Embedding of Embedding (EOE) : Joint Embedding for Coupled Heterogeneous Networks

Linchuan Xu
The Hong Kong Polytechnic University
Kowloon City, Hong Kong
cslcxu@comp.polyu.edu.hk

Xiaokai Wei
University of Illinois at Chicago
Chicago, IL, USA
weixiaokai@gmail.com

Jiannong Cao
The Hong Kong Polytechnic University
Kowloon City, Hong Kong
csjcao@comp.polyu.edu.hk

Philip S. Yu
University of Illinois at Chicago
Chicago, IL, USA
psyu@uic.edu

ABSTRACT

Network embedding is increasingly employed to assist network analysis as it is effective to learn latent features that encode linkage information. Various network embedding methods have been proposed, but they are only designed for a single network scenario. In the era of big data, different types of related information can be fused together to form a coupled heterogeneous network, which consists of two different but related sub-networks connected by inter-network edges. In this scenario, the inter-network edges can act as complementary information in the presence of intra-network ones. This complementary information is important because it can make latent features more comprehensive and accurate. And it is more important when the intra-network edges are absent, which can be referred to as the cold-start problem. In this paper, we thus propose a method named embedding of embedding (EOE) for coupled heterogeneous networks. In the EOE, latent features encode not only intra-network edges, but also inter-network ones. To tackle the challenge of heterogeneities of two networks, the EOE incorporates a harmonious embedding matrix to further embed the embeddings that only encode intra-network edges. Empirical experiments on a variety of real-world datasets demonstrate the EOE outperforms consistently single network embedding methods in applications including visualization, link prediction multi-class classification, and multi-label classification.

CCS Concepts

•**Information systems** → **Data mining;**

Keywords

Network Embedding; Coupled Heterogeneous Networks; Data Mining

WSDM 2017, February 06-10, 2017, Cambridge, United Kingdom

DOI: http://dx.doi.org/10.1145/3018661.3018723

1. INTRODUCTION

Various explicit and implicit interactions, such as friendship, co-authorship and co-concurrence, between data points make the network ubiquitous. As a result, the network representation is inevitable in many data mining applications. On the one hand, many data mining applications are designed for networks, such as community detection [24, 9] and link prediction [12]. On the other hand, other data mining applications can benefit from the analysis of networks, such as collective classification [16] and dimension reduction [25, 26, 11]. All of above applications rely on the analysis of network interactions or edges. Nowadays, network embedding is increasingly employed to assist network analysis as it is effective to learn latent features that encode linkage information [19, 20, 1, 13, 17]. The basic idea of network embedding is to preserve the network structure by presenting pairs of vertices with edges to be close in the latent space. The latent features are beneficial as they are more expressive than edges and can be directly employed by off-the-shelf machine learning techniques. Although various network embedding methods [13, 17] have been proposed before, they are designed only for learning representations on a single network.

Furthermore, in the era of big data, different types of related information are often available and can be fused together to form a coupled heterogeneous network, where each type of information can be represented as a separate homogeneous network. We define a coupled heterogeneous network as a network consisting of two different but related sub-networks connected by inter-network links, such as (1) *author* and *word* networks (authors can be linked by interactions between authors, such as co-authorship, words can be linked by co-concurrence relationships, and authors can be linked to words they use in their papers); (2) social network *user* and *word* networks (links can be similar to those in author and word networks); (3) *customer* and *movie* networks (links between movies can result from having common actors or directors); (4) *gene* and *chemical compound* networks [5] (genes can be linked by gene-gene interaction, chemical compounds can be linked by having the same ontology, and genes can be linked to chemical compounds through binding relationships).

To visually illustrate this concept, an example of author

Figure 1: A co-authorship network and a word co-concurrence network, where black straight lines between authors and words are the edges connecting them. The edges among authors and among words are left out for the sake of clear appearance. The size of a vertex is proportional to its degree.

and word network is presented in Fig. 1, where authors are linked by co-authorships, and the concurrent appearance in the same title is defined as the co-concurrence relationship between words. We sample the network from a DBLP dataset [18]. The sampled co-authorship network consists of authors who have published papers in two data mining conferences, KDD and ICDM, and two database conferences, SIGMOD and VLDB from 2000 to 2003. It is shown that authors form two clusters, and so do words, which can be generated by community detection methods. Moreover, experts of data mining have more edges to words from data mining domain than those from database domain, and the case with experts of database is similar.

To learn latent features for authors, it is expected that authors of the same domain should be close to each other in the latent space. One can see that the edges between authors and words can act as complementary information in the presence of the author edges. This is because authors of the same domain are more likely to have edges to words of their domain, which can be utilized to make latent features which are learned from only author edges more comprehensive and accurate. And the complementary information is more important when the author edges are absent, which can be referred to as the cold-start problem. The case with learning latent features for words is similar. There are embedding methods available for either the author network or the word network. However, it is not straightforward to extend from existing embedding methods for a single network to one for coupled heterogeneous networks.

The major challenge is imposed by heterogeneous characteristics of two different networks, which would result in two heterogeneous latent spaces. As a result, latent features of different networks cannot be directly matched. To tackle this challenge, we propose a method named embedding of embedding (EOE) to further embed the embeddings from one latent space to the other latent space by introducing a harmonious embedding matrix. Specifically, the proposed EOE can transform the latent features from one space into another space by multiplying the appropriately designed harmonious embedding matrix. With this harmonious embedding matrix, there are no barriers for computations between latent features of different networks. As an embedding method, the proposed EOE also presents vertices connected by edges to be close in the latent space. The key difference from existing embedding methods for a single network is that there are three kind of edges and two type of latent spaces corresponding to two networks. Moreover, the latent features of both networks have to be leaned simultaneously as either side can provide complementary information to the other side through the inter-network edges.

