Recommendation in Heterogeneous Information Networks Based on Generalized Random Walk Model and Bayesian Personalized Ranking

Zhengshen Jiang
Peking University
Beijing, China
jiangzhengshen@pku.edu.cn

Hongzhi Liu*
Peking University
Beijing, China
liuhz@pku.edu.cn

Bin Fu
Peking University
Beijing, China
fubin1990@pku.edu.cn

Zhonghai Wu*
Peking University
Beijing, China
wuzh@pku.edu.cn

Tao Zhang
Beijing Huapinborui Network
Technology Company
Beijing, China
kylen.zhang@kanzhun.com

ABSTRACT

Recommendation based on heterogeneous information network (HIN) is attracting more and more attention due to its ability to emulate collaborative filtering, content-based filtering, context-aware recommendation and combinations of any of these recommendation semantics. Random walk based methods are usually used to mine the paths, weigh the paths, and compute the closeness or relevance between two nodes in a HIN. A key for the success of these methods is how to properly set the weights of links in a HIN. In existing methods, the weights of links are mostly set heuristically. In this paper, we propose a Bayesian Personalized Ranking (BPR) based machine learning method, called HeteLearn, to learn the weights of links in a HIN. In order to model user preferences for personalized recommendation, we also propose a generalized random walk with restart model on HINs. We evaluate the proposed method in a personalized recommendation task and a tag recommendation task. Experimental results show that our method performs significantly better than both the traditional collaborative filtering and the state-of-the-art HIN-based recommendation methods.

CCS CONCEPTS

• **Information systems → Recommender systems**; *Personalization*; • **Computing methodologies → Ranking**;

KEYWORDS

recommender systems; Bayesian Personalized Ranking; heterogeneous information network; random walk

*Corresponding author.

WSDM 2018, February 5–9, 2018, Marina Del Rey, CA, USA

ACM ISBN 978-1-4503-5581-0/18/02…$15.00
https://doi.org/10.1145/3159652.3159715

ACM Reference Format:
Zhengshen Jiang, Hongzhi Liu, Bin Fu, Zhonghai Wu*, and Tao Zhang. 2018. Recommendation in Heterogeneous Information Networks Based on Generalized Random Walk Model and Bayesian Personalized Ranking. In *WSDM 2018: WSDM 2018: The Eleventh ACM International Conference on Web Search and Data Mining , February 5–9, 2018, Marina Del Rey, CA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3159652.3159715

1 INTRODUCTION

1.1 Background

Recommender systems have been used to provide users with recommendations for products or services in various domains, such as e-commerce platforms and social networks [28]. Traditional recommendation approaches include [16]:

- Content-based filtering (CB): These approaches utilize the content information to recommend items that are similar to those previously preferred by the target user.
- Collaborative filtering (CF): These approaches recommend items to users based on the preferences that other users have expressed for those items [5].
- Hybrid approaches: These approaches combine collaborative with content-based methods or with different variants of other collaborative methods.

As a kind of hybrid approach, recommendations based on heterogeneous information networks (HINs) provide a new perspective to design recommendation systems. HINs are logical networks involving multiple-typed objects and multiple-typed links denoting different relations [8].

For instance, Figure 1 illustrates an example of a heterogeneous information network, which contains five types of entities: users, movies, genres, actors and directors, together with their relations.

Since these approaches combine user feedback data with additional information such as items or users attributes and relationships, they could emulate collaborative-filtering, content-based filtering, context-aware recommendation and combinations of any of these recommendation semantics [13].

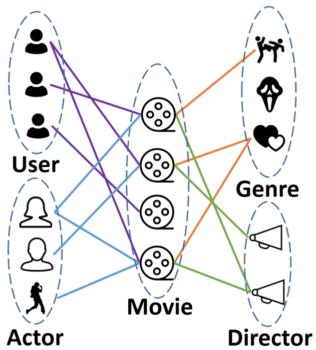

Figure 1: An example of a heterogeneous information network

1.2 Related Work

In this paper, we focus on recommendation based on HINs. Several algorithms have been presented in this field, and most of them are random walk based methods. In this subsection, we will first introduce the random walk model briefly, and then give a review on related work.

Consider a random walker that recursively moves to a random neighbour in a network. Denoting by $r_i^{(t)}$ the probability that the walker is located at node i at time t, and $r_i^{(t)}$ of all the nodes form a vector $\mathbf{r}^{(t)}$, we can write

$$\mathbf{r}^{(t+1)} = \mathbf{P} \cdot \mathbf{r}^{(t)}$$

where $\mathbf{P}$ is the transition probability matrix. The stationary probability $\mathbf{r}$ could be given by the equation $\mathbf{r} = \mathbf{P} \cdot \mathbf{r}$.

A lot of work is in close relationship with this model, such as the Local Random Walk [15], the Path-constrained random walk [11], HeteSim [23] and PathMining [12].

One important variant of random walk is the random walk with restart model. Considering a random walker starting from node i that recursively moves to a random neighbour with probability α and returns to node i with probability $1 - \alpha$, we can write the iterative equation as following

$$\mathbf{r}^{(t+1)} = \alpha \mathbf{P} \cdot \mathbf{r}^{(t)} + (1 - \alpha)\mathbf{t} \tag{1}$$

where $\mathbf{t}$ is called the "teleport vector", and it is usually set as $\mathbf{t} = \mathbf{r}^{(0)}$, i.e., the elements of $\mathbf{t}$ are all zero except that $t_i = 1$.

This kind of model was traditionally used in PageRank [18], and was introduced to recommender systems by Personalized PageRank [9]. Typical work include ObjectRank [1], ItemRank[7] and the HIN-based method PathRank [13].

As a representative recommendation method based on HINs, the PathRank model introduced the meta-paths on HINs into random walk with restart model. The authors proved that the PageRank, Personalized PageRank and Path-constrained Random Walk model are all special cases of PathRank.

