graph* respectively. As a result, we have three merged graphs for both TW-FS-FB dataset and TW-FS dataset as the input to EOE.

As we argued before, our model is able to model different levels of heterogeneity. To experimentally validate the benefits brought by restraining shared semantic space across different social networks, using a diagonal harmonious embedding matrix to reconcile the heterogeneities inside single networks and introducing co-regulation scheme to promote the collaboration of different networks, respectively, we also compare with three variant versions of our MV-URL model: MV-URL-V1, MV-URL-V2 and MV-URL-V3.

MV-URL-V1 is the variant version of MV-URL model assuming that different social networks have their own content space. More specifically, this model assumes that the embeddings for the same words on different social networks are different and uses $\boldsymbol{c}_j^k$ instead of $\boldsymbol{c}_j$ to represent the word embeddings in Equation 5.

MV-URL-V2 is the variant version of MV-URL model which does not use the diagonal harmonious embedding matrix inside single networks. Specifically, this model uses the following Equation instead of Equation 3 as the measurement of the closeness between vertexes of User-Item graph.

$$p(\boldsymbol{u}_i, \boldsymbol{v}_j) = \frac{1}{1 + exp\{-\boldsymbol{u}_i^T \boldsymbol{v}_j\}}$$

MV-URL-V3 is the variant version of MV-URL model which omits the co-regulation scheme and trains the representations of each social network independently. MV-URL-V3 uses the *average* of a user's representations on multiple social networks as his/her personal representation. More specifically, we use the following Equation instead of Equation 7 to compute robust representations based on the results on multiple social networks.

$$\boldsymbol{u} = \sum_{k \in \mathcal{S}_u} \frac{1}{|\mathcal{S}_u|} \boldsymbol{u}^k$$

4.1.3 Evaluation Method. In this part, we will introduce the evaluation methods for both tasks.

User Linkage Prediction. As described in Section 3.2, MV-URL predicts the user linkage between $\mathcal{G} = \{G^1, G^2, \ldots, G^K\}$ and G^{K+1}, where $K \geq 2$ as there is no meaning in learning weight for one graph (i.e., $K = 1$), thus we need at least three datasets for this task. So we only conduct the evaluation of this task on TW-FS-FB. We evaluate three scenarios where $\mathcal{G}$ is set to $\{TW, FS\}$, $\{TW, FB\}$ and $\{FS, FB\}$ respectively. For the shared users between $\mathcal{G}$ and G^{K+1}, we randomly sample 50% users as the labeled data $\mathcal{X}_{train}$ and use the other 50% users as the test data $\mathcal{X}_{test}$.

There are many linked user pairs (u_i, u_j) in $\mathcal{X}_{test}$ where $u_i \in \mathcal{G}$ and $u_j \in G^{K+1}$. For each $u_i \in \mathcal{G}$ in the test set, we want to predict his/her linked user in G^{K+1} by ranking the users in G^{K+1} according to the scores defined in Equation 2. Let $rank(u_j)$ denote the position of u_j in the ranking list. To evaluate the user linkage prediction thoroughly, we employ two widely-used metrics $Hits@n$ and *MRR* (i.e., Mean Reciprocal Rank) [3, 20] in top-n prediction tasks. While $Hits@n$ indicates the probability that the models put the actually linked users in the top-n predictions, *MRR* measures the ranking quality which assigns higher scores to the test cases where the ground-truth users are at higher ranking positions.

The computation of $Hits@n$ proceeds as follows. We form a top-n prediction list by picking the n top ranked users from the ranked list. If $rank(u_j) \leq n$, we have a hit. Otherwise, we have a