It is directly followed that there are three types of variables to be optimized in the learning objective of the EOE, which are two types of latent features corresponding to two networks, and the embedding matrix. We thus propose an alternating optimization algorithm in which the learning objective is optimized with respect to one type of variable at a time until convergence. This alternating optimization algorithm can replace the difficult joint optimization over the three variables with a sequence of easier optimizations [4]. The EOE model and the optimization algorithm are presented in great details in the following sections.

The contributions of this paper are summarized as follows:

1. To the best of our knowledge, we are the first to investigate the problem of joint embedding for coupled networks. We propose a joint embedding model, EOE, which incorporates a harmonious embedding matrix to further embed the embeddings that only encode intra-network edges.

2. We propose an alternating optimization algorithm to solve the learning objective of the EOE in which the learning objective is optimized with respect to one type of variable at a time until convergence.

3. We conduct comprehensive empirical evaluation on a variety of real-world coupled heterogeneous networks to demonstrate the advantages of joint learning on coupled networks. The proposed EOE outperforms the baselines in four applications including visualization, link prediction, multi-class classification, and multi-label classification.

4. This work shows the coupled heterogeneous network as an effective representation of fusing information from different information sources, where each information source is represented as a separate homogeneous network and the inter-network link captures the relationship between the heterogeneous network nodes.

The rest of the paper is organized as follows. Section 2 presents the related work. The problem description, the model formulation, and the optimization algorithm are presented in Section 3, 4, 5, respectively. Section 6 presents comprehensive empirical evaluation. In section 7, we conclude and introduce our future work.

2. RELATED WORK

The proposed EOE model is related to general graph embedding or network embedding methods to learn latent representations for graph or network vertices. A couple of graph or network embedding methods have been proposed previously [15, 21, 8, 3], but they are originally designed for dimension reduction of existing features. Specifically, their objectives are to learn low-dimensional latent representations of existing features so that learning complexity brought by feature dimension would be significantly reduced. In our scenarios, there exist no features of network vertices but the network edge information.

Another graph embedding method called graph factorization [1] learns latent features by utilizing network edges. It presents graphs as matrices where matrix elements correspond to edges between vertices, and then learns latent features by matrix factorization. But the method graph factorization only presents pairs of vertices with interactions to be close in the latent space. The proposed EOE model not only presents vertices with edges to be close, but also presents vertices without edges to be away from each other. The latter regulation is important because it would preserve the information that certain vertices are not likely to interact, which is a part of network structure information as well.

A recent network embedding model called DeepWalk [13] embeds local link information obtained from random walks. It is motivated by the connection between degree distribution of networks and word frequency distribution of natural languages. Based on this observation, the model for natural languages is re-purposed to model network community structures. However, random walks may cross multiple communities, which is not desired for the purpose of network structure preserving. Moreover, DeepWalk can only handle unweighted networks, while the proposed EOE model is applicable to both weighted and unweighted networks.

The state-of-the-art related model is LINE [17] for large-scale information network embedding. LINE preserves both interaction information and non-interaction information, which is similar to the proposed EOE. But the proposed EOE model differs from LINE in the formulation of cost function. Moreover, the proposed EOE is designed for embedding of couple heterogeneous networks. None of existing methods including LINE can handle coupled heterogeneous networks, which are common in real world and beneficial to embeddings of each of the coupled networks.

3. PRELIMINARIES

We formally define the concept of the coupled heterogeneous network as follows:

DEFINITION 1. A **coupled heterogeneous network** is comprised of two different but related sub-networks that are connected by inter-network edges. The term "different" means that vertices of two sub-networks are of different types. And the terms "related" means that vertices of two sub-networks have a particular type of interaction or relationship. With a sub-network defined as $G_u(U, E_u, W_u)$, and another as $G_v(V, E_v, W_v)$, the coupled heterogeneous network is denoted as $G_{uv}(G_u, E_{uv}, W_{uv}, G_v)$, where U and V are set of vertices, E_u and E_v are sets of edges within G_u and G_v, respectively, E_{uv} is the set of edges connecting vertices of G_u and those of G_v, and W with different subscriptions are corresponding sets of edge weights. All the edges can be weighted, unweighted, directed, and undirected.

To learn latent features for U and V, the EOE utilizes network edges as an embedding method. Specifically, pairs of vertices with edges are presented to be close in the latent space. The closeness of two vertices is defined as follows:

DEFINITION 2. A pair of vertices is **close** in a certain latent space if the probability that there exists an edge between them is considerably high, ideally 100%. The probability is quantified by the following equation:

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

where $\boldsymbol{u}_i$ and $\boldsymbol{u}_j$ are column vectors of embeddings for vertices indexed by i and j in the network G_u, respectively.

The Eq. (1) is measured in the latent space for G_u, and the formulation for probability measured in the latent space for G_v is the same. However, the formulation for probability of existence of edges between vertices from G_u and those from G_v cannot be the same as it involves two different latent spaces. To reconcile the heterogeneities of the two latent spaces, we introduce a harmonious embedding matrix to further embed the embeddings from one latent space to another latent space, which is defined as follows:

DEFINITION 3. A **harmonious embedding matrix** is a $d_u \times d_v$ real-valued matrix M, where d_u and d_v are the dimensions of latent features of U and V, respectively.

With the harmonious embedding matrix M, the probability for measuring the closeness between vertices of different networks can be quantified as follows:

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

4. THE EOE MODEL

Based on the preliminaries, we now present the proposed EOE model. The EOE model not only presents vertices with edges to be close, but also presents vertices without edges to be away from each other. The latter regulation is important because it would preserve the information that certain vertices are not likely to interact, which is a part of network structure information as well. To cast both these two regulations to an optimization problem, small probabilities of pairs of vertices with edges and large probabilities of pairs of vertices without edges should be penalized. The loss function to penalize the former is formulated as follows:

$$L(U) = \sum_{(i,j) \in E_u} (w_u)_{ij} \times f(p(\boldsymbol{u}_i, \boldsymbol{u}_j)), \tag{3}$$

where $(w_u)_{ij}$ is the weight of the edge between vertex $\boldsymbol{u}_i$ and $\boldsymbol{u}_j$, $f(x)$ is the penalty function and remains to be defined. The multiplication of $(w_u)_{ij}$ is to reflect the relationship strength indicated by the weight. If an unweighted network is given, $(w_u)_{ij}$ is set to a constant, such as 1.