One key problem in random walk is the setting of the transition matrix $\mathbf{P}$. In traditional random walk models, $\mathbf{P} = \mathbf{A}\mathbf{D}^{-1}$, where $\mathbf{A}$ is the adjacency matrix, and $\mathbf{D}$ is the degree matrix defined as $\mathbf{D} = \text{diag}(d_i)$ with d_i being the degree of the node i. According to physical models like resistance distance [10] and average commute time [6], the transition matrix could also be set by $\mathbf{P} = \mathbf{D}^{-1}\mathbf{A}$. In the work of [27], models using $\mathbf{P} = \mathbf{A}\mathbf{D}^{-1}$ are called "probabilistic spreading", and those using $\mathbf{P} = \mathbf{D}^{-1}\mathbf{A}$ are called "heat spreading".

In previous work, the transition matrix was mostly set heuristically, and recommendation results could only be provided by the topological relations of the network. As a result, the transition matrix could not adapt to different datasets automatically. In this work, we use a machine learning method to learn the transition matrix. We use a generalized random walk with restart method to model user preferences, and use a Bayesian Personalized Ranking (BPR) based machine learning method to learn the link weights of HINs. Experimental results show that the proposed method performs significantly better than both the traditional collaborative filtering and HIN-based recommendation methods.

1.3 Contributions

In this paper, we contribute to the field of recommender system both theoretically and empirically.

Firstly, we propose a new personalized recommendation method, called HeteLearn, on heterogeneous information networks. In this method, we use a BPR-based machine learning method to learn the weights of links in the networks. Instead of setting the weights of links or path-guides heuristically as in previous work, the proposed method could automatically learn them from the HINs. Therefore, the proposed method has a wider range of applications.

Secondly, we propose a generalized random walk with restart model to model user preferences on HINs. We also prove that, the result of random walk with restart is the accumulation of a series of random walks that consist of different random steps, and longer random walks have less impacts on the final results. In addition, the semantics of longer paths are less clear than short paths. Thus, we propose to use k-step random walk with restart model in our recommendation method.

Lastly, empirically, we demonstrate the effectiveness of our method in both item recommendation and tag recommendation tasks. According to the experimental results, the proposed method performs significantly better than both the traditional collaborative filtering and the state-of-the-art HIN-based recommendation methods.

2 PRELIMINARIES AND NOTATIONS

2.1 Heterogeneous Information Network

Following is the definition of heterogeneous information network given by Sun et al. [24].

Definition 2.1. **(Heterogeneous Information Network [24])** An information network is a directed graph $G = (V, E)$ with an object type mapping function $\phi : V \longrightarrow \mathcal{A}$ and a link type mapping function $\psi : E \longrightarrow \mathcal{R}$, where each object $v \in V$ belongs to one particular object type $\phi(v) \in \mathcal{A}$, and each link $e \in E$ belongs to a particular relation $\psi(e) \in \mathcal{R}$. When the types of objects satisfy $|\mathcal{A}| > 1$ or the types of relations satisfy $|\mathcal{R}| > 1$, the network is a **heterogeneous information network**; otherwise, it is a **homogeneous information network**.

In an information network, heterogeneous or homogeneous, each node $v_i \in V$ may have a set of attribute-value pairs which describe the properties of the node, and each edge $e(v_i, v_j) \in E$ has a corresponding weight w.

The network schema or meta-graph provides meta level description of a HIN. The definition is as follows.

Definition 2.2. **(Network Schema [24])** The network schema is a meta template for a heterogeneous network $G = (V, E)$ with the object type mapping $\phi : V \to \mathcal{A}$ and the link mapping $\psi : E \to \mathcal{R}$, which is a directed graph defined over object types $\mathcal{A}$, with edges as relations from $\mathcal{R}$, denoted as $T_G = (\mathcal{A}, \mathcal{R})$.

2.2 Implicit Feedback

With u and i denoting a user and an item, respectively, we define the user implicit feedback matrix R as following:

$$R_{ui} = \begin{cases} 1, & \text{if } (u, i) \text{ interaction is observed;} \\ 0, & \text{otherwise.} \end{cases}$$

We use U and I to denote the set of users and items, respectively. For each user $u \in U$, the items that satisfy $R_{ui} = 1$ are called positive items, and we denote the set of positive items by $I_u^+ \subset I$. And items in $I \backslash I_u^+$ are called negative items.

The goal is to recommend each user a personalized ranking list of items from $I \backslash I_u^+$. This problem is usually called one-class recommendation problem [19] or recommendation with implicit feedback [20, 22].

2.3 Bayesian Personalized Ranking (BPR)

In the BPR method, Rendle et al. [22] used a binary random variable $\delta((u, i) \succ (u, j))$ to denote whether user u prefers item i to item j or not. The function $\delta(z)$ equals 1 if z is true, and equals 0 otherwise. This representation is called a user's pairwise preference [20].

For a given user u, in order to calculate the overall likelihood of pairwise preferences among all the items, Rendle et al. [22] used the Bernoulli distribution over the binary random variable $\delta((u, i) \succ (u, j))$, as following

$$\begin{aligned} \text{BPR} &= \ln\{ \prod_{(u,i,j) \in U \times I \times I} p(\hat{r}_{ui} > \hat{r}_{uj})^{\delta((u,i) \succ (u,j))} \\ &\quad \times [1 - p(\hat{r}_{ui} > \hat{r}_{uj})]^{[1 - \delta((u,i) \succ (u,j))]} \} \\ &= \ln\{ \prod_{(u,i) \succ (u,j)} p(\hat{r}_{ui} > \hat{r}_{uj}) [1 - p(\hat{r}_{uj} > \hat{r}_{ui})] \} \\ &\propto \sum_{(u,i) \succ (u,j)} \ln p(\hat{r}_{ui} > \hat{r}_{uj}) \end{aligned}$$

where the summation is computed over all the pairs that satisfy $(u, i) \succ (u, j)$ in the training set, and $\hat{r}_{ui}$ represents the preference of user u for item i.

