Recommendation with Multi-Source Heterogeneous Information

Li Gao[*‡], **Hong Yang**[†], **Jia Wu**[♯], **Chuan Zhou**[*‡], **Weixue Lu**[§], **Yue Hu**[*‡]
[*]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[†]Centre for Artificial Intelligence, University of Technology Sydney, Australia
[♯]Department of Computing, Macquarie University, Sydney, Australia
[§]Data Science Lab, JD.com, Beijing, China
[‡]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
{gaoli, zhouchuan}@iie.ac.cn, hong.yang@student.uts.edu.au, Jia.Wu@mq.edu.au

Abstract

Network embedding has been recently used in social network recommendations by embedding low-dimensional representations of network items for recommendation. However, existing item recommendation models in social networks suffer from two limitations. First, these models partially use item information and mostly ignore important contextual information in social networks such as textual content and social tag information. Second, network embedding and item recommendations are learned in two independent steps without any interaction. To this end, we in this paper consider item recommendations based on heterogeneous information sources. Specifically, we combine item structure, textual content and tag information for recommendation. To model the multi-source heterogeneous information, we use two coupled neural networks to capture the deep network representations of items, based on which a new recommendation model Collaborative multi-source Deep Network Embedding (CDNE for short) is proposed to learn different latent representations. Experimental results on two real-world data sets demonstrate that CDNE can use network representation learning to boost the recommendation performance.

1 Introduction

With the massive amount of data generated by online social services, recommender systems are playing an important role in connecting users and information resources. To tackle the sparsity problem of user-item interactions, hybrid recommendation methods which combine collaborative filtering and auxiliary information sources such as item contents have shown promising results [Wang and Blei, 2011; Zhang *et al.*, 2016; Gao *et al.*, 2017; Yamasaki *et al.*, 2017; Dong *et al.*, 2017]. These methods focus on extracting a set of important factors for items obtained from auxiliary information.

Recently, network embedding [Perozzi *et al.*, 2014; Chang *et al.*, 2015; Zhang *et al.*, 2017; Liu *et al.*, 2018] has gained increasing popularity in social network recommendations. Network embedding aims to learn a vector representation of each node by mapping it into a low-dimensional vector space while preserving its neighborhood relationship. Because network embedding can capture the neighborhood similarity and community membership, it has been popularly used in recommendations [Chen *et al.*, 2015; Zhao *et al.*, 2016]. For example, the work [Zhao *et al.*, 2016] learns the network representation of each node in the built k-partite adoption network. The recommendation task is considered as a similarity evaluation problem by ranking the cosine similarity between user and item representations.

However, previous studies on network embedding for recommendations suffer from two shortcomings. First, they do not fully use the item information. The contextual information of items are often ignored, which leads to a shallow representation of the network. Second, network embedding and item recommendations are learned independently and their interactions are often ignored.

To address the above shortcomings, we integrate deep network representations of items with collaborative filtering for recommendation. Item information are combined from multiple heterogeneous information sources, such as item structure, textual content and tag information. We design a new deep network embedding component by using two coupled neural networks which can extract deep representations from multiple heterogeneous information sources. To combine collaborative filtering and network representations obtained from multiple sources, we present a new Collaborative multi-source Deep Network Embedding method (CDNE for short) to learn different latent representations. Figure 1 shows an illustration of CDNE for item recommendations.

The main contributions are summarized as follows:

- We present a new item recommendation framework that can embed deep network representations obtained from multiple information sources such as structure, textual content and tag information for item recommendations.
- We develop a new method that jointly performs multi-

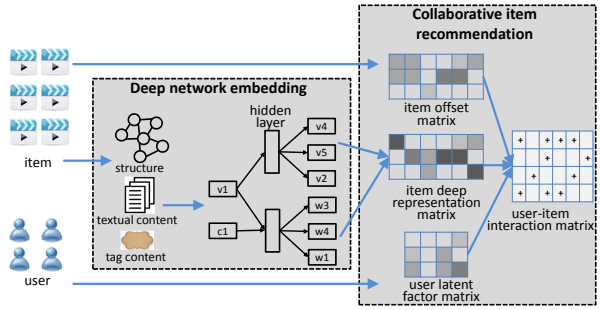


Figure 1: An illustration of the proposed method CDNE. Information obtained from multiple sources, such as item structure, textual content and tag information, is fed into a deep network embedding component to learn network representations of items, where CDNE jointly learns the inter-item relationship, item-content correlation and tag-content correspondence. In the collaborative item recommendation component, the item deep representation matrix, item offset matrix, and user latent factor matrix are combined for item recommendations.

source deep network embedding and collaborative filtering, where deep representations of items and interactions between users and items are learned collaboratively.

- We compare our method with state-of-the-art methods on two real-world data sets to evaluate the performance. Experimental results demonstrate that our method significantly outperforms the baseline methods in terms of the precision and MRR metrics.

