World Wide Web (2015) 18:1351–1371
DOI 10.1007/s11280-014-0307-z

A Graph-based model for context-aware recommendation using implicit feedback data

Weilong Yao · Jing He · Guangyan Huang · Jie Cao · Yanchun Zhang

Received: 30 December 2013 / Revised: 23 July 2014 /
Accepted: 1 August 2014 / Published online: 13 August 2014

Abstract Recommender systems have been successfully dealing with the problem of information overload. However, most recommendation methods suit to the scenarios where explicit feedback, e.g. ratings, are available, but might not be suitable for the most common scenarios with only implicit feedback. In addition, most existing methods only focus on user and item dimensions and neglect any additional contextual information, such as time and location. In this paper, we propose a graph-based generic recommendation framework, which constructs a Multi-Layer Context Graph (MLCG) from implicit feedback data, and then performs ranking algorithms in MLCG for context-aware recommendation. Specifically, MLCG incorporates a variety of contextual information into a recommendation process and models the interactions between users and items. Moreover, based on MLCG, two novel ranking methods are developed: Context-aware Personalized Random Walk (CPRW) captures user preferences and current situations, and Semantic Path-based Random Walk (SPRW) incorporates semantics of paths in MLCG into random walk model for recommendation. The experiments on two real-world datasets demonstrate the effectiveness of our approach.

W. Yao (✉)
University of Chinese Academy of Sciences, Beijing, China
e-mail: yaoweilong12@mails.ucas.ac.cn

J. He · Y. Zhang
Centre for Applied Informatics, Victoria University, Melbourne, Australia

J. He
e-mail: Jing.He@vu.edu.au

Y. Zhang
e-mail: Yanchun.Zhang@vu.edu.au

G. Huang
Deakin University, Melbourne, Australia
e-mail: Guangyan.Huang@deakin.edu.au

J. Cao
Nanjing University of Finance and Economics, Nanjing, China
e-mail: Jie.Cao@njue.edu.cn

Keywords Recommendation · Graph model · Context · Semantics · Implicit feedback

1 Introduction

Recommender systems attempt to tackle the problem of information overload by suggesting users the information that is potentially of interests. A substantial amount of research has already been conducted by both industry and academia in the area of recommender systems [2, 7, 20]. Most existing methods mainly focus on explicit feedback data, i.e., ratings, which represent the strength of the user's preference for one item [29]. However, explicit feedback is not always available in real-life scenarios, instead, most of the data produced by users is implicit, such as music listening logs or purchase histories, which are recorded automatically in the system. Adopting directly existing approaches in scenarios with only implicit feedback may not be effective [17]. Hence, how to generate more accurate recommendations based on only available implicit feedback data, which indirectly reflects user interests, remains a problem to be solved.

Compared to conventional recommender methods that operate in the two-dimensional $USER \times ITEM$ space [2], incorporating contextual information, such as time, location or genre, into a recommendation process can achieve better recommendation accuracy. For example, a vacation recommendation for a given user may depend on seasons, ages and interests. More specifically, in summer, an elderly user would probably prefer to enjoy his/her vacation on a peaceful beach rather than on a ski resort.

Generally speaking, we consider three types of contexts in recommender systems with implicit feedback: 1) User context describes a user's personal information, such as gender, age, education and social meta attributes [36]. We assume that users sharing more common user contexts tend to have similar tastes or preferences. 2) Item context enables measuring correlation between two items. E.g., the textual information of an item [32]. 3) The third type of contextual information called decision context [29, 35], involves context where the decision is made, such as time, location or mood. It is often observed that the same user with different decision context shows different preferences [2] and that is why we add the decision context into the proposed model. For instance, the style of songs that a user listens to at home on weekends may be different from the style of songs he/she listens to in office on weekdays. Only the model considering user, item and decision context in one whole infrastructure could have the better sense of the different preferences for recommender system. However, few methods can incorporate all the three types of contexts into a comprehensive recommendation process, and most approaches focus on only partial context (see Section 7). By adding users and items into the constructed context layer, the proposed context-aware recommendation model is expected to achieve the higher accuracy.

In this paper, we proposed Multi-Layer Context Graph (MLCG), a generic framework for context-aware recommendation using implicit feedback data. In MLCG, we consider all of the three types of contexts into recommender systems, and model the interactions between users and items in the corresponding decision context. We then provide two personalized random walk-based ranking methods in a MLCG to capture user preferences and incorporate semantic information among paths.