In the BPR method, it was defined that

$$\hat{x}_{uij} = \hat{r}_{ui} - \hat{r}_{uj}$$

and

$$p(\hat{r}_{ui} > \hat{r}_{uj}) = \frac{1}{1 + e^{-\hat{x}_{uij}}}.$$

Therefore, the gradient of BPR is

$$\frac{\partial \text{BPR}}{\partial \Theta} \propto \sum_{(u,i) \succ (u,j)} \frac{1}{1 + e^{\hat{x}_{uij}}} \cdot \frac{\partial}{\partial \Theta} \hat{x}_{uij}$$

BPR could be maximized using gradient descent as following:

$$\Theta \leftarrow \Theta + \eta \frac{\partial \text{BPR}}{\partial \Theta}$$

where η is the learning rate. When using stochastic gradient descent, Θ is updated as following:

$$\Theta \leftarrow \Theta + \eta \cdot \frac{1}{1 + e^{\hat{x}_{uij}}} \cdot \frac{\partial}{\partial \Theta} \hat{x}_{uij} \tag{2}$$

3 RECOMMENDATION ON HINS

In conventional random walk models, the transition matrix is set by $\mathbf{P} = \mathbf{A}\mathbf{D}^{-1}$. However, when using machine learning methods to learn the link weights in a network, such models will result in a non-convex target function, which could not be optimized using common methods like gradient descent.

In this section, we will propose a generalized random walk with restart model on HINs to model user preferences and facilitate the optimization. The learning method will be detailed in Section 4.

3.1 The Heterogeneous Random Walk with Restart Model

Following is the mathematical definition of our model.

Definition 3.1. **(Heterogeneous Random Walk with Restart)** Given a heterogeneous information network $G = (V, E)$ and its meta-graph $T_G = (\mathcal{A}, \mathcal{R})$, consider a random walker starting from node i that recursively moves to a random neighbour with probability α and returns to node i with probability $1 - \alpha$. Denote $\mathbf{P}$ to be the transition probability matrix, and $\mathbf{r}^{(t)}$ to be the distribution probability vectors representing the probabilities that the nodes are visited at time t. And denote the initial value of $\mathbf{r}$ to be $\mathbf{r}^{(0)}$, and satisfies that $r_i^{(0)} \geq 0$ and $\sum_i r_i^{(0)} = 1$. During each iteration, the vector $\mathbf{r}$ is updated according to the following equation

$$\mathbf{r}^{(t+1)} = \alpha \mathbf{P} \mathbf{r}^{(t)} + (1 - \alpha) \mathbf{t} \tag{3}$$

where $\mathbf{t}$ is called the "teleport vector" and is typically set to be $\mathbf{t} = \mathbf{r}^{(0)}$.

In this paper, we model the transition matrix $\mathbf{P}$ as

$$\mathbf{P}_{ij} = p(v_i | v_j) = p(\phi(v_i) | v_j) \cdot p(v_i | v_j, \phi(v_i))$$

where $\phi(v_i)$ is the type of v_i.

Further, we define

$$p(v_i | v_j, \phi(v_i)) = \frac{1}{|D_j^{\phi_i}|}$$

where

$$D_j^{\phi_i} \stackrel{def}{=} \{ v_k | e(v_j, v_k) \in E \land \phi(v_k) = \phi(v_i) \}$$

denotes the set of nodes that is connected to node v_j and the type is $\phi(v_i)$. According to this definition, the probability that node v_j goes to node v_i is inversely proportional to the number of neighbours of v_j that are of the same type as node v_i. In other words, node v_j transfers to all the nodes of type $\phi(v_i)$ with equal probability.

To model the distribution $p(\phi(v_i)|v_j)$, we assume that

$$p(\phi(v_i)|v_j) = p(\phi(v_i)|\phi(v_j)) \tag{4}$$

In other words, we assume that all the nodes of type $\phi(v_j)$ transfer to type $\phi(v_i)$ with equal probabilities. Since the parameters $p(\phi(v_i)|\phi(v_j))$ should be learned from the training data, such assumption greatly reduces the number of parameters to be learned, and makes the learning procedure more robust. Detailed learning procedure will be given in Section 4.

In this paper, we set the initial distribution $\mathbf{r}^{(0)}$ to be different for different user u as following

$$\mathbf{r}_i^{(0)} = \begin{cases} 1, & \text{node } v_i \text{ corresponds to user } u, \\ 0, & \text{otherwise.} \end{cases}$$

Therefore, our model could make personalized recommendations.

3.2 Finite Step Random Walk with Restart

The random walk with restart model can be proved to have a stable solution. However, in practice, it is usually costly to compute the stable solution. In this subsection, we will give some analysis on the random walk with restart model, and prove that it is usually not necessary to iterate the model until convergence. Thus, in this paper, finite step random walk with restart model is used instead.

Using the iterative equation in Equation 3, after substituting $\mathbf{t} = \mathbf{r}^{(0)}$, we get

$$\begin{aligned} \mathbf{r}^{(1)} &= \alpha\mathbf{P}\mathbf{r}^{(0)} + (1-\alpha)\mathbf{r}^{(0)} \\ \mathbf{r}^{(2)} &= \alpha^2\mathbf{P}^2\mathbf{r}^{(0)} + \alpha(1-\alpha)\mathbf{P}\mathbf{r}^{(0)} + (1-\alpha)\mathbf{r}^{(0)} \\ &\cdots \\ \mathbf{r}^{(k)} &= \alpha^k\mathbf{P}^k\mathbf{r}^{(0)} + \sum_{i=0}^{k-1}\alpha^i(1-\alpha)\mathbf{P}^i\mathbf{r}^{(0)} \end{aligned} \tag{5}$$

Note that $\mathbf{P}^i\mathbf{r}^{(0)}$ corresponds to a random walk that consists of i random steps. Therefore, the result of k-step random walk with restart is the accumulation of a series of random walks that consist of different random steps, and the weights of these random walks decrease exponentially with the increasing of random steps. So the random walks with more random steps have less impacts on the results.

In addition, the semantics of longer paths are less clear than short paths, and the correlation between the nodes connected through longer paths are also less reliable. Thus, using long paths in the learning procedure may result in bad generalization performance.

Therefore, in this paper, we use k-step random walk with restart model in both the training and recommendation procedure.

4 LEARNING OF THE WEIGHT MATRIX

In this section, we will propose a BPR-based algorithm (HeteLearn) to learn the transition matrix $\mathbf{P}$ on HINs.