2 Problem Statement

We denote users by $\mathcal{U} = \{u_1, u_2, \ldots, u_m\}$, and items by $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$. The browsing history of users is recorded by an $m \times n$ user-item interaction matrix $\mathbf{R}$. $\mathbf{R}_{ij}$ is either a rating score given by u_i on v_j or a missing value ($\mathbf{R}_{ij} = 0$). We consider item information from multiple sources: item structure, textual content and tag information. Given the above item information and the history of interactions between users and items, we aim to recommend each user u_i with a ranked list of interested items.

Since we explore to enhance the quality of recommendation by leveraging deep network representations, each item v_j is mapped to a node of network $G = (\mathcal{V}, \mathcal{E}, \mathcal{D}, \mathcal{C})$, where $e_{ij} \in \mathcal{E}$ denotes the edge relationship[1] from v_i to v_j, $d_j \in \mathcal{D}$ denotes the textual content associated with item v_j, and $c_j \in \mathcal{C}$ denotes the tag information associated with item v_j.

We assume that the connected items are statistically dependent, which learns the inter-item relationship from the random walk corpus generated from item structure $\mathcal{E}$. The textual content $\mathcal{D}$ captures the item-content correlation and the tag information $\mathcal{C}$ represents the tag-content correspondence between item tag and item content.

To obtain deep network representations from multiple sources, we develop a deep network embedding component by utilizing two coupled neural networks, where the deep representation vector $\boldsymbol{\theta}_j$ of item v_j and the latent representation vector $\boldsymbol{l}_j$ of c_j are taken as input. The output is the textual word vectors and the representation vectors of nodes in the contextual window. In the collaborative item recommendation component, the latent factor vector $\boldsymbol{V}_j$ is formulated as a combination of $\boldsymbol{\theta}_j$ and the latent offset vector $\boldsymbol{\epsilon}_j$. We then combine $\boldsymbol{V}_j$ and user latent factor vector $\boldsymbol{U}_i$ for joint learning of user-item interactions.

[1]Without loss of generality, we assume that G is a directed graph. The case of undirected networks can be readily adapted by replacing each undirected edge with two oppositely directed edges.

3 Preliminary: DeepWalk Model

Based on the Skip-gram model [Mikolov *et al.*, 2013a], DeepWalk [Perozzi *et al.*, 2014] constructs a corpus S that consists of random walks generated from the network. Each random walk $s = \{v_1, \ldots, v_{n^s}\}$ is considered as a sentence and each node v_j is regarded as a word in neural language models. Assume that $\mathbf{a}_j = \{v_{j-c}, \ldots, v_{j+c}\} \backslash v_j$ is the context vertices when given the target node v_j. DeepWalk aims to maximize the following objective

$$\sum_{s \in S} \sum_{j=c}^{n^s - c} \sum_{v_i \in \mathbf{a}_j} \ln P(v_i \mid v_j) \tag{1}$$

Note that DeepWalk only utilizes the network structure information for model learning.

4 Our Solution

In this section, we introduce our proposed method CDNE for recommendation, which integrates collaborative filtering with deep network representations of items.

4.1 Deep Network Embedding

Network embedding provides an effective way to capture neighborhood similarity and community membership, which is beneficial for recommendation [Zhang *et al.*, 2017]. We map each item v_j to a node of network G. We adapt DeepWalk model to learn the deep representation vector $\boldsymbol{\theta}_j$ of v_j from not only the network structure information, but also textual content and tag information augmented with each item.

For the random walk sequence generation [Pan *et al.*, 2016], we take network structure as input to construct corpus S. Each walk sequence samples uniformly a random node v_j as the root and randomly jumps to one node chosen from the neighbors of the last node visited.

For item v_j, we assume that 1) the deep representation vector $\boldsymbol{\theta}_j$ is influenced by the random walk sequences that have visited v_j, the textual content d_j, and the tag information c_j; and 2) c_j also specifies the words in d_j, which models the correspondence between item tags and item content. To be specific, as illustrated in Figure 1, we couple two neural networks by the item v_j, indicating that $\boldsymbol{\theta}_j$ acts as the input for both the two neural networks.