[1]https://www.nltk.org/

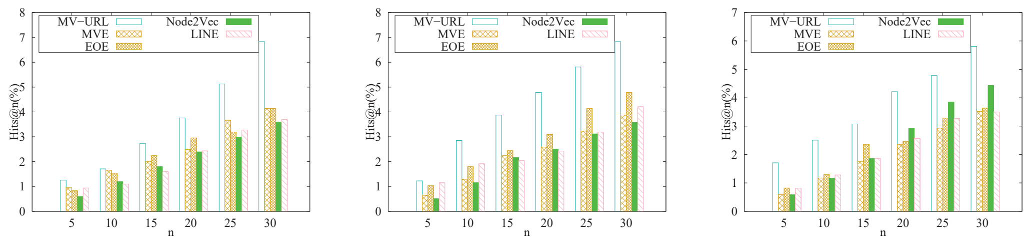
(a) MacroF1

Methods \ Attributes	Hometown	UserLocation
MV-URL	**0.333971**	**0.326309**
MVE	0.305339	0.306698
EOE	0.210579	0.203255
Node2Vec	0.094384	0.091984
LINE	0.031329	0.019743

(b) MicroF1

Methods \ Attributes	Hometown	UserLocation
MV-URL	**0.451953**	**0.418762**
MVE	0.375307	0.359624
EOE	0.277918	0.265274
Node2Vec	0.112893	0.094906
LINE	0.130575	0.095707

Table 3: User Attribute Prediction Results



(a) Foursquare (b) Twitter (c) Facebook

Figure 2: *Hits@n* of User Account Linkage Prediction

miss. We define *hit@n* for a single test case as either 1, if we have a hit for this case, or 0, if otherwise. The overall *Hits@n* is defined by averaging over all test cases:

$$Hits@n = \frac{\#hit@n}{|\mathcal{X}_{test}|} \tag{15}$$

where $\#hit@n$ denotes the number of hits in the test set, and $|\mathcal{X}_{test}|$ is the number of test cases.

$$MRR = \frac{1}{|\mathcal{X}_{test}|} \sum_{(u_i,u_j)\in\mathcal{X}_{test}} \frac{1}{rank(u_j)} \tag{16}$$

MRR is defined as Equation 16. It is an average of the reciprocal rank of a positive example, and a good prediction model should have a bigger *MRR* value. In contrast to mean rank, *MRR* is less sensitive to outliers.

User Attribute Prediction. We evaluate this task on TW-FS as detailed attribute information has been provided on this dataset. More specifically, we evaluate the prediction of the attribute *Hometown* on Foursquare and *UserLocation* on Twitter. We choose these two attributes for two reasons. First, many real applications can benefit from predicting these attributes. For example, we can recommend points of interests (e.g., restaurants, hotels, etc) to a specific user if we know his/her current location. Second, these two attributes have reasonable granularities (i.e., not too coarse like *Gender* which makes the prediction trivial, and not too fine like *UserName* which makes the prediction hard and meaningless). There are 121 and 161 classes for *Hometown* and *UserLocation* respectively.

The prediction of one attribute is actually a multi-label classification task. We use the user representations learned by different algorithms as input features, and train one-vs-rest linear classifiers using the LibLinear package [8]. 50% labeled data are used to learn the weights of different social platforms. The results are evaluated with a pair of widely used metrics in multi-class classification tasks: MacroF1 and MicroF1 [18, 25]. We will not report the details of these two metrics considering the limited space.

Hyper-parameter Optimization. For the hyper parameters in MV-URL, we perform cross-validation. Specifically, we use the grid search algorithm to obtain the optimal hyper parameter setup on the validation dataset: α, β, γ, δ, and ϵ. We will selectively present the parameter sensitivity analysis for some important hyper-parameters.

Methods \ Test Sets	Foursquare	Twitter	Facebook
MV-URL	0.014417	0.014387	0.016667
MVE	0.010999	0.009340	0.008564
EOE	0.010959	0.011901	0.011253
Node2Vec	0.008727	0.008936	0.009178
LINE	0.007544	0.011578	0.009737

Table 4: *MRR* of User Account Linkage Prediction

4.2 Prediction Effectiveness

In this part, we present the overall performance of all embedding methods with well-tuned parameters. For all approaches, the dimension of node representation is set as 100. For LINE, EOE, MVE and MV-URL, the number of negative samples N is set to 5, and the initial learning rate is set to 0.025, as suggested in [25, 30]. For node2vec, the window size is set to 10, the walk length is set as 40, parameters p and q are selected based on the labeled data, following [11]. For MVE, parameter η is set to 0.05, as suggested in [25].