4.1 Objective Function

According to the assumption in Equation 4, the parameters to be learned are the probabilities $p(\phi(v_i)|\phi(v_j))$, and the total number of the parameters equals the number of link types in network G, i.e., the number of edges in the network schema T_G. Hence, in this subsection, we also use p_τ to denote the parameters, where $\tau \in \mathcal{R}$ is an edge type in network G.

We follow the BPR optimization framework to construct our objective function. The idea is that the model with appropriate parameters should rank the positive items higher than negative items, i.e., positive items should have higher probabilities than negative ones. To achieve this, we design the following objective function to be maximized

$$\underset{\Theta}{\text{argmax}} \quad Obj = \sum_{(u,i,j)\in D_S} \ln\sigma(\hat{r}_{ui} - \hat{r}_{uj}) \tag{6}$$

where

$$D_S \overset{def}{=} \{(u,i,j) | i \in I_u^+ \wedge j \in I \backslash I_u^+\}$$

and

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

is the sigmoid function.

4.2 Updating Rule

The objective function could be optimized using stochastic gradient descent, and the update rule is:

$$\Theta \leftarrow \Theta + \eta \cdot \frac{1}{1+e^{\hat{r}_{ui}-\hat{r}_{uj}}} \cdot \left(\frac{\partial}{\partial\Theta}\hat{r}_{ui} - \frac{\partial}{\partial\Theta}\hat{r}_{uj}\right) \tag{7}$$

where η is the learning rate.

We use the heterogeneous random walk with restart model to estimate $\hat{r}$. Using Equation 5, taking $k = 3$ as an example, we could get

$$\hat{r}_{ui} = \alpha^3(\mathbf{P}^3)_{iu} + \alpha^2(1-\alpha)(\mathbf{P}^2)_{iu} + \alpha(1-\alpha)\mathbf{P}_{iu}$$

and

$$\begin{aligned} (\mathbf{P}^3)_{iu} &= \sum_{v_p, v_q \in V} p(v_i|v_p)\cdot p(v_p|v_q)\cdot p(v_q|v_u) \\ (\mathbf{P}^2)_{iu} &= \sum_{v_p\in V} p(v_i|v_p)\cdot p(v_p|v_u). \end{aligned}$$

Considering our model of the transition matrix and our assumption, we get

$$\mathbf{P}_{ij} = p(v_i|v_j) = p(\phi(v_i)|\phi(v_j)) \cdot \frac{1}{|D_j^{\phi_i}|}$$

We use $\psi(e(v_j, v_i))$ to denote the type of the edge from v_j to v_i. Taking all these equations into consideration, the gradient with respect to an edge type τ is

$$\begin{aligned} \frac{\partial\hat{r}_{ui}}{\partial p_\tau} &= \alpha^3 \cdot \Big(\sum_{\psi(ip)=\tau} \frac{w_{pq}w_{qu}}{|D_p^{\phi_i}|} + \sum_{\psi(pq)=\tau} \frac{w_{ip}w_{qu}}{|D_q^{\phi_p}|} \\ &+ \sum_{\psi(qu)=\tau} \frac{w_{ip}w_{pq}}{|D_u^{\phi_q}|} \Big) + \alpha^2(1-\alpha)\cdot\Big(\sum_{\psi(ip)=\tau} \frac{w_{pu}}{|D_p^{\phi_i}|} \\ &+ \sum_{\psi(pu)=\tau} \frac{w_{ip}}{|D_u^{\phi_p}|} \Big) + \alpha(1-\alpha)\cdot\frac{\delta(\psi(iu)=\tau)}{|D_u^{\phi_i}|} \end{aligned} \tag{8}$$

where we abbreviate $p(v_i|v_j)$ to w_{ij} for simplicity, and $\psi(ip) = \tau$ is short for $\psi(e(v_p, v_i)) = \tau$.

After computing the gradients of $\hat{r}_{ui}$ and $\hat{r}_{uj}$, we could update the learning parameters p_τ using Equation 7. And the learning scheme is summarized in Algorithm 1. During the learning process, if any of the parameters becomes below zero after update, the learning rate is halved (lines 5-7). At the end of each iteration, the parameters are normalized so that the sum of all the parameters is 1 (line 9).

Algorithm 1 HeteLearn

Input: the heterogeneous information network $G = (V, E)$, the network schema $T_G = (\mathcal{A}, \mathcal{R})$, the training set D_S, the parameters α and k in Equation 5, the initial learning rate η

Output: the learned parameters $\Theta \overset{def}{=} (p_{\tau_1}, p_{\tau_2}, ..., p_{\tau_{|\mathcal{R}|}})^T$

1: randomly initialize Θ
2: **repeat**
3: randomly draw (u, i, j) from D_S
4: $grad_{\tau_\alpha} \leftarrow$ using Equation 7 to compute the gradients of p_{τ_α}, $\forall \alpha \in \{1, 2, ..., |\mathcal{R}|\}$
5: **while** exists τ_α that satisfies $p_{\tau_\alpha} + \eta \cdot grad_{\tau_\alpha} < 0$ **do**
6: $\eta \leftarrow \eta/2$
7: **end while**
8: $p_{\tau_\alpha} \leftarrow p_{\tau_\alpha} + \eta \cdot grad_{\tau_\alpha}, \forall \alpha \in \{1, 2, ..., |\mathcal{R}|\}$
9: $p_{\tau_\alpha} \leftarrow p_{\tau_\alpha} / \sum_\alpha p_{\tau_\alpha}$
10: **until** convergence
11: **return** $p_{\tau_1}, p_{\tau_2}, ..., p_{\tau_{|\mathcal{R}|}}$

5 EXPERIMENTS AND RESULTS

In this section, we demonstrate the effectiveness of our method on two kinds of recommendation tasks, i.e., the item recommendation task and the tag recommendation task. We use the the Restaurant & Consumer dataset and the HetRec 2011 MovieLens dataset on the item recommendation task, and use a variant of the HetRec 2011 MovieLens dataset on the tag recommendation task.

Following we will first introduce the used datasets in Section 5.1, and then give a brief description of the compared methods in Section 5.2. In section 5.3 we will show the effects of the parameters in HeteLearn. After that, the results of the compared methods on item recommendation and tag recommendation task are shown in section 5.4 and 5.5, respectively.