The first neural network models the generated random walk sequences, for which the input is $\boldsymbol{\theta}_j$ and the output is the deep representations of its context items $\mathbf{a}_j = \{v_{j-c}, \ldots, v_{j+c}\} \backslash v_j$. The objective function can be formulated as follows

$$\mathcal{L}_s = \sum_{s \in S} \sum_{j=c}^{n^s - c} \sum_{v_k \in \mathbf{a}_j} \ln P(v_k \mid v_j) \tag{2}$$

The second neural network models the textual content d_j, for which the input is $\boldsymbol{\theta}_j$ and $\boldsymbol{l}_j$, and the output is the latent vectors of words in d_j. $\boldsymbol{l}_j$ denotes the latent tag representation vector for c_j. Assume that $\mathbf{w}_j = \{w_1^j, \ldots, w_{2c}^j\}$ is a sequence of textual words within a contextual window. The objective aims to maximize the following likelihood function

$$\begin{aligned}\mathcal{L}_t &= \sum_{j=1}^{n} \ln P(\mathbf{w}_j \mid v_j) + \sum_{j=1}^{n} \ln P(\mathbf{w}_j \mid c_j) \\ &= \sum_{j=1}^{n} \sum_{w_k \in \mathbf{w}_j} \ln P(w_k \mid v_j) + \sum_{j=1}^{n} \sum_{w_k \in \mathbf{w}_j} \ln P(w_k \mid c_j)\end{aligned} \tag{3}$$

Note that the first term is similar to the paragraph vector model [Le and Mikolov, 2014] that learns the latent representation for each document from the textual information.

Combining the above two objectives in Eq. (2) and Eq. (3), our coupled neural networks aim to maximize the following objective function

$$\mathcal{L}_{st} = \sigma_s^{'} \mathcal{L}_s + \sigma_t^{'} \mathcal{L}_t \tag{4}$$

where $\sigma_s^{'}$ and $\sigma_t^{'}$ are used to balance the weights of item structure, textual content and tag information.

As suggested by the previous work [Mikolov *et al.*, 2013b], the probability $P(v_k \mid v_j)$ in Eq. (2) can be calculated by using the softmax function as follows

$$P(v_k \mid v_j) = \frac{\exp(\boldsymbol{o}_{v_j}^T \boldsymbol{o}_{v_k}^{'})}{\sum_{i=1}^{n} \exp(\boldsymbol{o}_{v_j}^T \boldsymbol{o}_{v_i}^{'})} \tag{5}$$

where $\boldsymbol{o}_{v_j}$ and $\boldsymbol{o}_{v_j}^{'}$ are the input and output vector representation of item v_j. $P(w_k \mid v_j)$ and $P(w_k \mid c_j)$ in Eq. (3) can be readily calculated using softmax function as in Eq. (5). After training the neural network model, the input vector $\boldsymbol{o}_{v_j}$ can be used as the deep representation vector $\boldsymbol{\theta}_j$ of v_j.

4.2 Collaborative Item Recommendation

Most successful collaborative filtering recommendation methods are latent factor models, among which matrix factorization performs well [Salakhutdinov and Mnih, 2007; Gao *et al.*, 2016]. We represent users and items in a shared latent low-dimensional space of dimension K, where user u_i is represented by a latent factor vector $\boldsymbol{U}_i \in \mathbb{R}^K$ and item v_j by a latent factor vector $\boldsymbol{V}_j \in \mathbb{R}^K$. The user-item interactions can be formulated as

$$\boldsymbol{R}_{ij} \sim \mathcal{N}(\boldsymbol{U}_i^T \boldsymbol{V}_j, \sigma_{ij}^{-1}) \tag{6}$$

where the variable σ_{ij} serves as a confidence parameter for rating $\boldsymbol{R}_{ij}$. we set $\sigma_{ij} = a$, if u_i has rated v_j; otherwise, $\sigma_{ij} = b$, where a and b are tuning parameters satisfying $a > b > 0$. A similar strategy is used in [Wang and Blei, 2011].

We introduce a latent variable $\boldsymbol{\epsilon}_j \in \mathbb{R}^K$, $\boldsymbol{\epsilon}_j \sim \mathcal{N}(0, \sigma_v^{-1} I)$, to offset the deep network embedding $\boldsymbol{\theta}_j$ when modeling the historical user-item interactions. To collaboratively capture an item's latent deep representation from multi-source items' information and latent factor vector in collaborative filtering, the item latent vector $\boldsymbol{V}_j$ is formulated as

$$\boldsymbol{V}_j = \boldsymbol{\theta}_j + \boldsymbol{\epsilon}_j \tag{7}$$

The generative process of CDNE that recommends items with multi-source deep network embedding is described as follows

1. For user u_i, draw a latent factor vector $\boldsymbol{U}_i \sim \mathcal{N}(0, \sigma_u^{-1} I)$,
2. Considering deep network embedding learned from multiple sources:
   (a) Given item v_j, for the random walk sequence $s \in S$, draw from the probability $P(\mathbf{a}_j \mid v_j)$,
   (b) For each item v_j with its textual content d_j, draw from the probability $P(\mathbf{w}_j \mid v_j)$,
   (c) For each item v_j with its tag information c_j, draw from the probability $P(\mathbf{w}_j \mid c_j)$.
3. For item v_j, draw an item latent offset $\boldsymbol{\epsilon}_j \sim \mathcal{N}(0, \sigma_v^{-1} I)$ and set $\boldsymbol{V}_j = \boldsymbol{\theta}_j + \boldsymbol{\epsilon}_j$,
4. For each user-item pair (u_i, v_j), draw the rating $\boldsymbol{R}_{ij} \sim \mathcal{N}(\boldsymbol{U}_i^T \boldsymbol{V}_j, \sigma_{ij}^{-1})$.