The most related work that also incorporates contextual information on graph by utilizing nodes to represent multidimensional data is Graph-based Flexible Recommendation (GFREC) proposed in [22]. The major differences between MLCG and GFREC lie in two aspects: First, GFREC strengthens the connections between users and items by merging

different dimensions of contexts, whereas our method focuses on the instant situation where users interacts with items. That is, MLCG emphasizes the fact that user-item interactions occur in a certain context. Second, as a bipartite graph, GFREC can not incorporate additional but helpful information for ranking, such as semantics. However, with the multi-layer structure, the proposed MLCG can be easily integrated with various semantic information in the ranking stage (see Section 4.3) to yield better recommendations. Since GFREC shares a close relationship with the proposed MLCG in terms of dealing with contextual information, we choose GFREC as the main baseline to compare in the experiments.

To summarize, our main contributions are as follows:

1. We propose a Multi-Layer Context Graph (MLCG) model that incorporates all of the three types of contexts extracted from implicit feedback data for recommendation. In particular, MLCG emphasizes the interactional and real-time situation of the user-item interactions.
2. Based on MLCG, we provide a new ranking algorithm, Context-aware Personalized Random Walk (CPRW), which extends PageRank for increasing accuracy of top-K recommendation through running in a MLCG that models the intra-/inter-layer influence flow.
3. We further develop a Semantic Path-based Random Walk (SPRW) algorithm to capture various semantics of different paths in MLCG for recommendation.
4. We conduct a comprehensive experimental study on two real-world implicit feedback datasets. And the results demonstrate the effectiveness of our proposed method.

The remainder of this paper is organized as follows. Section 2 formally defines the context and research problem involved in this work. Section 3 demonstrates how to construct a MLCG from implicit feedback data. Two ranking methods for recommendation are presented in Section 4, followed by a discussion for the flexibility of the proposed framework in Section 5. Section 6 reports the experimental study. Section 7 presents related work. Section 8 concludes this paper.

2 Problem formulation

In this section, we formally define three types of contexts involved in this paper, and then we define the context-aware implicit feedback recommendation problem. Let $U = \{u_k|k = 1, 2, ..., |U|\}$ be a set of users, $I = \{i_k|k = 1, 2, ..., |I|\}$ a set of items.

Definition 1 User context is defined as $C_U = \left\{C_{U_k}|k = 1, 2, ..., |C_U|\right\}$, where C_{U_k} is a domain of user context (e.g., AGE, GENDER). C_{U_k} can be a categorical domain or categorical set domain.

For instance, $C_U = \{\langle Gender : Male\rangle, \langle Friends : Ted, Mike\rangle\}$ depicts a male user having two friends: Ted and Mike. The subscript U represents the context of users.

Definition 2 Item context is defined as $C_I = \left\{C_{I_k}|k = 1, 2, ..., |C_I|\right\}$, where C_{I_k} is a domain of item context (e.g., GENRE, ARTIST). C_{I_k} can be a categorical domain or categorical set domain.

For instance, $C_I = \{\langle Genre : Rock\rangle, \langle Artist : MichaelJackson\rangle\}$ means a rock style song from Michael Jackson.

Definition 3 Decision context is defined as $C_D = \left\{C_{D_k} | k = 1, 2, ..., |C_D|\right\}$, where C_{D_k} is a domain of decision context (e.g., TIME, LOCATION). C_{D_k} can be a categorical domain or categorical set domain.

For instance, $C_D = \{\langle Time : Weekend\rangle, \langle Location : Home\rangle\}$ depicts the interaction that occurred at home during weekend.

For simplicity, we assume that the contextual information is denoted by categorical values. For real value domains, we can transform the numerical values into categorical values. E.g., age values can be categorized into "Old", "Middle" and "Young".

Definition 4 Context-aware recommendation is defined as: given a user $u \in U$ with user context $c_u = \left\{c_{u_k} | c_{u_k} \in C_{U_k}, k = 1, 2, ..., |C_U|\right\}$ in a particular decision context $c_d = \left\{c_{d_k} | c_{d_k} \in C_{D_k}, k = 1, 2, ..., |C_D|\right\}$, a ranked list of items $R(u, c_d) \subseteq I$ is provided as the potential items ranked by relevance scoring function $f(x)$, where $f(x)$ is defined as $f(u, c_u, c_d, i)$ to measure the relevance between tuple $\langle u, c_u, c_d\rangle$ and item $i \in I$.

For example, recommending songs for a user, *Ted*, when he stayed *home* at *Saturday night* can be interpreted as finding the songs with top-K relevance with tuple $\langle u, c_u, c_d\rangle$, where $u = Ted$, $c_u = \{\langle Gender : M\rangle\}$ and $c_d = \{\langle$ *DayofWeek : Saturday* $\rangle, \langle$ *Time : Night* $\rangle, \langle$ *Location : Home* $\rangle\}$.

3 MLCG construction

In this section, we illustrate how to construct a multi-layer context graph from implicit feedback data.