5.1 Datasets

We validated the proposed method using two datasets. The smaller one is the Restaurant & Consumer dataset from the UCI Machine Learning Repository [14]. And the bigger one is the HetRec 2011 MovieLens dataset published by GroupLens research group [1].

The Restaurant & Consumer dataset was collected during a seven months period. It contains the ratings of the restaurants from the users and information about the users and restaurants. Table 1 shows the statistics of this dataset.

The HetRec 2011 MovieLens dataset is an extension of MovieLens10M dataset, and it links the movies of MovieLens dataset with their corresponding web pages at Internet Movie Database (IMDb) [2] and Rotten Tomatoes movie review systems [3]. Table 2 is the statistics of the dataset.

[1]http://www.grouplens.org
[2]http://www.imdb.com
[3]http://www.rottentomatoes.com

Table 1: Statistics of the Restaurant & Consumer dataset.

Types	Total No.	Average No. (per Restaurant)	Average No. (per User)
User	138	-	-
Restaurant	934	-	-
Parking Lot	7	-	-
Payment	11	2.14	1.33
Cuisine	59	1.19	2.39
Rating	1161	8.93	8.41

Table 2: Statistics of the HetRec 2011 MovieLens dataset.

Types	Total No.	Average No. (per Movie)	Average No. (per User)
User	2113	-	-
Movie	10197	-	-
Genre	20	2.040	-
Director	4060	1.0	-
Actor	95321	22.778	-
Country	72	1.0	-
Tag	13222	8.117	22.696
Rating	855598	84.637	404.921

5.2 Baselines

We compared the performance of our method with the following state-of-the-art methods.

5.2.1 User-Based Collaborative Filtering (User CF). These approaches predict a user's interest in an item based on rating information from similar user profiles [26]. In our experiment, we took the non-normalized neighborhood collaborative filtering approach as representative. It is a modification of traditional k-nearest neighborhood collaborative filtering (KNN CF), and it is known to perform better than traditional KNN CF for top-k recommendation [13]. For a given user and a given item, the rating $r_{u,i}$ is defined as $r_{u,i} = \sum_{u' \in \hat{U}} sim(u, u') \times r_{u',i}$ where $\hat{U}$ is the set of n_k most similar users, and the $sim(u, u')$ is the similarity between two users [3]. In our experiments, the parameter n_k was selected by grid search, and the cosine coefficient was selected as the similarity measure.

5.2.2 Item-Based Collaborative Filtering (Item CF). These approaches apply the same idea with User CF, but use similarity between items instead of users. For a given user and a given item, the rating $r_{u,i}$ is defined as $r_{u,i} = \sum_{i' \in \hat{I}} sim(i, i') \times r_{u,i'}$ where $\hat{I}$ is the set of n_k most similar items. Similar to User CF, n_k was selected by grid search, and the cosine similarity was used.

5.2.3 Non-negative Matrix Factorization (NMF). As a representative matrix factorization method, NMF tries to find two non-negative matrix factors where the product of the two matrices is an approximation of the original matrix. This method was originally proposed for image analysis. However, it has been widely used in collaborative filtering recently [2, 17].

5.2.4 Latent Factor Model (LFM). Also known as SVD (Singular Value Decomposition) models. The key idea of these models is to factorize the user-item rating matrix to a product of two lower rank matrices, one containing the so-called "user factors", while the other containing the so-called "item-factors" [3].

5.2.5 Personalized PageRank (PRank). A graph-based node ranking method [9] based on the classic PageRank algorithm [18]. In this method, the stable solution of Equation 1 is used as the ratings of a given user for the nodes in a graph. And the stable solution could be achieved by iterating Equation 1 until convergence.

5.2.6 Path-constrained Random Walk (PCRW). Such methods are based on single paths in the heterogeneous information network. They are also called semantic recommendation since each path has its corresponding semantic [23]. For instance, the paths *UMUM* and *UMTM* can be interpreted as results of user-based collaborative filtering and content(tag)-based filtering, respectively. According to the semantics of the paths and previous work, 3-step paths are the most common and important. In our experiments, the paths with their length less than or equal to 3 were considered.

5.2.7 PathRank. A state-of-the-art heterogeneous information network based method [13]. In our experiments, we took the same settings as in [13]: $w_{restart} = 0.1, w_{trans} = 0.25, w_{path} = 0.65$. For the HetRec 2011 MovieLens dataset, the path-guide *UMTM +UMU(20)M +UMUM(20)* was used since it showed the best performance in the original paper of PathRank. For the Restaurant & Consumer dataset, since no tag information is available, the path-guide *UMU(20)M+UMUM(20)* was used instead.

5.2.8 HeteRS. A state-of-the-art HIN-based method [21]. The most significant difference between our method and HeteRS is the user preference model. HeteRS uses multivariate Markov chain to model user preferences, and uses the stable solution of the model.

As for the proposed method, HeteLearn, the initial learning rate η was set to be 0.5. The parameters k and α in the generalized random walk with restart model will be discussed in section 5.3.

In the following experiments, all of the not mentioned parameters of the compared methods were decided by grid search.

5.3 Effects of the Parameters

In this section, we will show the effects of two key parameters k and α in HeteLearn.

The performances with different k and α values in Equation 5 are plotted in Figure 2. The best performance is achieved at $k = 3$ or $k = 5$, and after that the performance decreases slightly with the increase of k, which confirmed the assumption that longer random walks have less impacts on the results, as is discussed in Section 3.2.

In the following experiments, we set k to be 3. Since the performance varies with different datasets and different values of α, we choose α to be 0.8 in the HeteLearn algorithm.

5.4 Task 1: Item Recommendation

In this section, we validate our proposed method on the item recommendation task using two datasets. The used datasets and the compared methods have been shown and discussed in section 5.1 and 5.2, respectively.