Based on the above steps, computing the full posterior of the parameters is intractable. As suggested by the previous work [Wang and Blei, 2011], maximizing the posterior probability of $\boldsymbol{U}$, $\boldsymbol{V}$, $\boldsymbol{\theta}$ and $\boldsymbol{l}$ is equivalent to minimizing the complete negative log-likelihood as follows

$$\begin{aligned}\min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{\theta},\boldsymbol{l}} & \sum_{i=1}^{m} \sum_{j=1}^{n} \sigma_{ij} (\boldsymbol{R}_{ij} - \boldsymbol{U}_i^T \boldsymbol{V}_j)^2 \\ & -\sigma_s \sum_{s \in S} \sum_{j=c}^{n^s - c} \sum_{v_k \in \mathbf{a}_j} \ln P(v_k \mid v_j) + \sum_{i=1}^{m} \sigma_u \|\boldsymbol{U}_i\|_2^2 \\ & + \sum_{j=1}^{n} \sigma_v \|\boldsymbol{V}_j - \boldsymbol{\theta}_j\|_2^2 - \sigma_t \sum_{j=1}^{n} \sum_{w_k \in \mathbf{w}_j} \ln P(w_k \mid v_j) \\ & -\sigma_t \sum_{j=1}^{n} \sum_{w_k \in \mathbf{w}_j} \ln P(w_k \mid c_j)\end{aligned} \tag{8}$$

where σ_{ij}, σ_s, σ_u, σ_v and σ_t are the weight parameters.

4.3 Parameter Optimization

We use stochastic gradient descent to solve the objective in Eq. (8). As shown in Eq. (5), the probability $P(v_k \mid v_j)$ ($P(w_k \mid v_j)$ or $P(w_k \mid c_j)$ are the same) is calculated by the softmax function. To reduce the computation cost of the gradient of $P(v_k \mid v_j)$, $P(w_k \mid v_j)$ or $P(w_k \mid c_j)$, we instead use the hierarchical softmax [Morin and Bengio, 2005] to approximate the probability distribution.

Specifically, for the calculation of $P(v_k \mid v_j)$, we assign distinct nodes as leaves of a binary tree, which is built using the Huffman coding [Mikolov *et al.*, 2013b] to assign shorter paths to the frequent nodes in random walks. While for $P(w_k \mid v_j)$ or $P(w_k \mid c_j)$, we assign the distinct words as leaves of another binary tree. Then, there is a unique path from the root to each leaf. Suppose the path to node v_j is identified by a sequence of tree nodes, $f_0, f_1, \ldots, f_m$, $P(v_k \mid v_j)$ is then calculated by the probability of the specific path. We have

$$P(v_k \mid v_j) = \prod_{t=1}^{m} P(f_t \mid v_j) \tag{9}$$

where $P(f_t \mid v_j)$ is defined as $P(f_t \mid v_j) = 1/(1 + e^{-\boldsymbol{o}_{v_j}^T \boldsymbol{o}_{f_t}^{'}})$. $\boldsymbol{o}_{f_t}^{'} \in \mathbb{R}^K$ denotes the representation assigned to tree node

f_t's parent. Similarly, as in Eq. (9), we can use hierarchy softmax technique to calculate $P(w_k \mid v_j)$ or $P(w_k \mid c_j)$.

Substituting for $P(v_k \mid v_j)$, $P(w_k \mid v_j)$ and $P(w_k \mid c_j)$ in Eq. (8) using hierarchy softmax, we denote $\mathcal{L}$ as the objective. The model parameter set becomes $\Theta = \{\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{\theta}, \boldsymbol{l}, \Psi\}$, where Ψ denotes the representations assigned to the interval nodes of the binary tree. In each iteration, we use the gradient to update each parameter in Θ as follows

$$\Theta_l = \Theta_l - \alpha_t \frac{\partial \mathcal{L}}{\partial \Theta_l} \tag{10}$$

where α_t is the learning rate. Due to space limitations, we omit the concrete formulations of parameters update.

4.4 Prediction

Given the history of user-item interactions, as well as the multi-source heterogeneous information of items, we obtain the optimal parameters after solving the objective function in Eq. (8). We then recommend each user u_i with a list of items, $v_1, v_2, \ldots, v_h$, using the ranking criterion $\boldsymbol{U}_i^T \boldsymbol{V}_1 \geq \boldsymbol{U}_i^T \boldsymbol{V}_2 \geq \cdots \geq \boldsymbol{U}_i^T \boldsymbol{V}_h$.