Table 3 demonstrate the user attribute prediction results on TW-FS. Some observations can be made from Table 3: 1) MV-URL *outperforms all the other models* consistently on this task. More specifically, compared with the second best model MVE, MV-URL brings 9.38% (MacroF1) and 20.42% (MicroF1) improvements for the prediction of users' hometown, and also brings 6.39% (MacroF1) and 16.44% (MicroF1) improvements for the prediction of users' location; 2) MVE performs the second best consistently for this task. The possible reason is that both MV-URL and MVE are supervised learning methods and they are capable in taking advantage of the user attribute information other than the graph information in this task while the other methods are not.

For user linkage prediction task, we present both *Hits@n* where $n = 5, 10, 15, 20, 25, 30$ and *MRR* when the social network to be linked (i.e., denoted as G^{K+1}) is set to Facebook, Foursquare and

(a) MacroF1

Methods \ Attributes	Hometown	UserLocation
MV-URL	0.333971	0.326309
MV-URL-V1	0.325669	0.290353
MV-URL-V2	0.326109	0.287120
MV-URL-V3	0.293468	0.291086

(b) MicroF1

Methods \ Attributes	Hometown	UserLocation
MV-URL	0.451953	0.418762
MV-URL-V1	0.438136	0.396818
MV-URL-V2	0.443412	0.406480
MV-URL-V3	0.410801	0.381989

Table 5: Impact of Factors in User Attribute Prediction

Methods \ Test Sets	Foursquare	Twitter	Facebook
MV-URL	0.014417	0.014387	0.016667
MV-URL-V1	0.012885	0.013774	0.015611
MV-URL-V2	0.013637	0.012391	0.012149
MV-URL-V3	0.009305	0.010630	0.008452

Table 6: Impact of Factors in User Account Linkage Prediction Measured by *MRR*

Twitter respectively. Figure 2 show methods' performance measured by *Hits@n*, while Table 4 show the results of *MRR*. From Figure 2 and Table 4, we can see that the improvements brought by MV-URL is significant compared with the comparison methods. For *Hits@n*, we take top-20 prediction as an example. Compared with the second performers, MV-URL has brought 27.19%, 54.27%, and 44.63% improvements on Foursquare, Twitter and Facebook, respectively. This indicates that, the proposed MV-URL has much larger probability in putting the actually linked users across social media in the top-20 predictions. From Table 4, we can see that, compared with the second best methods, MV-URL brings 31.08%, 20.89% and 48.11% improvements on Foursquare, Twitter and Facebook, respectively. This demonstrates that MV-URL prefers to rank the ground-truth user in the top positions. Note that, different from the user attribute prediction task where MV-URL and MVE are capable in using user attribute information in addition to graph information while the other methods are not, all the methods have used the same amount of information in user linkage prediction.

4.3 Impact of Different Factors

Analysis of Various Heterogeneity Reconciling Strategies. To evaluate key components of MV-URL, we implement and compare with three variant versions: MV-URL-V1, MV-URL-V2 and MV-URL-V3. The results are demonstrated in Table 5, 6 and Figure 3.

Several observations can be made from the results in Table 5, 6 and Figure 3: 1) MV-URL outperforms MV-URL-V1, MV-URL-V2 and MV-URL-V3 on all tasks, which evaluate the benefit brought by sharing the same semantic space, using the diagonal harmonious embedding matrix to reconcile the heterogeneities inside of single networks and adopting co-regulation scheme in promoting the collaboration of different networks, respectively; 2) MV-URL-V3 performs worst compared with the other two variant versions. Essentially, the possible reason is that MV-URL-V3 is a late fusion method, where the user linkage information is used in fusing the representation learned from different social networks while the other methods use this linkage information in training the representations. This performance discrepancy shows that in our problem setting, user representation learning usually benefits more from the early fusion methods compared to late fusion methods.