As an appropriate penalty function, $f(x)$ must be monotonically decreasing, and decreasing to zero when the probability approaches one. Monotonically decreasing guarantees that smaller penalties are given to larger probabilities of links. This property is necessary as we only expect to penalize small probabilities of links required by the network structure preserving. Besides, for the sake of simple optimization, $f(x)$ should be convex and continuously differentiable. This property makes the loss function (3) solvable by commonly used gradient descent algorithms, and ensures a global optimal solution. A simple function bearing these two properties is the negative natural logarithmic function, and hence Eq. (3) is reformulated as follows:

$$L(U) = -\sum_{(i,j)\in E_u} (w_u)_{ij} \times \log(p(\boldsymbol{u}_i, \boldsymbol{u}_j)) \tag{4}$$

Similarly, the penalty function $f(x)$ for pairs of vertices without an edge is also defined in this way except that the necessary property is adjusted to be monotonically increasing, and approaching zero when the probability approaches zero. Accordingly, the loss function to penalize large probabilities of non-existing edges can be formulated as follows:

$$L(U) = -\sum_{(h,k)\notin E_u} \log(1 - p(\boldsymbol{u}_h, \boldsymbol{u}_k)), \tag{5}$$

where $\boldsymbol{u}_h$ and $\boldsymbol{u}_k$ is a pair of vertices without an edge compared with a pair of vertices $\boldsymbol{u}_i$ and $\boldsymbol{u}_j$ with an edge. In the rest of paper, pairs of vertices with an edge and pairs of vertices without an edge are denoted in this way. It is not practical for this loss function to be summarized on total pairs of vertices without edges when dealing with a large-scale network. To compromise, the number of pairs of vertices without edges is sampled to several times larger than that of pairs with an edge.

The loss functions for G_v and pairs of vertices from different networks are all quantified in this way. All these loss functions should be jointly optimized as each sub-network can provide complementary information to the other one through the inter-network edges. A straightforward way to combine all these loss functions is to directly add them together. We leave more sophisticated ways for the combination as future work. Hence, adding ℓ_2-norm and ℓ_1-norm regularization terms to avoid overfitting, the overall loss function for embedding the coupled heterogeneous network $G_{uv}(G_u, E_{uv}, W_{uv}, G_v)$ is quantified as follows: $L(U, M, V) =$

$$\begin{aligned}
&-\Big[\sum_{(i,j)\in E_u} (w_u)_{ij} \log(p(\boldsymbol{u}_i, \boldsymbol{u}_j)) + \sum_{(i,j)\in E_{uv}} (w_{uv})_{ij} \log(p(\boldsymbol{u}_i, \boldsymbol{v}_j)) \\
&+ \sum_{(i,j)\in E_v} (w_v)_{ij} \log(p(\boldsymbol{v}_i, \boldsymbol{v}_j))\Big] - \Big[\sum_{(h,k)\notin E_u} \log(1 - p(\boldsymbol{u}_h, \boldsymbol{u}_k)) \\
&+ \sum_{(h,k)\notin E_{uv}} \log(1 - p(\boldsymbol{u}_h, \boldsymbol{v}_k)) + \sum_{(h,k)\notin E_v} \log(1 - p(\boldsymbol{v}_h, \boldsymbol{v}_k))\Big] \\
&+ \lambda \sum_{n=1}^{N_u} ||\boldsymbol{u}_n||_2 + \beta||M|| + \eta \sum_{n=1}^{N_v} ||\boldsymbol{v}_n||_2,
\end{aligned} \tag{6}$$

where λ, β, and $\gamma \in \mathbb{R}$ are regularization coefficients, N_u and N_v are the number of the vertices in G_u and G_v, respectively. The ℓ_1-norm for the harmonious embedding matrix is to perform feature selection for reconciling the two latent spaces at it would introduce sparsity.

5. THE OPTIMIZATION ALGORITHM

Minimizing the loss function $L(U, M, V)$ is a convex optimization problem. We thus can employ gradient-based algorithms to perform the optimization. The gradient for $\boldsymbol{u}_i$ can be obtained by differentiating $L(U, M, V)$ with respect to $\boldsymbol{u}_i$ as follows: $\frac{\partial L(U,M,V)}{\partial \boldsymbol{u}_i} =$

$$\begin{aligned}
&-\sum\left[\frac{(w_u)_{ij}\exp\{-\boldsymbol{u}_i^\top \boldsymbol{u}_j\}}{1+\exp\{-\boldsymbol{u}_i^\top \boldsymbol{u}_j\}}\boldsymbol{u}_j + \frac{(w_{uv})_{ij}\exp\{-\boldsymbol{u}_i^\top M\boldsymbol{v}_j\}}{1+\exp\{-\boldsymbol{u}_i^\top M\boldsymbol{v}_j\}}M\boldsymbol{v}_j\right] \\
&+\sum\left[\frac{\boldsymbol{u}_k}{1+\exp\{-\boldsymbol{u}_i^\top \boldsymbol{u}_k\}} + \frac{(w_{uv})_{ik}\exp\{-\boldsymbol{u}_i^\top M\boldsymbol{v}_k\}}{p(\boldsymbol{u}_i, \boldsymbol{v}_k) - 1}p^2(\boldsymbol{u}_i, \boldsymbol{v}_k)\right. \\
&\left.\times M\boldsymbol{v}_k\right] + \lambda\sum_{d=1}^{D_u} 2(\boldsymbol{u}_i^d),
\end{aligned} \tag{7}$$

where D_u is the dimension of latent features. All the summarization subscripts are left out due to space consideration.