5 Experiments

5.1 Data Sets

We use two real-world data sets [Wang *et al.*, 2013] extracted from CiteULike[2] for experimental analysis.

The first data set, called *citeulike-a*, contains 16,980 items (*i.e.,* articles) and 7,386 tags for the items. The second data set, called *citeulike-t*, contains 8,311 tags and 25,975 items. Following the work [Wang *et al.*, 2013], for each article in the data sets, we use its title and abstract as the textual content. We remove stop words and use tf-idf to choose the top 20,000 distinct words as the vocabulary. Because citation information is not provided in CiteULike, to construct social networks between items, we use the user-article information following the same procedure as in [Wang *et al.*, 2013]. For each data set, if two items share more than four users, they are linked in the social network. This is because two articles with similar users typically have similar topics. The items' network structure then contains 294,072 links and 180,103 for *citeulike-a* and *citeulike-t* data set, respectively.

5.2 Experimental Settings

Baselines

We compare our method **CDNE** with the following benchmark methods: 1)**PMF** [Salakhutdinov and Mnih, 2007] is an effective probabilistic matrix factorization method for recommendation; 2) **CTR** [Wang and Blei, 2011] combines traditional collaborative filtering with topic modeling for recommendation; 3) **SLG** [Chen *et al.*, 2015] proposes music recommendation method (we denote it as SLG and adapt it for our recommendation tasks) by integrating the network representations into factorization machines; 4) **NERM** [Zhao *et al.*, 2016] recommends items by ranking the cosine similarity between representations of users and items obtained from the bipartite adoption graph; 5) **CDNE-st** is a variant method of CDNE, which considers multi-source items' information excluding items' network structure; 6) **CDNE-tc** is a variant that excludes items' textual content; 7) **CDNE-ta** is a variant that excludes items' tag information.

Evaluation Metrics

Two metrics, Precision@n (P@n) and Mean Reciprocal Rank (MRR), are used to measure the performance of item recommendations. P@n measures the ratio of successfully recommended items to the top-k recommendations and MRR measures the reciprocal of the first occurrence position of the ground truth item for each user [Liu, 2015]. The two metrics are first calculated separately on each user's recommendation list and then are averaged among all the test users. The higher values of the two metrics are favored in comparisons.

Settings

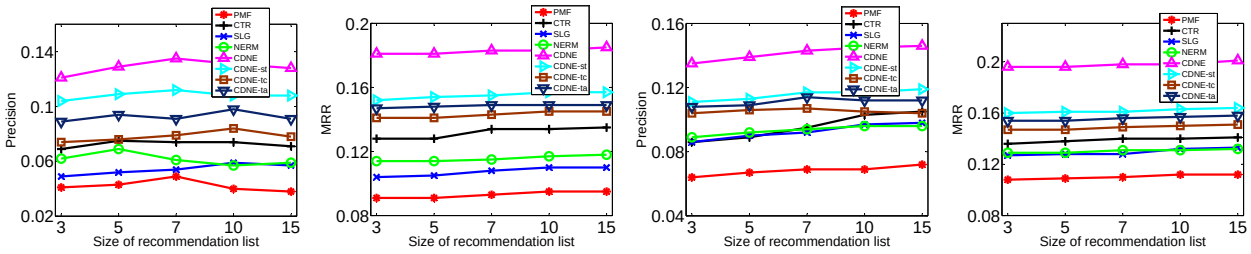
We randomly partition each of the two data sets into training and testing sets. For each user u_i, 70% of the items (*i.e.,* articles) are randomly sampled as the training data, and the remaining 30% are used for testing. We then randomly choose one record of each user from the training data set to construct the validation data. All compared methods use the same number of latent factors K, $K = 200$. For all neural network models, the window size c is set as $c = 8$. The P@10 performance on the validation data for each data set is used to select the optimal parameters. As a result, we set the hyperparameters as $a = 1$, $b = 0.01$, $\sigma_s = 1$, $\sigma_t = 0.5$, $\sigma_u = 0.1$, $\sigma_v = 1$. The learning rate α is set as $\alpha = 0.01$. For each model, we run the experiments 100 times and report the averaged results.

5.3 Experimental Results

Figure 2 shows the experimental results of the precision and MRR metrics on *citeulike-a* and *citeulike-t* data sets with respect to a range of recommendation list sizes. From the figure, we see that the SLG and NERM methods outperform the basic PMF method for both the data sets. For example, for the *citeulike-a* data set, the method SLG averagely improves the precision and MRR by 28.44% and 15.48%, and the method NERM improves by 45.97% and 24.30%. The above results demonstrate that the proper integration of network representations learned from the auxiliary information can boost the recommendation performance. As the representative recommendation method for textual content analysis, even though no network embedding is considered, the method CTR outperforms the methods SLG and NERM for *citeulike-a* (over the range of recommendation list size) and *citeulike-t* (when the recommendation list size is larger than 7) data sets under both the precision and MRR metrics.