Sensitivity Analysis. In this experiment, we investigate the sensitivity of MV-URL w.r.t. two important hyper-parameters: the dimension of node representations D and the number of negative samples for each positive instance N. Considering the limited space, for user account linkage prediction, the sensitivity analysis is done only when the target is Facebook.

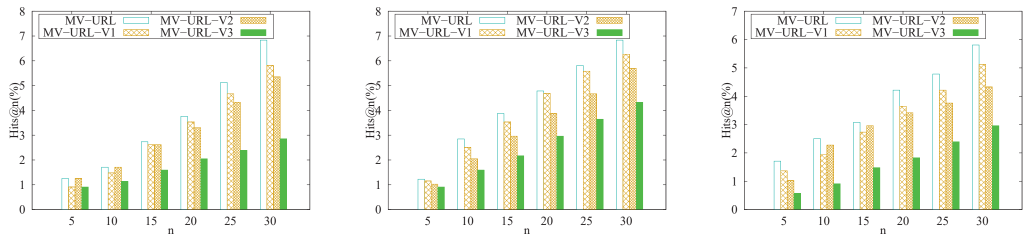
Figure 4 contain the impact of varying D on user attribute prediction and user linkage prediction. We observe that the performance in both tasks first improves by increasing D and then the increment becomes small. The reason is that D represents the model complexity. Thus, when D is too small, the model has limited ability to describe the data. However, when D exceeds a threshold (e.g., D = 60 for user linkage task measured with MRR), the model is complex enough to handle the data. At this point, it is less helpful to improve the model performance by increasing D.

Figure 5 show how MV-URL performs with various N on user attribute prediction and user linkage prediction. This figure shows MV-URL gains a significant improvement by increasing N from 1 to 2 for all the tasks. Then the model's performance becomes steady. This demonstrates that our proposed model does not require too many negative samples to reach a good performance (i.e., MV-URL is able to achieve a very good performance with $N = 2$). A smaller number of negative samples means much less training time. For example, assuming that the training time of the model with 5 negative samples is t, the training time is roughly $t/2$ for the model with 2 negative samples according to our experiments.

5 RELATED WORK

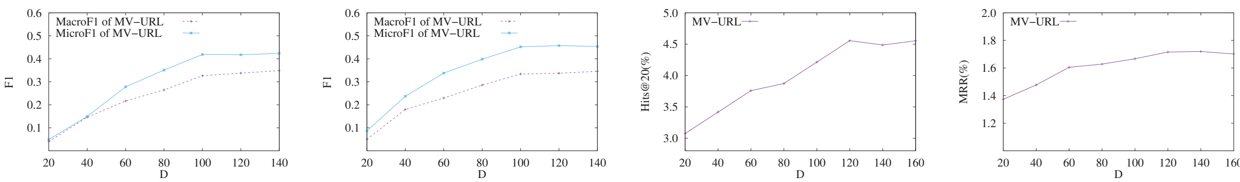
There are two lines of work that are most related to our work in this paper: user modeling on social media and network embedding.

User Modeling on Social Media. User modeling on social media has drawn much research interest, ranging from link prediction containing user linkage prediction [5, 6, 19] and friendship prediction [13, 15, 36, 40], user attribute prediction [2, 23, 24], community detection [14, 35, 37] to item recommendation [21, 36, 42]. Our work focus on two application tasks: user account linkage prediction and user attribute prediction. The essential idea of user account linkage prediction is building users' profiles with the information on social media and then computing the similarities between users based on the built users' profiles. The users with highest similarities on two platforms are identified as the same user. Most existing work in this area focus on pair-wise user linkage between two platforms only [5, 12, 17, 19] while our work focus on user account linkage across multiple platforms. User attribute prediction is actually a classification task and most existing work only leverage the information on one social platform to train a classification model [2, 23, 24]. Our research work is able to leverage information from multiple social platforms in predicting the users' attributes.



