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
School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
††Department of Computing, Macquarie University, Sydney, Australia
§Data Science Lab, JD.com, 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-
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 $s = \{v_1, \ldots, v_n\}$ is considered as a sentence and each node $v_j$ is regarded as a word in neural language models. Assume that $a_j = \{v_{j-c}, \ldots, v_{j+c}\} \backslash v_j$ is the context vertices when given the target node $v_j$. DeepWalk aims to maximize the following objective

$$
\sum_{s \in S} \sum_{j = c}^{n-c} \sum_{v_k \in a_j} \ln P(v_k | v_j)
$$

(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 $\mathbf{\theta}_j$ 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 $\mathbf{\theta}_j$ 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 $\mathbf{\theta}_j$ 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 $\mathbf{\theta}_j$ and the output is the deep representations of its context items $a_j = \{v_{j-c}, \ldots, v_{j+c}\} \backslash v_j$. The objective function can be formulated as follows

$$
\mathcal{L}_s = \sum_{s \in S} \sum_{j = c}^{n-c} \sum_{v_k \in a_j} \ln P(v_k | v_j)
$$

(2)
The second neural network models the textual content $d_j$, for which the input is $\theta_j$ and $l_j$, and the output is the latent vectors of words in $d_j$, $l_j^i$ denotes the latent tag representation vector for $c_j$. Assume that $w_j = \{w_{j}^{1}, \ldots, w_{j}^{j}\}$ is a sequence of textual words within a contextual window. The objective aims to maximize the following likelihood function

$$L_t = \sum_{j=1}^{n} \ln P(w_j | v_j) + \sum_{j=1}^{n} \ln P(w_j | c_j)$$

$$= \sum_{j=1}^{n} \sum_{w_k \in w_j} \ln P(w_k | v_j) + \sum_{j=1}^{n} \sum_{w_k \in w_j} \ln P(w_k | c_j)$$

(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

$$L_{st} = \sigma'_t L_s + \sigma'_t L_t$$

(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 $P(w_k | v_j)$ in Eq. (2) can be calculated by using the softmax function as follows

$$P(w_k | v_j) = \frac{\exp(o_{w_j}^T o_{v_j})}{\sum_{i=1}^{n} \exp(o_{v_j}^T o_{v_i})}$$

(5)

where $o_{w_j}$ and $o_{v_j}$ are the input and output vector representation of item $v_j$. $P(w_k | v_j)$ and $P(w_k | c_j)$ in Eq. (3) can be readily calculated using softmax function as in Eq. (5). After training the neural network model, the input vector $o_{v_j}$ can be used as the deep representation vector $\theta_j$ 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 $U_i \in \mathbb{R}^K$ and item $v_j$ by a latent factor vector $V_j \in \mathbb{R}^K$. The user-item interactions can be formulated as

$$R_{ij} \sim \mathcal{N}(U_i^T V_j, \sigma_{ij}^2)$$

(6)

where the variable $\sigma_{ij}$ serves as a confidence parameter for rating $R_{ij}$. We set $\sigma_{ij} = a$, if $u_i$ has rated $vj$; 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 $\epsilon_j \in \mathbb{R}^K$, $\epsilon_j \sim \mathcal{N}(0, \sigma^{-1}_j I)$, to offset the deep network embedding $\theta_j$ 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 $V_j$ is formulated as

$$V_j = \theta_j + \epsilon_j$$

(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 $U_i \sim \mathcal{N}(0, \sigma_{ui}^{-1} I)$.
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 $P(a_j | v_j)$.
   (b) For each item $v_j$ with its textual content $d_j$, draw from the probability $P(w_j | v_j)$.
   (c) For each item $v_j$ with its tag information $c_j$, draw from the probability $P(w_j | c_j)$.
3. For item $v_j$, draw an item latent offset $\epsilon_j \sim \mathcal{N}(0, \sigma_{v_j}^{-1} I)$ and set $V_j = \theta_j + \epsilon_j$.
4. For each user-item pair $(u_i, v_j)$, draw the rating $R_{ij} \sim \mathcal{N}(U_i^T V_j, \sigma_{ij}^2)$.

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 $U, V, \theta$ and $I$ is equivalent to minimizing the complete negative log-likelihood as follows

$$\min_{U, V, \theta, I} \sum_{i=1}^{n} \sum_{j=1}^{m} \sigma_{ij}(R_{ij} - U_i^T V_j)^2$$

$$-\sigma_j \sum_{s \in S} \sum_{s \notin \epsilon_j} \ln P(v_k | v_j) + \sum_{i=1}^{m} \sigma_{ui} ||U_i||^2$$

$$+ \sum_{j=1}^{n} \sigma_{v_j} ||V_j - \theta_j||^2 - \sigma_{i} \sum_{j=1}^{n} \sum_{w_k \in w_j} \ln P(w_k | v_j)$$

$$-\sigma_{i} \sum_{j=1}^{n} \sum_{w_k \in w_j} \ln P(w_k | c_j)$$

(8)

where $\sigma_{ij}, \sigma_{v_j}, \sigma_{ui}, \sigma_{v_j}$ and $\sigma_{i}$ 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 $P(w_k | v_j)$ ($P(w_k | v_j)$ or $P(w_k | c_j)$ are the same) is calculated by the softmax function. To reduce the computation cost of the gradient of $P(w_k | v_j)$, $P(w_k | v_j)$ or $P(w_k | c_j)$, we instead use the hierarchical softmax [Morin and Bengio, 2005] to approximate the probability distribution.