We also show the experimental results of variants (CDNE-st, CDNE-tc and CDNE-ta) of our method CDNE. The method CDNE-st performs the best among the three variants for both the data sets. For example, for the *citeulike-a* data set, compared with the CDNE-tc and CDNE-ta methods, CDNE-st averagely improves the precision by 38.36% and 16.85%, and the MRR by 8.39% and 4.45%. The reason may be that the interested topics have more influence on the

[2]The detailed information of the data can be found at http://www.citeulike.org/faq/data.adp. CiteULike allows users to create their own collections of articles. There are titles, abstracts and tags for each article. Other information about the article, such as authors, publications and keywords, is not used in this paper.



(a) Precision (*citeulike-a*) (b) MRR (*citeulike-a*) (c) Precision (*citeulike-t*) (d) MRR (*citeulike-t*)

Figure 2: Precision and MRR performance comparisons with respect to different recommendation list sizes.

Data set	Metric	PMF	CTR	SLG	NERM	CDNE-st	CDNE-tc	CDNE-ta	CDNE
citeulike-a	Precision	0.0422	0.0726	0.0542	0.0616	0.1082	0.0782	0.0926	**0.1288**
	MRR	0.0930	0.1318	0.1074	0.1156	0.1550	0.1430	0.1484	**0.1826**
citeulike-t	Precision	0.0682	0.0956	0.0926	0.0934	0.1154	0.1052	0.1110	**0.1416**
	MRR	0.1102	0.1390	0.1296	0.1304	0.1618	0.1488	0.1558	**0.1978**

Table 1: Average precision and MRR results over a range of recommendation list sizes.

selection of articles for a user, where the topics can be clearly mined from the articles' textual content and tag content, such as the title, abstract and research area (e.g., *artificial intelligence*) the article belongs to. CDNE-tc and CDNE-ta outperforms the CTR method for all the evaluations except for the case that is measured by P@15 on the *citeulike-t* data set.

In all cases, our method CDNE significantly outperforms the baselines. Compared with SLG, instead of finding possible random walk path based on the user-item matrix, we combine items' deep network representations, items' offset vectors and users' latent factor vectors for jointly learning of user-item interactions. The random walks are generated on the network structure (citation information among articles) to learn network representations from items' multi-source information. The method NERM aims to separately learn network embedding on the bipartite adoption graph and make item recommendations with these embeddings, ignoring the interactions between network representation and the objective of recommendation. We instead propose a unified framework to learn different latent factor vectors collaboratively. In this way, we can properly capture their interactions.

Table 1 summarizes the precision and MRR performance of all the compared methods averaged over different recommendation list sizes, which shows the similar results as the above. In summary, by averaging the performance when different recommendation list sizes and different data sets are applied, our method CDNE improves the methods PMF, CTR, SLG, NERM, CDNE-st, CDNE-tc and CDNE-ta by 156.42%, 62.76%, 95.28%, 80.35%, 20.87%, 49.65% and 33.33% in terms of the precision metric, and by 87.92%, 40.42%, 61.32%, 54.82%, 20.03%, 30.31% and 25.00% in terms of the MRR metric.

5.4 Case Study

Following the work [Wang *et al.*, 2013], two articles linked in the social network typically have similar topics. The random walks are generated on the citation graph, where the items' network embeddings are determined by item structure, textual content and tag information. From this point of view, the network embedding can be represented as a latent topic distribution, and users are assumed to have topic interests. Then, we can recommend articles to the users using the latent topic distribution and topic interests. To give a clear illustration of the recommendation performance, Table 2 shows the top-5 recommendation results on the *citeulike-t* data set for an example user (userID: 2975), when the user creates her own collections of the article (#6774). All the compared methods predict the results that the example user may be interested in. Our method CDNE gives the best results.

6 Related Work

Our work relates to the research area of collaborative filtering and network representation learning.

Collaborative filtering based methods [Salakhutdinov and Mnih, 2007] use historical interactions or preferences to recommend items. However, due to the issues of sparsity of user-item interactions, collaborative filtering based models usually suffer from the limited performance. By using auxiliary information, hybrid recommender models [Wang and Blei, 2011; Qiao *et al.*, 2014; Zhang *et al.*, 2016; Gao *et al.*, 2017; Dong *et al.*, 2017] usually obtain better recommendation results. The work [Wang and Blei, 2011] proposed a collaborative topic regression model which combines traditional collaborative filtering methods with topic modeling. A hybrid model [Dong *et al.*, 2017] was proposed to jointly learn deep users and items latent factors from side information and collaborative filtering from the rating matrix.