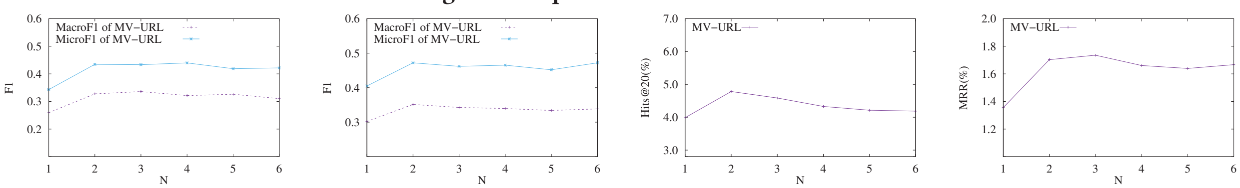
(a) Foursquare (b) Twitter (c) Facebook

Figure 3: Impact of Factors in User Account Linkage Prediction Measured by $Hits@n$



(a) User Location Prediction (b) User Hometown Prediction (c) User Linkage ($Hits@20$) (d) User Linkage (MRR)

Figure 4: Impact of Latent Dimensions



(a) User Location Prediction (b) User Hometown Prediction (c) User Linkage ($Hits@20$) (d) User Linkage (MRR)

Figure 5: Impact of Negative Samples

Network Embedding. Network embedding has become a very hot research problem in recent years and has been proven fundamental for many network based applications [10, 26, 28]. The aim is to project a graph-structured data into the feature vector representations automatically. With these feature vector representations, many machine learning algorithms can be applied.

One major line of research work treat the relations as the translations between the entities [1, 16]. In recent years, some embedding works have been proposed based on random walk [11] and deep learning models [22], which extend the techniques in language modeling and deep learning from sequences of words to graphs by treating walks as the equivalent of sentences.

Most the above techniques are based on single networks. However, many real systems can be naturally projected to multiple linked networks [26], like movie knowledge library entries can be found from both IMDB[2] and Douban Movie sites[3]. Broad Learning [37, 39] is a new type of learning task, which focuses on fusing multiple information sources together and carrying out data mining tasks across these fused sources in one unified analytic. The current research work in Broad Learning can be categorized into two major types: learning over multiple networks with same entities but different relations (e.g., multi-view learning, multi-source learning and multi-model learning) [9, 25], and transferring the information from other networks to improve the learning on one specific network [38]. Different from these existing work, our work focuses on joint representation learning by integrating multiple heterogeneous networks where each network involved is a heterogeneous network containing different links and diverse nodes.

[2] http://www.imdb.com/
[3] https://www.douban.com/

6 CONCLUSION

In this paper, we proposed a novel user representation learning model, MV-URL, to jointly and automatically embed users over multiple linked heterogeneous social networks, which effectively overcomes the challenges arising from modeling heterogeneous information sources. MV-URL assumes that all the social networks share the same semantic space by restraining the content words share the same latent representation across different social networks. To reconcile the heterogeneities of different latent spaces inside of single networks, MV-URL uses a diagonal harmonious embedding matrix to project the embeddings from one latent space to another latent space. On the other hand, a co-regulation scheme is introduced to promote the collaborations of different networks. Extensive experiments have been conducted on two collections of real world linked networks. The evaluation results show that MV-URL outperforms both state-of-the-art approaches for user embedding with individual networks and other competitive approaches with multiple networks.

ACKNOWLEDGMENTS

This work was supported by ARC (Australian Research Council) Discovery Project (No. DP190101985 and DP170103954), ARC Discovery Early Career Researcher Award (No. DE160100308), and Shanxi Provincial Natural Science Foundation in China (No. 2018JM6063).

REFERENCES

[1] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *NIPS*. 2787–2795.

[2] Nina Cesare, Christan Grant, and Elaine O. Nsoesie. 2017. Detection of User Demographics on Social Media: A Review of Methods and Recommendations for Best Practices. *CoRR* abs/1702.01807 (2017).