The gradient for $\boldsymbol{v}_i$ can be obtained by differentiating $L(U, M, V)$ with respect to $\boldsymbol{v}_i$ as follows: $\frac{\partial L(U,M,V)}{\partial \boldsymbol{v}_i} =$

$$\begin{aligned}
&-\sum\left[\frac{(w_v)_{ij}\exp\{-\boldsymbol{v}_i^\top \boldsymbol{v}_j\}}{1+\exp\{-\boldsymbol{v}_i^\top \boldsymbol{v}_j\}}\boldsymbol{v}_j + \frac{(w_{uv})_{ji}\exp\{-\boldsymbol{u}_j^\top M\boldsymbol{v}_i\}}{1+\exp\{-\boldsymbol{u}_j^\top M\boldsymbol{v}_i\}}Mu_i\right] \\
&+\sum\left[\frac{\boldsymbol{v}_k}{1+\exp\{-\boldsymbol{v}_i^\top \boldsymbol{v}_k\}} + \frac{(w_{uv})_{ki}\exp\{-\boldsymbol{u}_k^\top M\boldsymbol{v}_i\}}{p(u_k, \boldsymbol{v}_i) - 1}p^2(\boldsymbol{u}_k, \boldsymbol{v}_i)\right. \\
&\left.\times M\boldsymbol{u}_k\right] + \eta\sum_{d=1}^{D_v} 2(\boldsymbol{v}_i^d),
\end{aligned} \tag{8}$$

where D_v is the dimension of latent features of V.

The loss function $L(U, M, V)$ is not differentiable with respect to zero elements of M. We thus employ sub-gradient gradients for M, which can be obtained by differentiating $L(U, M, V)$ with respect to M as follows: $\frac{\partial L(U,M,V)}{\partial M} =$

$$\begin{aligned}
&-\sum\left[\frac{(w_{uv})_{ij}\exp\{-\boldsymbol{u}_i^\top M\boldsymbol{v}_j\}}{1+\exp\{-\boldsymbol{u}_i^\top M\boldsymbol{v}_j\}}\boldsymbol{u}_i\boldsymbol{v}_j^\top\right] + \sum\left[\frac{\exp\{-\boldsymbol{u}_h^\top M\boldsymbol{v}_k\}}{p(\boldsymbol{u}_h, \boldsymbol{v}_k) - 1}\right. \\
&\left.\times p^2(\boldsymbol{u}_h, \boldsymbol{v}_k)\boldsymbol{u}_i\boldsymbol{v}_j^\top\right] + \beta\sum_{i=1}^{D_u}\sum_{i=1}^{D_v} sign(m_{ij}),
\end{aligned} \tag{9}$$

where $sign(x) = 1$ if $x > 0$, $sign(x) = -1$ if $x < 0$, and $sign(x) = 0$ if $x = 0$, m_{ij} is the i_{th} row and j_{th} column element of M. Considering that m_{ij} falling on zero is a rare case in practice, we manipulate the value of m_{ij} to zero if it crosses zero in the descent process. This mechanism is called lazy update [6] to encourage sparsity.

With these gradients, we propose a gradient-based alternating optimization algorithm in which the loss function is minimized with respect to one type of variable at a time until convergence. This alternating optimization algorithm can replace the difficult joint optimization over the three variables with a sequence of easier optimizations [4]. With respect to the alternation of variables, it is not performed until the current variable-specific minimization converges. This is because each type of variable would influence the other two

Input : $G_{uv}(G_u, E_{uv}, W_{uv}, G_v)$, d_u, d_v, λ, β, and η
Output : Embeddings of U and V

Initializing U, M,V by assigning zeors;
while *(not converge)* **do**
  compute gradient $\frac{\partial L(U)}{\partial \boldsymbol{u}_i}$ for all U;
  step size $\eta_u \leftarrow$ backtracking line search;
  Update $\boldsymbol{u}_i^{p+1} = \boldsymbol{u}_i^p - \eta_u \frac{\partial L(U)}{\partial \boldsymbol{u}_i}$ for all U simultaneously, where p is the iteration index;
perform gradient descent for pre-training V by procedures similar to lines from 2th to 5th line;
while *(not converge)* **do**
  Fixing U and V, find the optimal M with gradient descent;
  Fixing V and M, find the optimal U with gradient descent;
  Fixing U and M, find the optimal V with gradient descent;
return Embeddings of U and V

Algorithm 1: Optimization Algorithm for the EOE

types of variables, and if the current variable is not optimized, it may propagate negative influences. An intuitive example is that if two similar words with edges are not close in the latent space before the alternation to authors, two authors that should be close due to co-authorships may be put away from each other in the case that they have edges to different aforementioned words. The pseudo-codes of the proposed algorithm are presented in Algorithm 1.

In Algorithm 1, pre-training is performed on G_u and G_v to learn latent features for U and V separately, which is again for not propagating negative influences to other variables. The gradient $\frac{\partial L(U)}{\partial \boldsymbol{u}_i^a}$ is not presented before, but can be easily obtained by removing polynomials including M from the gradient $\frac{\partial L(U,M,V)}{\partial \boldsymbol{u}_i^a}$. For the learning rate, we employ the backtracking line search [2] to learn an appropriate one for each of the iteration. The condition to determine whether all minimizations converge is that the relative loss between current iteration and last iteration is smaller than a considerably small value, such as 0.001.