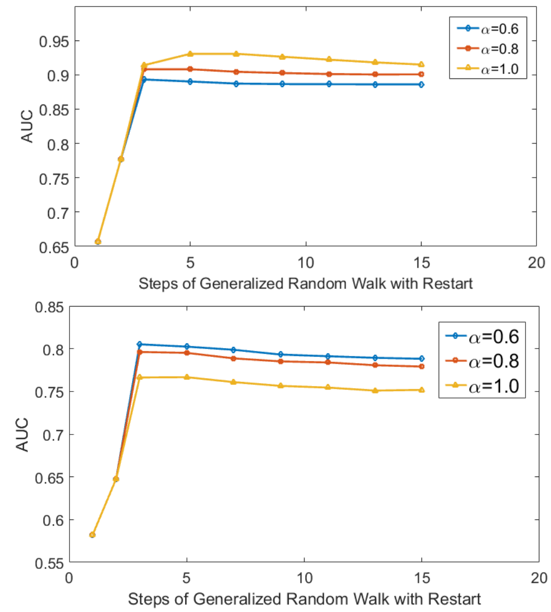

Figure 2: Performances of the generalized random walk with restart model with different steps k and different α values on the Restaurant & Consumer dataset (the upper plot) and the HetRec 2011 MovieLens dataset (the lower plot).

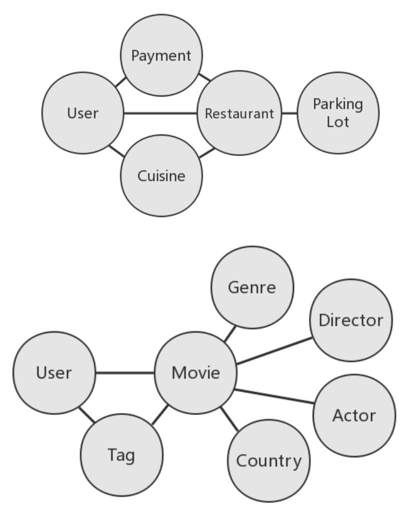

Figure 3: The network schemas of the HINs used in the item recommendation experiment. The upper network is constructed using the Restaurant & Consumer dataset and the lower network is constructed using the HetRec 2011 MovieLens dataset.

5.4.1 Experimental Setup. For each dataset, we create a heterogeneous information network based on the training set and the meta-data.

Using the Restaurant & Consumer dataset, we construct 5 types of nodes, and we abbreviate them as **U**: USER, **R**: RESTAURANT, **P**: PAYMENT, **C**: CUISINE, and **L**: PARKING LOT. We also construct 6 types of links. In this dataset, possible rating values are 0, 1 and 2, where 0 indicates that the user does not like the restaurant, and 2 denotes a high preference. We create a link between a user and a restaurant if the user rated 2 for the restaurant. That is to say, we treat these restaurants as the positive feedback to simulate the one-class feedback problem. The network schema is shown in the upper plot in Figure 3.

Using the HetRec 2011 MovieLens dataset, we construct 7 types of nodes, and use the capital of the first letter of each node type name as its abbreviation (**U**: USER, **M**: MOVIE, **T**: TAG, **G**: GENRE, **D**: DIRECTOR, **A**: ACTOR, **C**: COUNTRY). We also construct 7 types of links. For every rating on a movie m by a user u, we create a link between u and m, if and only if the rating is greater or equal to the average rating of the user u. In other words, we treat the movies whose ratings are larger than the user's personal average rating as the positive feedback. The network schema is shown in the lower plot in Figure 3.

We use the AUC (Area Under Curve of the receiver operating characteristic) criterion as the evaluation metric. We first evaluate AUC for each user from the test data, and then obtain the averaged AUC results over all users. For the Restaurant & Consumer dataset, in order to simulate sparse data, we use 50% of the dataset as the training data and the other 50% as the testing data. For the HetRec 2011 MovieLens dataset, we sort the ratings by their timestamps and use 80% as the training set and the rest 20% as the testing set.

5.4.2 Results. The experimental results of the compared methods on the two datasets are shown in Table 3. The best result on each dataset is displayed in bold. An entry is marked with '$*$' (or '$**$') if HeteLearn is significantly better than the compared method based on paired t-test at the significance level 0.05 (or 0.01).

Among all the compared methods, the proposed method, HeteLearn, achieves the best results on both datasets. And in most cases, the improvements of our method are significant at the level of 0.01.

The traditional user and item based collaborative filtering methods perform quite poor on the Restaurant & Consumer dataset. This is mainly because of the sparsity of the user-item rating matrix of this dataset.

Despite of the sparsity of the Restaurant & Consumer dataset, the matrix factorization (MF) methods, NMF and LFM, still perform quite well, which demonstrates the effectiveness of matrix factorization methods.

On both datasets, the performances of semantic recommendations using a single path (also known as the Path-constrained Random Walk method) vary greatly with the paths. The best path on the Restaurant & Consumer dataset is *UPUR*, which could be seen as a kind of user-based collaborative filtering. The best path on the HetRec 2011 MovieLens dataset is *UMGM*, which corresponds to genre-based filtering.

Both PathRank and Personalized PageRank could be seen as hybrid recommendation methods based on HINs. In our experimental results, these methods achieve better results than any of the single path methods, which demonstrates the effectiveness of HIN-based hybrid recommendation methods.

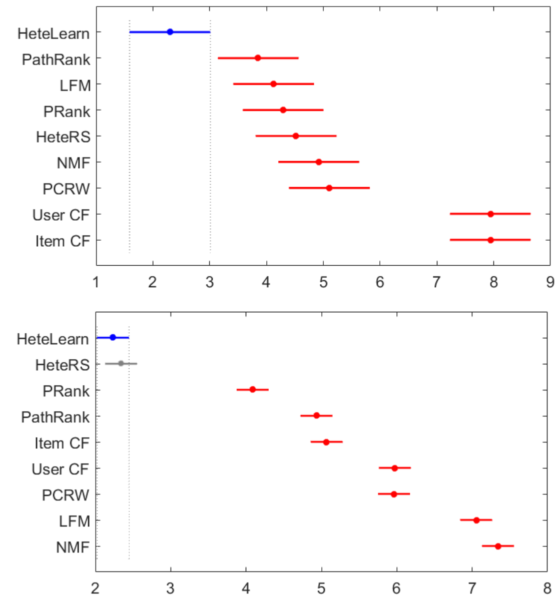

Figure 4: Freidman test results on the Restaurant & Consumer dataset (the upper plot) and the HetRec 2011 MovieLens dataset (the lower plot) on the item recommendation task.