[3] Hongxu Chen, Hongzhi Yin, Weiqing Wang, Hao Wang, Quoc Viet Hung Nguyen, and Xue Li. 2018. PME: Projected Metric Embedding on Heterogeneous Networks for Link Prediction. In *SIGKDD*. 1177–1186.

[4] Ting Chen and Yizhou Sun. 2017. Task-Guided and Path-Augmented Heterogeneous Network Embedding for Author Identification. In *WSDM*. 295–304.

[5] Wei Chen, Hongzhi Yin, Weiqing Wang, Lei Zhao, Wen Hua, and Xiaofang Zhou. 2017. Exploiting Spatio-Temporal User Behaviors for User Linkage. In *CIKM*. 517–526.

[6] Wei Chen, Hongzhi Yin, Weiqing Wang, Lei Zhao, and Xiaofang Zhou. 2018. Effective and Efficient User Account Linkage across Location Based Social Networks. In *ICDE*. 1085–1096.

[7] John C. Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research* 12 (2011), 2121–2159.

[8] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research* 9 (2008), 1871–1874.

[9] Jing Gao, Jiawei Han, Jialu Liu, and Chi Wang. 2013. Multi-View Clustering via Joint Nonnegative Matrix Factorization. In *SIAM*. 252–260.

[10] Ming Gao, Leihui Chen, Xiangnan He, and Aoying Zhou. 2018. BiNE: Bipartite Network Embedding. In *SIGIR*. 715–724.

[11] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable Feature Learning for Networks. In *SIGKDD*. 855–864.

[12] Xiangnan Kong, Jiawei Zhang, and Philip S. Yu. 2013. Inferring anchor links across multiple heterogeneous social networks. In *CIKM*. 179–188.

[13] Vincent Leroy, Berkant Barla Cambazoglu, and Francesco Bonchi. 2010. Cold start link prediction. In *SIGKDD*. 393–402.

[14] Jure Leskovec, Kevin J. Lang, and Michael W. Mahoney. 2010. Empirical comparison of algorithms for network community detection. In *WWW*. 631–640.

[15] Ryan Lichtenwalter, Jake T. Lussier, and Nitesh V. Chawla. 2010. New perspectives and methods in link prediction. In *SIGKDD*. 243–252.

[16] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In *AAAI*. 2181–2187.

[17] Siyuan Liu, Shuhui Wang, Feida Zhu, Jinbo Zhang, and Ramayya Krishnan. 2014. HYDRA: large-scale social identity linkage via heterogeneous behavior modeling. In *SIGMOD*. 51–62.

[18] Mohamed Morchid, Mohamed Bouallegue, Richard Dufour, Georges Linarès, Driss Matrouf, and Renato De Mori. 2015. Compact Multiview Representation of Documents Based on the Total Variability Space. *IEEE/ACM Trans. Audio, Speech & Language Processing* 23, 8 (2015), 1295–1308.

[19] Xin Mu, Feida Zhu, Ee-Peng Lim, Jing Xiao, Jianzong Wang, and Zhi-Hua Zhou. 2016. User Identity Linkage by Latent User Space Modelling. In *SIGKDD*. 1775–1784.

[20] Maximilian Nickel, Lorenzo Rosasco, and Tomaso A. Poggio. 2016. Holographic Embeddings of Knowledge Graphs. In *AAAI*. 1955–1961.

[21] Wei Pan, Nadav Aharony, and Alex Pentland. 2011. Composite Social Network for Predicting Mobile Apps Installation. In *AAAI*.

[22] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: online learning of social representations. In *SIGKDD*. 701–710.

[23] Daniel Preotiuc-Pietro, Wei Xu, and Lyle H. Ungar. 2016. Discovering User Attribute Stylistic Differences via Paraphrasing. In *AAAI*. 3030–3037.

[24] Minghui Qiu, Yanchuan Sim, Noah A. Smith, and Jing Jiang. 2015. Modeling User Arguments, Interactions, and Attributes for Stance Prediction in Online Debate Forums. In *SIAM*. 855–863.