The complexity of the Algorithm 1 is proportional to the complexity of the gradients of vertex embeddings and the harmonious embedding matrix. The complexities of these gradients are at the same level, which is $O(nd_u \times d_v)$, where n is number of pairs of vertices with edges, d_u and d_v are the dimensions of embeddings of vertices of U and V, respectively. As a result, the complexity of Algorithm 1 is $O(nd_u \times d_v i)$, where i is the iteration times. Accordingly, the proposed EOE can be solved in polynomial time.

6. EVALUATION

6.1 Experiment settings

The experiment is to evaluate the effectiveness of the latent features learned by the proposed EOE model on common data mining tasks including visualization, link prediction, multi-class classification, and multi-label classification. And we compare the proposed EOE against the following three latent feature learning methods:

1. Spectral clustering (SC) [20]: This method proposes to use spectral clustering to learn latent features. Specifically, the top d eigenvectors of the normalized Laplacian matrix are used as the feature vectors.
2. DeepWalk [13]: DeepWalk learns latent features for vertices by modeling random walks as sentences of a natural language, and then re-formulating language modeling as its learning objective. Since DeepWalk is only applicable to unweighted edges, all weights are set to 1 as inputs to DeepWalk.
3. LINE [17]: LINE is proposed to embed large-scale information networks, and is applicable to directed, undirected, weighted and unweighted networks. The proposed EOE is applicable to large-scale network as well since it can be solved in polynomial time. Moreover, the EOE model is applicable to all type networks in terms of edge direction and weight as it puts no requirements on directions of edges and can handle edge weights. LINE has two variants, LINE(1st) and LINE(2nd), which preserve the first-order interaction and the second-order interaction, respectively. Correspondingly, the proposed EOE model preserves the first-order interaction. Since the re-weighting and re-balancing mechanisms are unknown to combine embeddings learned by LINE(1st) and LINE(2nd), we do not make the comparison with LINE (1st+2nd).

For the implementation of the EOE, we set the embedding length as 128, which is used in DeepWalk and LINE, ratio of pairs of vertices without an edge to pairs of vertices with an edge as 5, which is used in LINE, commonly used settings for backtracking line search, all the coefficients for regularization terms as 1, and 0.001 as the relative loss difference to determine whether the gradient decent algorithm converges.

6.2 Embedding Visualization

Visualization of embeddings on a two dimensional space is an important application of network embedding [17]. If the learned embeddings preserve the network structure well, visualization provides an easy way to generate the layout of a network. The author and word network is visualized to illustrate this point. We sample a larger network from the DBLP dataset than the one used in the introduction. Specifically, popular conferences from four research fields are selected, which are SIGMOD, VLDB, ICDE, EDBT, and PODS for Database, KDD, ICDM, SDM, and PAKDD for Data Mining, ICML, NIPS, AAAI, IJCAI and ECML for Machine Learning, and SIGIR, WSDM, WWW, CIKM, and ECIR for Information Retrieval. Moreover, the papers published from 2000 to 2009 are selected. Authors with papers less than 3 are filtered out, and stop words, such as "where" and "how", are filtered out. The sampled author network consists of 4941 authors, which have total 17372 co-authorships. Correspondingly, the statistics of word networks are 6615 and 78217, respectively. The number of links between authors and words is 92899. All baselines learn embeddings for authors and words separately. The EOE learns these embeddings via joint inference instead. The embeddings generated by all the methods are of 128 lengths. We thus employ the t-SNE [22] tool to map them into a two-dimension space, which are all presented in Fig. 2.

The visualization of both author sub-network and word sub-network should display a mixture of four clusters as the

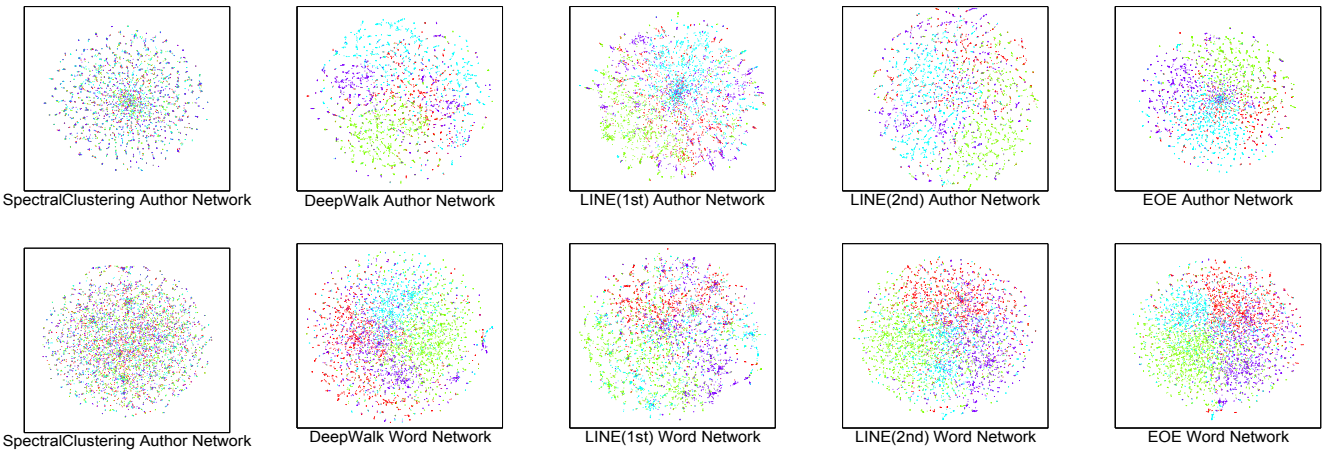

Figure 2: Visualization of embeddings for DBLP coauthorship network and word co-concurrence network, where green points are for authors (words) from DB, light blue for IR, dark blue for DM, and red for ML.

selected four research fields are closely related. The network layouts by Spectral Clustering do not meet the expectation. This is because the learning objective of Spectral Clustering is not to preserve the network structure. Although DeepWalk and LINE work better to some extent, they fail to display a mixture of four distinct clusters as well. The problem with DeepWalk may be that random walks may cross multiple research fields. As a result, data points from different fields may be distributed together. The problem with LINE is similar as authors may have links to others from multiple domains. The network layout by the proposed EOE is the closest to the expectation. This is because the EOE can overcome the aforementioned problem by utilizing the complementary information. More specifically, even though some authors have edges to others from multiple fields, the words from their papers can provide complementary information about their major fields. The case with words is similar.