Specifically, for the calculation of $P(w_k | v_j)$, 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 $P(w_k | v_j)$ or $P(w_k | c_j)$, 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, $f_0, f_1, \ldots, f_m$. $P(v_k | v_j)$ is then calculated by the probability of the specific path. We have

$$P(v_k | v_j) = \prod_{i=1}^{m} P(f_i | v_j)$$

(9)

where $P(f_i | v_j)$ is defined as $P(f_i | v_j) = 1/(1 + e^{-\sigma_{f_i}^T o_{f_i}^T})$, $o_{f_i} \in \mathbb{R}^K$ denotes the representation assigned to tree node
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 \( L \) as the objective. The model parameter set becomes \( \Theta = \{U, V, \Psi\} \), where \( \Psi \) denotes the representations assigned to the interval nodes of the binary tree. In each iteration, we use the gradient to update each parameter in \( \Theta \) as follows

\[
\Theta_t = \Theta_t - \alpha_t \frac{\partial L}{\partial \Theta_t}
\]

(10)

where \( \alpha_t \) 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 \( U_i^T V_1 \geq U_i^T V_2 \geq \cdots \geq U_i^T V_h \).

## 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 \( a = 1, b = 0.01, \sigma_s = 1, \sigma_t = 0.5, \sigma_a = 0.1, \sigma_v = 1 \). The learning rate \( \alpha \) 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
Figure 2: Precision and MRR performance comparisons with respect to different recommendation list sizes.

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

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

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 intelli-
gence) the article belongs to. CDNE-tc and CDNE-ta outper-
forms 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 pos-
sible 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 infor-
mation. The method NERM aims to separately learn network
embedding on the bipartite adoption graph and make item
recommendations with these embeddings, ignoring the inter-
actions 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 recom-
mendation 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%, 61.32%, 54.82%, 20.03%, 30.31% and 25.00% 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, tex-
tual content and tag information. From this point of view, the
network embedding can be represented as a latent topic dis-
tribution, 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 re-
commend 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 in-
fornation, 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 re-
sults. The work [Wang and Blei, 2011] proposed a collab-
orative topic regression model which combines traditional col-
laborative 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 col-
laborative filtering from the rating matrix.

Hybrid recommendation models aim to learn latent fac-
tors of users and items from user-item interactions and aux-
iliary 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] at-
tracts increased attention. This type of method aims to learn
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.

<table>
<thead>
<tr>
<th>CDNE:</th>
<th>PMF:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Item-based collaborative filtering recommendation algorithms</td>
<td>1. Effective missing data prediction for collaborative filtering</td>
</tr>
<tr>
<td>2. Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness</td>
<td>2. GroupLens: applying collaborative filtering to Usenet news</td>
</tr>
<tr>
<td>4. Incorporating contextual information in recommender systems using a multidimensional approach</td>
<td>4. An efficient boosting algorithm for combining preferences</td>
</tr>
<tr>
<td>5. Factorization meets the neighborhood: a multifaceted collaborative filtering model</td>
<td>5. Incorporating contextual information in recommender systems using a multidimensional approach</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CTR:</th>
<th>SLG:</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Analysis of recommendation algorithms for e-commerce</td>
<td>2. Information Filtering: Overview of Issues, Research and Systems</td>
</tr>
<tr>
<td>3. Item-based collaborative filtering recommendation algorithms</td>
<td>3. GroupLens: applying collaborative filtering to Usenet news</td>
</tr>
<tr>
<td>4. Effective missing data prediction for collaborative filtering</td>
<td>4. Analysis of recommendation algorithms for e-commerce</td>
</tr>
<tr>
<td>5. Amazon.com recommendations: item-to-item collaborative filtering</td>
<td>5. Methods and metrics for cold-start recommendations</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NERM:</th>
<th>CDNE-te:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Item-based collaborative filtering recommendation algorithms</td>
<td>1. Item-based collaborative filtering recommendation algorithms</td>
</tr>
<tr>
<td>2. An efficient boosting algorithm for combining preferences</td>
<td>2. Effective missing data prediction for collaborative filtering</td>
</tr>
<tr>
<td>3. Information Filtering: Overview of Issues, Research and Systems</td>
<td>3. Analysis of recommendation algorithms for e-commerce</td>
</tr>
<tr>
<td>4. Effective missing data prediction for collaborative filtering</td>
<td>4. Amazon.com recommendations: item-to-item collaborative filtering</td>
</tr>
<tr>
<td>5. How oversight improves member-maintained communities</td>
<td>5. How much can behavioral targeting help online advertising?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CDNE-tc:</th>
<th>CDNE-ta:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness</td>
<td>1. Methods and metrics for cold-start recommendations</td>
</tr>
<tr>
<td>2. Factorization meets the neighborhood: a multifaceted collaborative filtering model</td>
<td>2. Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness</td>
</tr>
<tr>
<td>3. GroupLens: applying collaborative filtering to Usenet news</td>
<td>3. Item-based collaborative filtering recommendation algorithms</td>
</tr>
<tr>
<td>4. Effective missing data prediction for collaborative filtering</td>
<td>4. Information Filtering: Overview of Issues, Research and Systems</td>
</tr>
<tr>
<td>5. Item-based collaborative filtering recommendation algorithms</td>
<td>5. Large-scale behavioral targeting</td>
</tr>
</tbody>
</table>

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.

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