[25] Meng Qu, Jian Tang, Jingbo Shang, Xiang Ren, Ming Zhang, and Jiawei Han. 2017. An Attention-based Collaboration Framework for Multi-View Network Representation Learning. In *CIKM*. 1767–1776.

[26] Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun, and Philip S. Yu. 2017. A Survey of Heterogeneous Information Network Analysis. *TKDE* 29, 1 (2017), 17–37.

[27] Ajit Paul Singh and Geoffrey J. Gordon. 2008. Relational learning via collective matrix factorization. In *SIGKDD*. 650–658.

[28] Yizhou Sun and Jiawei Han. 2012. Mining heterogeneous information networks: a structural analysis approach. *SIGKDD Explorations* 14, 2 (2012), 20–28.

[29] Jian Tang, Meng Qu, and Qiaozhu Mei. 2015. PTE: Predictive Text Embedding through Large-scale Heterogeneous Text Networks. In *SIGKDD*. 1165–1174.

[30] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. LINE: Large-scale Information Network Embedding. In *WWW*. 1067–1077.

[31] Lei Tang and Huan Liu. 2011. Leveraging social media networks for classification. *Data Min. Knowl. Discov.* 23, 3 (2011), 447–478.

[32] Weiqing Wang, Hongzhi Yin, Ling Chen, Yizhou Sun, Shazia Wasim Sadiq, and Xiaofang Zhou. 2015. Geo-SAGE: A Geographical Sparse Additive Generative Model for Spatial Item Recommendation. In *SIGKDD*. 1255–1264.

[33] Linchuan Xu, Xiaokai Wei, Jiannong Cao, and Philip S. Yu. 2017. Embedding of Embedding (EOE): Joint Embedding for Coupled Heterogeneous Networks. In *WSDM*. 741–749.

[34] Hongzhi Yin, Bin Cui, Xiaofang Zhou, Weiqing Wang, Zi Huang, and Shazia W. Sadiq. 2016. Joint Modeling of User Check-in Behaviors for Real-time Point-of-Interest Recommendation. *TOIS* 35, 2 (2016), 11:1–11:44.

[35] Hongzhi Yin, Zhiting Hu, Xiaofang Zhou, Hao Wang, Kai Zheng, Nguyen Quoc Viet Hung, and Shazia Wasim Sadiq. 2016. Discovering interpretable geo-social communities for user behavior prediction. In *ICDE*. 942–953.

[36] Jiawei Zhang, Jianhui Chen, Shi Zhi, Yi Chang, Philip S. Yu, and Jiawei Han. 2017. Link Prediction across Aligned Networks with Sparse and Low Rank Matrix Estimation. In *ICDE*. 971–982.

[37] Jiawei Zhang, Limeng Cui, Philip S. Yu, and Yuanhua Lv. 2017. BL-ECD: Broad Learning based Enterprise Community Detection via Hierarchical Structure Fusion. In *CIKM*. 859–868.

[38] Jiawei Zhang, Xiangnan Kong, and Philip S. Yu. 2014. Transferring heterogeneous links across location-based social networks. In *WSDM*. 303–312.

[39] Jiawei Zhang, Congying Xia, Chenwei Zhang, Limeng Cui, Yanjie Fu, and Philip S. Yu. 2017. BL-MNE: Emerging Heterogeneous Social Network Embedding Through Broad Learning with Aligned Autoencoder. In *ICDM*. 605–614.

[40] Jiawei Zhang, Philip S. Yu, and Zhi-Hua Zhou. 2014. Meta-path based multi-network collective link prediction. In *SIGKDD*. 1286–1295.

[41] Erheng Zhong, Wei Fan, Junwei Wang, Lei Xiao, and Yong Li. 2012. ComSoc: adaptive transfer of user behaviors over composite social network. In *SIGKDD*. 696–704.

[42] Junxing Zhu, Jiawei Zhang, Lifang He, Quanyuan Wu, Bin Zhou, Chenwei Zhang, and Philip S. Yu. 2017. Broad Learning based Multi-Source Collaborative Recommendation. In *CIKM*. 1409–1418.