6.3 Link Prediction

The link prediction problem [12] refers to inferring new interactions between network vertices by measuring the similarity between them. We deploy two scenarios of link prediction for evaluation of the latent features, which are future link prediction and missing link prediction. The future link prediction problem is to infer interactions that would happen in the future while the missing link prediction problem is to infer existing interactions that are not observed.

In the future link prediction, we employ the DBLP coauthorship network used in visualization as the training network, and then perform predictions of co-authorships that occur during 2010, 2010 and 2011, 2011 and 2012, and the three-year interval from 2010 to 2012. Besides the future coauthorships (positive) during these four periods, the same number of pairs of vertices without co-authorship (negative) are randomly generated for measuring the capability of detecting negative co-authorships. The similarity used to infer new co-authorships is the probability quantified by the following equation:

$$p(\boldsymbol{v}_i, \boldsymbol{v}_j) = \frac{1}{1 + \exp\{-\boldsymbol{v}_i^\top \boldsymbol{v}_j\}}, \tag{10}$$

where $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ are embeddings of two vertices.

AUC	During 2010	2010-2011	2011-2012	2010-2012
SC	63.56	62.02	62.21	62.03
DeepWalk	77.61	77.33	75.88	75.09
LINE(1st)	70.55	71.96	70.18	72.14
LINE(2nd)	77.18	76.48	75.47	75.26
EOE	**79.37**	**77.92**	**77.10**	**77.86**

Table 1: AUC scores on future link prediction for the DBLP coauthorship network. Scores have been multiplied by 100%.

The commonly used AUC (area under the curve) score, which measures the general predictive power of binary classifiers, is employed to evaluate the performance as presented in Table 1. Table 1 shows that the proposed EOE outperforms consistently three baselines in all four link prediction tasks. Also, all the network embedding methods significantly outperform Spectral Clustering, which demonstrates the effectiveness of the network embedding to learn latent features for network vertices.

To explore the reason behind the superior performance of the EOE from the experiment perspective, we examine how all these methods make predictions on test instances, and present two representatives in Table 2. The ground truth is that these two co-authorships occur in CVPR'10 and SIGIR'10, respectively. All the baselines give low estimates of probabilities that they would co-author in the future. By contrast, the EOE can give the closest prediction by leveraging the information that they share similar research interests, which are demonstrated by the common words. The probabilities given by LINE(2nd) are left out because it produces probabilities close to 100% for all test instances including non-existing co-authorships. We thus do not know how to interpret it. The reason behind the performance of LINE(2nd) is beyond the scope of this paper. Nevertheless, we know that the learning objective of LINE(2nd), which is to preserve second-order similarity, does not match the first-order similarity used to infer new interactions.

For the missing link prediction, we deploy other three tasks, which are paper citation prediction, friendship prediction for social network users, and gene interaction prediction. For the paper citation prediction, we construct a paper and word network, where the word sub-network is constructed in the same way as used in the previous author and word

Coauthorship	SC	DeepWalk	LINE(1st)	**EOE**	Common words	Coauthored paper
Michael I. Jordan, Raquel Urtasun	0.42	0.65	0.51	**0.84**	factorization gaussian discriminative matrix	Sufficient dimension reduction for visual sequence classification (CVPR'10)
Nick Craswell, Filip Radlinski	0.41	0.58	0.48	**0.80**	search query web relevance	Metrics for assessing sets of subtopics (SIGIR'10)

Table 2: Comparison on detailed predictions, where numbers are probabilities of positive links.

Link Prediction	Algorithm	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%
Paper citation	SC	–	48.29	51.33	56.88	63.85	68.45	71.38	73.73	76.58	78.07
	DeepWalk	–	55.12	71.27	80.74	85.26	88.51	89.86	91.81	92.75	93.06
	LINE(1st)	–	50.35	58.83	65.64	72.19	76.47	80.07	83.08	85.07	86.78
	LINE(2nd)	–	55.78	71.40	78.79	83.07	86.14	88.44	89.81	91.76	91.75
	EOE	51.96	**57.87**	**78.30**	**82.30**	**87.52**	**91.49**	**92.02**	**92.60**	**93.67**	**94.85**
Gene interaction	SC	–	43.19	44.31	48.23	51.56	53.51	55.17	57.13	58.52	59.38
	DeepWalk	–	53.08	57.60	63.85	69.13	71.07	74.77	74.11	74.65	77.14
	LINE(1st)	–	40.84	36.93	36.12	35.54	37.73	35.70	35.43	35.41	36.23
	LINE(2nd)	–	39.52	41.71	42.91	46.00	46.67	49.30	50.92	52.88	54.02
	EOE	43.17	**58.00**	**62.03**	**63.35**	**69.85**	**72.13**	**76.96**	**75.04**	**79.80**	**81.71**
Blog user friendship	SC	–	44.66	46.14	47.88	49.35	51.78	53.12	54.16	55.89	57.05
	DeepWalk	–	55.67	60.46	63.36	65.02	67.42	69.15	70.66	72.14	73.63
	LINE(1st)	–	38.31	39.84	41.39	41.61	42.09	42.95	44.15	43.99	44.03
	LINE(2nd)	–	42.73	48.62	53.08	55.32	57.05	58.50	61.14	62.35	62.42
	EOE	52.42	**55.99**	**61.32**	**65.44**	**67.31**	**72.68**	**71.39**	**75.93**	**80.50**	**84.43**

Table 3: AUC scores on link prediction when different ratios of links are used in the training phase.