To better compare the performance of the compared recommendation methods, we perform the Freidman test in conjunction with the Bonferroni-Dunn test [4] at significance level 0.05. The results are shown in Figure 4. The dots indicate the average ranks, the bars indicate the critical difference with the Bonferroni-Dunn test at significance level 0.05, and compared methods having non-overlapped bars are significantly different. For every kind of recommendation method, the best result is used to perform the test. On the Restaurant & Consumer dataset, HeteLearn shows significantly better result than all the other methods. And on the HetRec 2011 MovieLens dataset, HeteLearn is significantly better than all the other methods except HeteRS. Thus, the results confirm the effectiveness of our method.

It should be noticed that HeteLearn and PathRank perform better than all the traditional collaborative filtering methods. That is because of the heterogeneous information added to the user-item rating matrix, and both HeteLearn and PathRank could get use of such information effectively. Instead of setting the path-guides manually in PathRank, the proposed method, HeteLearn, could automatically learn them from the HINs. Therefore, the HeteLearn method has a broader range of potential applications.

5.5 Task 2: Tag Recommendation

We also compare our proposed method with other methods on the tag recommendation task, the goal of which is to find the set of proper words (tags) to describe the resources. Existing tag recommendation methods include content-based methods and co-occurrence based methods [25]. Content-based methods directly adopt the content of items, such as genre of movies, to perform tag recommendation. Co-occurrence based methods use the co-occurrence of

Table 3: AUC results of the compared methods on the item recommendation task. The best result on each dataset is displayed in bold. An entry is marked with '∗' (or '∗∗') if HeteLearn is significantly better than the compared method based on paired t-test at the significance level 0.05 (or 0.01).

Method	AUC (RCdata)		AUC (HetRec MovieLens)	
User CF	0.7241**		0.7004**	
Item CF	0.7241**		0.7190**	
NMF	0.8953**		0.6528**	
LFM	0.9147**		0.6587**	
Personalized PageRank	$\alpha = 0.6$	0.8932**	$\alpha = 0.6$	0.7551**
	$\alpha = 0.8$	0.9080**	$\alpha = 0.8$	0.7462**
	$\alpha = 1.0$	0.9139**	$\alpha = 1.0$	0.7163**
Path-constrained Random Walk	UR	0.6571**	UM	0.5212**
	UCR	0.6662**	UTM	0.5098**
	UPR	0.7941**	UMCM	0.5110**
	URCR	0.6745**	UTUM	0.5600**
	URUR	0.7242**	UMDM	0.6107**
	URLR	0.7563**	UMUM	0.6525**
	UCUR	0.8202**	UMAM	0.6619**
	URPR	0.8238**	UMTM	0.6648**
	UPUR	0.8684**	UMGM	0.6865**
PathRank	UMU(20)M+ UMUM(20)	0.9164**	UMTM+ UMU(20)M+ UMUM(20)	0.7127**
HeteRS	-	0.9154**	-	0.8152*
HeteLearn	$\alpha = 0.8$	**0.9396**	$\alpha = 0.8$	**0.8217**

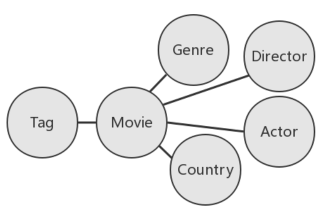

Figure 5: The network schema of the HIN used in the tag recommendation experiment. This network is constructed using the HetRec 2011 MovieLens dataset.

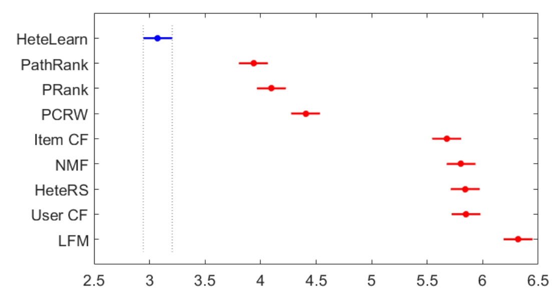

Figure 6: Freidman test result on the HetRec 2011 MovieLens dataset on the tag recommendation task.

tags among items (i.e., the item-tag matrix) for tagging. Our method, HeteLearn, is a hybrid method which combines both content-based methods and co-occurrence methods.

5.5.1 Experimental Setup. We test our method using the HetRec 2011 MovieLens dataset. We construct 6 types of nodes and used the capital of the first letter of each node type name as its abbreviation (**M**: MOVIE, **T**: TAG, **G**: GENRE, **D**: DIRECTOR, **A**: ACTOR, **C**: COUNTRY). We also construct 5 types of links. For every tag t assigned to a movie m, we create a link between t and m, if and only if the tag weight is greater or equal to the average tag weight assigned to the movie. The network schema is shown in Figure 5.

Similar to the item recommendation task, we also use the AUC criterion as the evaluation metric, and we sort the tag assignments by their timestamps and use 80% as the training set and the rest 20% as the testing set.

5.5.2 Results. The performances of the compared methods and the Friedman test result are shown in Table 4 and Figure 6.

Among the all the compared methods, HeteLearn achieves the best results, which again confirms the effectiveness of our method.

The co-occurrence methods, including User CF, Item CF, NMF and LFM, perform worse than the HIN-based methods, which coincides with the fact that hybrid methods usually achieve better performance than co-occurrence methods. Note that although NMF and LFM achieve higher average AUC results than User CF and Item CF, they do not show significantly better results according to

Table 4: AUC results of the compared methods on the tag recommendation task. The best result is in bold. An entry is marked with '∗' (or '∗∗') if HeteLearn is significantly better than the compared method based on paired t-test at the significance level 0.05 (or 0.01).

Method	AUC	
User CF	0.7128**	
Item CF	0.7115**	
NMF	0.8195**	
LFM	0.8180**	
Personalized PageRank	$\alpha = 0.6$	0.8924**
	$\alpha = 0.8$	0.8908**
	$\alpha = 1.0$	0.8423**
Path-constrained Random Walk	MT	0.5149**
	MDMT	0.5715**
	MTMT	0.7188**
	MAMT	0.7229**
	MCMT	0.8063**
	MGMT	0.8668**
PathRank	MTM(20)T+ MTMT(20)	0.8902**
HeteRS	-	0.8029**
HeteLearn	$\alpha = 0.8$	**0.9017**

Friedman test. This means that they only achieve good AUC results in a few individual cases, while in most cases they perform worse than User CF and Item CF.