3.1 Construction algorithm

As a toy example, Tables 1-3 provide examples of implicit feedback data: in Table 1, C_U is $\{\langle Gender\rangle, \langle AGE\rangle\}$, and C_I is $\{\langle Artist\rangle, \langle Genre\rangle\}$ in Table 2. Meanwhile, log data in Table 3 describes the situation when a user listened to a song. For instance, the first log record in Table 3 shows that the user, *Ted*, listened to a *rock* style song named *Beat it* from the artist, *Michael Jackson*, when he was at *home* on *Saturday*.

In graph-based methods, typically each item or user is represented as a node. Most graph-based methods describe the interaction between a user and an item by directly creating an

Table 1 Example of user data

USER	GENDER	AGE
Ted	M	[18-30]
Mike	M	[18-30]

Table 2 Example of item data

SONG	ARTIST	GENRE
Beat It	Michael Jackson	Rock
Rock with you	Michael Jackson	Rock

Table 3 Example of listening log data

USER	DayofWeek	LOCATION	SONG
Ted	Saturday	Home	Beat it
Mike	Sunday	Office	Rock with it
Mike	Sunday	Office	Beat it
Ted	Sunday	Office	Beat it

edge between the two nodes [5, 11, 12, 18, 22, 33]. However, these methods ignore the effect of decision contexts, while we argue that, in a recommender system, it is in a certain decision context where users interact with items. That is, user preferences on items should flow though the particular decision context before reaching the item nodes.

So, we design a new node type, decision node $D = \langle u, c_{d_1}, c_{d_2}, ..., c_{d_{|C_D|}} \rangle$, where $u \in U$ and $c_{d_k} \in C_{D_k}$, to characterize the decision context and model the user-item interactions. Note that decision node D is a combined node with a user and a decision context. The underlying intuition of D is that decision context is a local effect and should not be shared by all users as a global effect. That is, the same decision context for different users may have different impacts on decision making. The proposed MLCG construction algorithm is outlined in Algorithm 1.

Algorithm 1 Construct a Multi-Layer Context Graph

Input: Set of users U and context C_U; Set of Items I and context C_I; Decision Context C_D; Log record table $LogTable$, where each log is in the form of $< u, c_{d_1}, c_{d_2}, ..., c_{d_{|C_D|}}, i >$, $u \in U$, $i \in I$ and $c_{d_k} \in C_{D_k}$

Output: Multi-Layer Context Graph $\mathcal{G}$

Initialize a graph $\mathcal{G}$ with $USER\text{-}Layer$, $ITEM\text{-}Layer$ and $Decision\text{-}Layer$
$CreateLayer(C_U, U, USER\text{-}Layer)$
$CreateLayer(C_I, I, ITEM\text{-}Layer)$
for each context domain $c_d \in C_D$ **do**
    for each context value $v \in c_d$ **do**
        Create a decision context node for v on $Decision\text{-}Layer$
    end for
end for
for each log record $log \in LogTable$ **do**
    Create a decision node $v =< u, c_{d_1}, c_{d_2}, ..., c_{d_{|C_D|}} >$ on Decision Layer
    Connect nodes $c_{d_1}, c_{d_2}, ..., c_{d_{|C_D|}}$ and v
    Connect user node u and v
    Connect item node i node and v
end for
Return $\mathcal{G}$

Subroutine CreateLayer$(C, T, Layer)$
for each context domain $c \in C$ **do**
    for each context value $v \in c$ **do**
        Create a context node for v on Layer
    end for
end for
for each entity $t \in T$ **do**
    Create a node for t on $Layer$
    Connect node t and its corresponding context nodes
end for