Multi-class classification	Algorithm	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Paper classification	SC	45.12	46.69	50.22	52.65	54.26	55.32	56.13	57.09	57.59	58.32
	DeepWalk	47.90	56.39	62.20	64.80	66.55	67.46	68.75	69.28	70.06	70.32
	LINE(1st)	53.78	57.58	59.48	61.72	63.39	64.38	66.25	67.49	68.05	68.59
	LINE(2nd)	44.10	48.42	54.25	58.36	61.28	61.96	63.28	63.89	64.85	64.99
	EOE	**64.53**	**66.05**	**67.16**	**68.01**	**68.73**	**69.26**	**69.45**	**70.01**	**70.39**	**70.58**

Table 4: Accuracy on prediction of research field of papers when different ratios of links are used in the training phase.

		Micro-F1					Macro-F1				
Multi-label classification	Algorithm	20%	40%	60%	80%	100%	20%	40%	60%	80%	100%
Author classification	SC	59.32	61.51	62.85	64.95	66.63	59.10	61.47	62.31	64.12	66.20
	DeepWalk	71.16	74.72	77.05	78.23	78.91	71.85	74.55	76.92	77.30	78.12
	LINE(1st)	70.55	73.56	76.44	78.21	78.43	70.21	73.03	76.15	76.38	78.27
	LINE(2nd)	65.66	70.27	72.69	75.25	75.84	64.62	69.60	71.82	74.96	75.31
	EOE	**77.63**	**78.19**	**78.57**	**79.16**	**79.52**	**76.96**	**77.16**	**77.38**	**77.88**	**78.51**
User classification	SC	52.02	55.57	55.78	58.15	61.01	50.01	51.56	52.53	54.32	55.12
	DeepWalk	63.45	67.73	71.42	71.56	73.77	36.58	47.76	55.73	55.97	60.25
	LINE(1st)	64.42	67.64	70.98	72.45	73.52	39.03	45.90	54.16	57.92	60.33
	LINE(2nd)	62.66	65.26	70.24	72.01	73.51	33.09	36.58	51.57	57.56	60.62
	EOE	**72.41**	**73.77**	**77.15**	**78.19**	**80.53**	**62.35**	**65.59**	**68.71**	**70.88**	**75.81**

Table 5: Micro-F1 and Macro-F1 scores on author domain classification and BlogCatalog user category classification when different ratios of links are used in the training phase.

network. And the network is sampled from the same conferences from 2000 to 2009 as well. Moreover, the references that are not published during the time span are filtered out, and words with frequency less than 5 are filter out. As a result, the paper citation sub-network consists of 6904 papers and 19801 citations, the word sub-network consists of 8992 words and 118518 edges, and the number of edges between papers and words is 59471.

For gene interaction prediction, we construct a gene and chemical compound network from a SLAP [7] dataset, which consists of multiple types of nodes, such as genes, chemical compounds, and drugs, and edges between them. Links between chemical compounds are established if they share the same chemical ontology, which results in 883 chemical compounds and 70746 edges between them. We then select genes with edges to the selected chemical compounds, which results in 2472 genes and 4369 edges between genes. The number of edges between genes and compounds is 1134.

For social network user friendship prediction, we construct a BlogCatalog user and word network, where BlogCatalog [23] is social blog directory where its users can register blogs under multiple categories, and can make friends. We select users with major categories of four popular types including Art, Computers, Music, and Photography, which results in 5009 users and 28406 edges. And their blogs related to the four categories are used to construct the word sub-network, which consists of 9247 words and 915655 edges. Words with frequency less than 8 are filtered out. Size of the sliding window is set as 5 for the generation of co-concurrence relationship between words. The number of edges between users and words is 350434.

We deploy 10 experiments in which the training edges range from 0% to 90% of all the edges. The results are presented in Table 3. 0% of edges imposes the cold-start problem, which cannot be solved by any of the baselines as all of them rely on the edges to learn latent features. By contrast, the EOE can leverage the complementary information so as to tackle the problem. When more edges are used for training, EOE still outperforms the baseline methods consistently. This illustrates the advantages of learning embedding jointly for coupled networks.

6.4 Multi-class Classification

In multi-class classification, it is to predict the class label of each instance and the class can take multiple values. For the demonstration of this application, the paper and word network used in the paper citation prediction task is employed here. Specifically, the classification task is to predict the research field of each paper, which can be one of the four fields, namely Database, Data Mining, Machine Learning and Information Retrieval.

Similarly to the experiment settings in the missing link prediction, we deploy 10 experiments where the training links range from 10% to 100% of the total links. After learning the latent features, we employ the SVM with linear kernel implemented in Weka [10] to perform the classification task. Results of averaged accuracy of 10-fold cross validation are presented in Table 4. The proposed EOE outperforms the three baselines in all the ten tasks, especially those with a small proportion of links, such as 10% and 20%. Moreover, the EOE performs pretty well in tasks with a few links as compared with baselines. This is expected because the abstract of a paper has rich information about the research

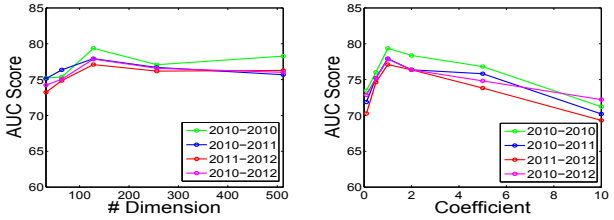

Figure 3: Parameter sensitivity

filed of the paper. And the EOE can elegantly take advantage of this information.

6.5 Multi-label Classification

In multi-label classification, more than one labels are assigned to an instance. The author and word network used in visualization is used to learn the representations of authors. Some authors may be multi-domain experts as they publish papers in conferences of multiple fields. We employ the binary-relevance based SVM implemented in MEKA [14] to perform the multi-domain prediction task. Also, the BlogCatalog user and word network used in link prediction task is also used in this demonstration as users may register blogs under multiple categories.