Hybrid recommendation models aim to learn latent factors of users and items from user-item interactions and auxiliary information, which motivates our work in this paper. On the other hand, network representation learning [Chang *et al.*, 2015; Wang *et al.*, 2016; Ribeiro *et al.*, 2017] attracts increased attention. This type of method aims to learn

Article's title: Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions	
CDNE: **1**. Item-based collaborative filtering recommendation algorithms 2. Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness **3**. Information Filtering: Overview of Issues, Research and Systems **4**. Incorporating contextual information in recommender systems using a multidimensional approach **5**. Factorization meets the neighborhood: a multifaceted collaborative filtering model	PMF: 1. Effective missing data prediction for collaborative filtering 2. GroupLens: applying collaborative filtering to Usenet news 3. Content-based book recommending using learning for text categorization 4. An efficient boosting algorithm for combining preferences **5**. Incorporating contextual information in recommender systems using a multidimensional approach
CTR: 1. Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness 2. Analysis of recommendation algorithms for e-commerce **3**. Item-based collaborative filtering recommendation algorithms 4. Effective missing data prediction for collaborative filtering **5**. Amazon.com recommendations: item-to-item collaborative filtering	SLG: 1. Content-based book recommending using learning for text categorization **2**. Information Filtering: Overview of Issues, Research and Systems 3. GroupLens: applying collaborative filtering to Usenet news 4. Analysis of recommendation algorithms for e-commerce 5. Methods and metrics for cold-start recommendations
NERM: **1**. Item-based collaborative filtering recommendation algorithms 2. An efficient boosting algorithm for combining preferences **3**. Information Filtering: Overview of Issues, Research and Systems 4. Effective missing data prediction for collaborative filtering 5. How oversight improves member-maintained communities	CDNE-st: **1**. Item-based collaborative filtering recommendation algorithms 2. Effective missing data prediction for collaborative filtering 3. Analysis of recommendation algorithms for e-commerce **4**. Amazon.com recommendations: item-to-item collaborative filtering **5**. How much can behavioral targeting help online advertising?
CDNE-tc: 1. Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness **2**. Factorization meets the neighborhood: a multifaceted collaborative filtering model 3. GroupLens: applying collaborative filtering to Usenet news 4. Effective missing data prediction for collaborative filtering **5**. Item-based collaborative filtering recommendation algorithms	CDNE-ta: 1. Methods and metrics for cold-start recommendations 2. Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness **3**. Item-based collaborative filtering recommendation algorithms **4**. Information Filtering: Overview of Issues, Research and Systems **5**. Large-scale behavioral targeting

Table 2: Top-5 recommendation results on the *citeulike-t* data set for an example user (userID: 2975) when the user creates her own collections of the article (#6774). The number in bold indicates that the corresponding article is correctly predicted.

low-dimensional latent representations of nodes in networks. These methods are widely used in network classification [Perozzi *et al.*, 2014], network visualization [Tang *et al.*, 2015b], link prediction [Grover and Leskovec, 2016], text classification [Tang *et al.*, 2015a] and community detection [Wang *et al.*, 2017].

However, using network representation models for item recommendations has not been fully studied. In this paper, we aim to solve the recommendation problem based on multi-source network representation methods. The work [Chen *et al.*, 2015] proposes to integrate network representations learned from the DeepWalk model into the factorization machines and obtains better recommendation results. By using the network embedding, the work [Zhao *et al.*, 2016] treats the recommendation problem as a cosine similarity calculation between representations of users and items.

In fact, previous studies only use network structure for representation learning, ignoring other types of auxiliary information. They also separate the learning of network embedding and recommendation into two independent steps. Zhang et al. [Zhang *et al.*, 2017] leverage the heterogeneous information to improve the recommendation results. However, the types of multi-source item information used in our work are different from their study. We further consider the tag information of items to capture the tag-content correspondence between item tag and item content. The work [Zhang *et al.*, 2016] maps items to entries in the knowledge base, using existing deep learning method to learn item semantics, while we map items to nodes in the network using network representation learning based on the DeepWalk model.

7 Conclusion

In this paper, we develop a new network representation learning for item recommendations. We consider multiple information sources, *i.e.,* item structure, textual content and tag information and present a new method CDNE that integrates collaborative filtering with deep network representations of items for recommendation. Compared with the baseline methods, CDNE obtains the better experimental results in terms of the Precision and MRR metrics. Therefore, by exploiting deep network embedding of items obtained from multi-source heterogeneous information and user-item interactions, our method can be used to boost the recommendation performance.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments and suggestions. This work was supported by the National Key Research and Development Program of China (No.2017YFB0803300), the NSFC (No.61502479), the MQNS Grant (No.9201701203), the MQ Enterprise Partnership Scheme Pilot Res Grant (No. 9201701455), and the Youth Innovation Promotion Association CAS (No. 2017210). C. Zhou is the corresponding author.