6 CONCLUSION

In this paper, we proposed a new personalized recommendation method in heterogeneous information networks. In this method, we used a generalized random walk with restart model to model the user preferences, and used a BPR-based learning method to learn the weights of the links in the network. Our experimental results confirmed the effectiveness of the proposed method and provided several useful insights for the field of recommender systems. Both our theoretical and empirical results indicate that learning on heterogeneous information networks is feasible and worthwhile. And the proposed method, HeteLearn, provides a successful case for the important tasks of personalized recommendation and tag recommendation.

ACKNOWLEDGMENTS

The work is supported by the National 863 Program of China under Grant No.: 2015AA016009, the National Natural Science Fund of China under Grant No.: 61232005 and the National Key R&D Program of China under Grant No.: 2017YFB1002002.

REFERENCES

[1] Andrey Balmin, Vagelis Hristidis, and Yannis Papakonstantinou. 2004. Objectrank: Authority-based keyword search in databases. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. VLDB Endowment, 564–575.

[2] Yang Bao, Hui Fang, and Jie Zhang. 2014. TopicMF: Simultaneously Exploiting Ratings and Reviews for Recommendation.. In *AAAI*, Vol. 14. 2–8.

[3] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. 2010. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 39–46.

[4] Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research* 7, Jan (2006), 1–30.

[5] Michael D Ekstrand, John T Riedl, Joseph A Konstan, et al. 2011. Collaborative filtering recommender systems. *Foundations and Trends® in Human–Computer Interaction* 4, 2 (2011), 81–173.

[6] Francois Fouss, Alain Pirotte, Jean-Michel Renders, and Marco Saerens. 2007. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Transactions on knowledge and data engineering* 19, 3 (2007).

[7] Marco Gori, Augusto Pucci, V Roma, and I Siena. 2007. ItemRank: A Random-Walk Based Scoring Algorithm for Recommender Engines.. In *International Joint Conference on Artificial Intelligence*, Vol. 7. 2766–2771.

[8] Jiawei Han. 2009. Mining heterogeneous information networks by exploring the power of links. In *International Conference on Algorithmic Learning Theory*. 20–28.

[9] Taher H Haveliwala. 2003. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *IEEE transactions on knowledge and data engineering* 15, 4 (2003), 784–796.

[10] Douglas J Klein and Milan Randić. 1993. Resistance distance. *Journal of Mathematical Chemistry* 12, 1 (1993), 81–95.

[11] Ni Lao and William W Cohen. 2010. Fast query execution for retrieval models based on path-constrained random walks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 881–888.

[12] Sangkeun Lee, Sanghyeb Lee, and Byoung-Hoon Park. 2015. PathMining: A Path-Based User Profiling Algorithm for Heterogeneous Graph-Based Recommender Systems.. In *FLAIRS Conference*. 519–523.

[13] Sangkeun Lee, Sungchan Park, Minsuk Kahng, and Sanggoo Lee. 2013. PathRank: Ranking nodes on a heterogeneous graph for flexible hybrid recommender systems. *Expert Systems With Applications* 40, 2 (2013), 684–697.

[14] M. Lichman. 2013. UCI Machine Learning Repository. (2013). http://archive.ics.uci.edu/ml

[15] Weiping Liu and Linyuan Lü. 2010. Link prediction based on local random walk. *EPL (Europhysics Letters)* 89, 5 (2010), 58007.

[16] Linyuan Lu, Matus Medo, Ho Yeung Chi, Yi Cheng Zhang, Zi Ke Zhang, and Tao Zhou. 2012. Recommender systems. *Physics Reports* 519, 1 (2012), 1–49.

[17] Hendri Murfi and Klaus Obermayer. 2009. A Two-Level Learning Hierarchy of Concept Based Keyword Extraction for Tag Recommendations.. In *DC@ PKDD/ECML*.

[18] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank citation ranking: Bringing order to the web.* Technical Report. Stanford InfoLab.

[19] Rong Pan, Yunhong Zhou, Bin Cao, Nathan N Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. 2008. One-Class Collaborative Filtering. In *IEEE International Conference on Data Mining*. 502–511.

[20] Weike Pan and Li Chen. 2013. GBPR: group preference based Bayesian personalized ranking for one-class collaborative filtering. In *International Joint Conference on Artificial Intelligence*. 2691–2697.

[21] Tuan-Anh Nguyen Pham, Xutao Li, Gao Cong, and Zhenjie Zhang. 2016. A general recommendation model for heterogeneous networks. *IEEE Transactions on Knowledge and Data Engineering* 28, 12 (2016), 3140–3153.

[22] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press, 452–461.

[23] Chuan Shi, Xiangnan Kong, Yue Huang, and P. S Yu. 2014. HeteSim: A General Framework for Relevance Measure in Heterogeneous Networks. *Knowledge & Data Engineering IEEE Transactions on* 26, 10 (2014), 2479–2492.

[24] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S. Yu, and Tianyi Wu. 2011. PathSim: Meta Path-Based Top-K Similarity Search in Heterogeneous Information Networks. *Proceedings of the Vldb Endowment* 4, 11 (2011), 992–1003.

[25] Hao Wang, Binyi Chen, and Wu-Jun Li. 2013. Collaborative Topic Regression with Social Regularization for Tag Recommendation.. In *IJCAI*.

[26] Jun Wang, Arjen P De Vries, and Marcel JT Reinders. 2006. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 501–508.

[27] Fei Yu, An Zeng, Sébastien Gillard, and Matúš Medo. 2016. Network-based recommendation algorithms: A review. *Physica A: Statistical Mechanics and its Applications* 452 (2016), 192–208.

[28] Xiao Yu, Xiang Ren, Yizhou Sun, Bradley Sturt, Urvashi Khandelwal, Quanquan Gu, Brandon Norick, and Jiawei Han. 2013. Recommendation in heterogeneous information networks with implicit user feedback. In *ACM Conference on Recommender Systems*. 347–350.