The experiment settings are similar to those in the multi-class classification except that only five runs of experiment are conducted due to limited space, and the results on Micro-F1 and Macro-F1 are presented in Table 5. Similar findings can be observed that the proposed EOE outperforms all the baselines in all tasks, and the advantage is more visible when there are few links available in the training phase. Combining the results from the visualization, the link prediction tasks, the multi-class classification, it shows that complementary information provided by the inter-networks edges is beneficial as it can make latent features learned from intra-network edges more comprehensive and accurate, and is more important when suffering the cold-start problem.

6.6 Parameter Sensitivity

There are two major types of hyper-parameters in the proposed optimization algorithm, which are the coefficients of regularization terms and the dimension of latent features. In all the previous experiments, coefficients and dimension are set as constants, which are 1.0 and 128, respectively. We thus in this section study how these parameters influence the performance of the EOE by setting values of the dimension as 32, 64, 128, 256, and 512, and values of coefficients as 0.1, 0.5, 1.0, 2.0, 50.0 and 10.0, respectively. Please be noted that the value of coefficient remains as 1.0 while studying the dimension, and the value of dimension remains as 128 while studying the coefficient.

Due to limited space, we only present results on future link prediction of co-authorships in Fig. 3. From the left-hand figure in Fig.3, we see that the performance of the EOE is not very sensitive to the dimension of latent features as long as it is not too small (eg., less than 32). And the optimal performance is obtained at the dimension of 128, which is used in the previous experiments as well. Form the right-hand figure, the observation is that the performance of the EOE is relatively more sensitive to the regularization coefficients. Nevertheless, the performance around the coefficient of 1.0 is relatively stable and is at the optimal.

7. CONCLUSION AND FUTURE WORK

This paper proposes the embedding of embedding (EOE) for joint embedding of coupled heterogeneous networks, which are two different but related networks connected by inter-network edges. In this case, each of the networks can provide complementary information to the other. The complementary information can make latent features learned only from intra-network edges more comprehensive and accurate, especially in the cold-start scenario in which intra-network edges are limited or unavailable. To reconcile the heterogeneities between two different networks, the proposed EOE incorporates a harmonious embedding matrix to further embed embeddings from one latent space to another latent space. Empirical evaluation on a variety of coupled heterogeneous networks demonstrates that the EOE outperforms state-of-the-art embedding models for a single network in four applications including visualization, link prediction, multi-class classification, and multi-label classification. In the future, we plan to extend the proposed model for more than two networks so as to fuse multiple sources of information.

8. ACKNOWLEDGMENTS

This work is supported in part by PolyU project of strategic importance 1-ZE26, HK RGC General Research Found grant PolyU·5/04/13E, NSF through grants IIS-1526499, and CNS-1626432.

9. REFERENCES

[1] A. Ahmed, N. Shervashidze, S. Narayanamurthy, V. Josifovski, and A. J. Smola. Distributed large-scale natural graph factorization. In *Proceedings of the 22nd international conference on World Wide Web*, pages 37–48. International World Wide Web Conferences Steering Committee, 2013.

[2] L. Armijo. Minimization of functions having lipschitz continuous first partial derivatives. *Pacific Journal of mathematics*, 16(1):1–3, 1966.

[3] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, volume 14, pages 585–591, 2001.

[4] J. C. Bezdek and R. J. Hathaway. Some notes on alternating optimization. In *AFSS International Conference on Fuzzy Systems*, pages 288–300. Springer, 2002.

[5] B. Cao, X. Kong, and S. Y. Philip. Collective prediction of multiple types of links in heterogeneous information networks. In *2014 IEEE International Conference on Data Mining*, pages 50–59. IEEE, 2014.

[6] B. Carpenter. Lazy sparse stochastic gradient descent for regularized multinomial logistic regression. *Alias-i, Inc., Tech. Rep*, pages 1–20, 2008.

[7] B. Chen, Y. Ding, and D. J. Wild. Assessing drug target association using semantic linked data. *PLoS Comput Biol*, 8(7):e1002574, 2012.

[8] T. F. Cox and M. A. Cox. *Multidimensional scaling*. CRC press, 2000.

[9] S. Fortunato. Community detection in graphs. *Physics reports*, 486(3):75–174, 2010.

[10] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[11] J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

[12] D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031, 2007.

[13] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014.

[14] J. Read, P. Reutemann, B. Pfahringer, and G. Holmes. Meka: a multi-label/multi-target extension to weka. *Journal of Machine Learning Research*, 17(21):1–5, 2016.

[15] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[16] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93, 2008.

[17] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077. International World Wide Web Conferences Steering Committee, 2015.

[18] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 990–998. ACM, 2008.

[19] L. Tang and H. Liu. Relational learning via latent social dimensions. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 817–826. ACM, 2009.

[20] L. Tang and H. Liu. Leveraging social media networks for classification. *Data Mining and Knowledge Discovery*, 23(3):447–478, 2011.

[21] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[22] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

[23] X. Wang, L. Tang, H. Gao, and H. Liu. Discovering overlapping groups in social media. In *2010 IEEE international conference on data mining*, pages 569–578. IEEE, 2010.

[24] X. Wei, B. Cao, W. Shao, C.-T. Lu, and P. S. Yu. Community detection with partially observable links and node attributes. In *IEEE International Conference on Big Data*, 2016.

[25] X. Wei, B. Cao, and P. S. Yu. Unsupervised feature selection on networks: A generative view. In *AAAI*, 2016.

[26] X. Wei, S. Xie, and P. S. Yu. Efficient partial order preserving unsupervised feature selection on networks. In *SDM*, pages 82–90. SIAM, 2015.