References

[Chang *et al.*, 2015] Shiyu Chang, Wei Han, Jiliang Tang, Guo Jun Qi, Charu C. Aggarwal, and Thomas S. Huang. Heterogeneous network embedding via deep architectures. In *KDD*, pages 119–128, 2015.

[Chen *et al.*, 2015] Chih-Ming Chen, Po-Chuan Chien, Yu-Ching Lin, Ming-Feng Tsai, and Yi-Hsuan Yang. Exploiting latent social listening representations for music recommendations. In *RecSys Poster*, 2015.

[Dong *et al.*, 2017] Xin Dong, Lei Yu, Zhonghuo Wu, Yuxia Sun, Lingfeng Yuan, and Fangxi Zhang. A hybrid collaborative filtering model with deep structure for recommender systems. In *AAAI*, pages 1309–1315, 2017.

[Gao *et al.*, 2016] Li Gao, Jia Wu, Zhi Qiao, Chuan Zhou, Hong Yang, and Yue Hu. Collaborative social group influence for event recommendation. In *CIKM*, pages 1941–1944, 2016.

[Gao *et al.*, 2017] Li Gao, Jia Wu, Chuan Zhou, and Yue Hu. Collaborative dynamic sparse topic regression with user profile evolution for item recommendation. In *AAAI*, pages 1316–1322, 2017.

[Grover and Leskovec, 2016] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *KDD*, pages 855–864, 2016.

[Le and Mikolov, 2014] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *ICML*, pages 1188–1196, 2014.

[Liu *et al.*, 2018] Chun-Yi Liu, Chuan Zhou, Jia Wu, Yue Hu, and Li Guo. Social recommendation with an essential preference space. In *AAAI*, pages 346–353, 2018.

[Liu, 2015] Xin Liu. Modeling users' dynamic preference for personalized recommendation. In *IJCAI*, pages 1785–1791, 2015.

[Mikolov *et al.*, 2013a] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[Mikolov *et al.*, 2013b] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.

[Morin and Bengio, 2005] Frederic Morin and Yoshua Bengio. Hierarchical probabilistic neural network language model. In *AISTATS*, 2005.

[Pan *et al.*, 2016] Shirui Pan, Jia Wu, Xingquan Zhu, Chengqi Zhang, and Yang Wang. Tri-party deep network representation. In *IJCAI*, pages 1895–1901, 2016.

[Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: online learning of social representations. In *KDD*, pages 701–710, 2014.

[Qiao *et al.*, 2014] Zhi Qiao, Peng Zhang, Yanan Cao, Chuan Zhou, Li Guo, and Binxing Fang. Combining heterogenous social and geographical information for event recommendation. In *AAAI*, pages 145–151, 2014.

[Ribeiro *et al.*, 2017] Leonardo FR Ribeiro, Pedro HP Saverese, and Daniel R Figueiredo. struc2vec: Learning node representations from structural identity. In *KDD*, pages 385–394, 2017.

[Salakhutdinov and Mnih, 2007] Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. In *NIPS*, pages 1257–1264, 2007.

[Tang *et al.*, 2015a] Jian Tang, Meng Qu, and Qiaozhu Mei. PTE: predictive text embedding through large-scale heterogeneous text networks. In *KDD*, pages 1165–1174, 2015.

[Tang *et al.*, 2015b] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. LINE: large-scale information network embedding. In *WWW*, pages 1067–1077, 2015.

[Wang and Blei, 2011] Chong Wang and David M. Blei. Collaborative topic modeling for recommending scientific articles. In *KDD*, pages 448–456, 2011.

[Wang *et al.*, 2013] Hao Wang, Binyi Chen, and Wu-Jun Li. Collaborative topic regression with social regularization for tag recommendation. In *IJCAI*, 2013.

[Wang *et al.*, 2016] Daixin Wang, Peng Cui, and Wenwu Zhu. Structural deep network embedding. In *KDD*, pages 1225–1234, 2016.

[Wang *et al.*, 2017] Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. Community preserving network embedding. In *AAAI*, pages 203–209, 2017.

[Yamasaki *et al.*, 2017] Toshihiko Yamasaki, Jiani Hu, Shumpei Sano, and Kiyoharu Aizawa. Folkpopularityrank: Tag recommendation for enhancing social popularity using text tags in content sharing services. In *IJCAI*, pages 3231–3237, 2017.

[Zhang *et al.*, 2016] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. Collaborative knowledge base embedding for recommender systems. In *KDD*, pages 353–362, 2016.

[Zhang *et al.*, 2017] Yongfeng Zhang, Qingyao Ai, Xu Chen, and W. Bruce Croft. Joint representation learning for top-n recommendation with heterogeneous information sources. In *CIKM*, pages 1449–1458, 2017.

[Zhao *et al.*, 2016] Wayne Xin Zhao, Jin Huang, and Ji-Rong Wen. Learning distributed representations for recommender systems with a network embedding approach. In *AIRS*, pages 224–236, 2016.