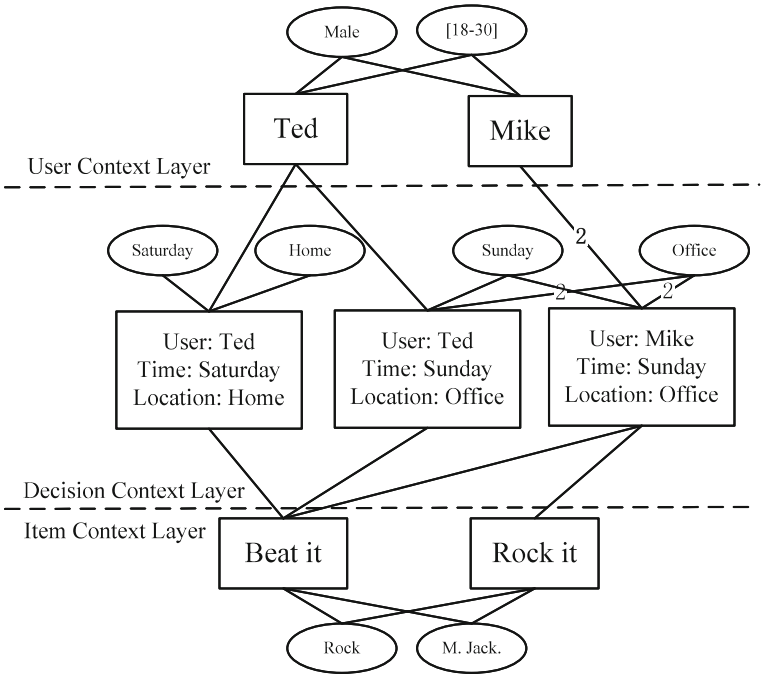


Figure 1 An example: the MLCG constructed based on Tables 1-3

As the Algorithm 1 shows, a MLCG, denoted as $\mathcal{G}$, is a three-layer graph that consists of a user context layer, an item context layer and a decision context layer. Figure 1 illustrates an example of MLCG constructed from Tables 1-3. The number on the edge represents the co-occurrence of two end-nodes. For example, in Table 3, user, *Mike*, expressed two interactions in office, thus, the co-occurrence of nodes *Mike* and *office* is 2.

Only one type of entity node is denoted as a square node in Figure 1, in each layer, such as D on decision context layer. Entity nodes are characterized by their own context nodes on the same layer. For every context/attribute of an entity, there is a corresponding edge between the entity node and the context node. As Figure 1 shows, node M describes the gender of the user nodes connected with it. Furthermore, the nodes of the same entity type interact through their sharing context nodes. That is, influence from an entity node propagates to another entity through co-linked context nodes. In the case of Figure 1, the more rock songs a user listened to, the more his/her preferences flow to other rock songs through the node *ROCK*.

In addition to the intra-layer edges, inter-layer edges are also available to represent the interactions between layers. In our model, user nodes do not directly interact with items, instead, they should interact through a certain decision context. Thus, the interaction is expressed as: an edge from a user node in user context layer to the corresponding decision node and the other edge from the decision node to a song node in item context layer. For an active user, songs which were listened to in the same context are connected to the same decision node. In this way, the effect of current context is distributed precisely over these songs listened in that context, as shown in Figure 2.

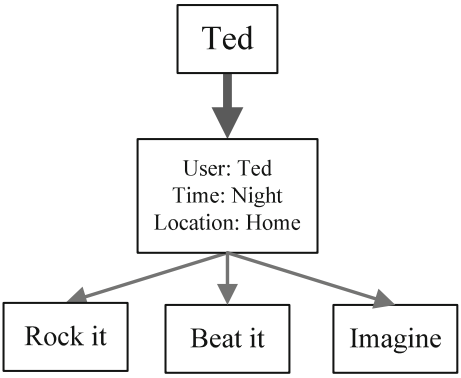


Figure 2 Preference in decision context is distributed precisely over songs

3.2 Weight assignment

Most graph-based recommendation methods consider a recommendation process as a node ranking task on a graph, hence several random walk-based ranking measures are proposed [12, 13, 22, 33]. However, these ranking methods are performed on a homogeneous graph, ignoring the different types of edges, so they do not work for MLCG. By fusing different edge types, we transform the heterogeneous multi-layer graph into a homogeneous graph.

We denote $N(j)$ as a set of nodes connected with node j, $N_s(j) \subseteq N(j)$ a set of nodes on the same layer with node j, and $N_d(j) = N(j) \setminus N_s(j)$. Given node j and $k \in N_s(j)$ on any layer, the edge weight $w(j,k)$ is defined as follows:

$$w(j,k) = \begin{cases} \frac{\alpha}{n_c(j)} \frac{f(j,k)}{\sum_{t \in N_s(j)} f(j,t)} & \text{if } |N_d(j)| > 0 \\ \frac{co\text{-}occu(j,k)}{\sum_{t \in N_s(j)} co\text{-}occu(j,t)} & \text{if } |N_d(j)| = 0 \\ 0 & otherwise, \end{cases} \tag{1}$$

where function $f(j,k)$ denotes the importance score for node k with regard to node j, and $co\text{-}occu(j,k)$ is the co-occurrence of node j and node k. $n_c(j)$ represents the number of contextual nodes types of node j.

Meanwhile, $f(j,k)$ should satisfy the following intuition-based criteria: 1) Rare contexts/attributes are likely to be more important, whereas common contexts/attributes are less important. For example, thousands of people like football but only *Ted* and *Mike* like Ping-Pong, in which case node *PingPong* carries more benefit for looking for similar user than node *Football*. 2) The more co-occurrence of two nodes, the more related they are. For example, if a user who is inclined to listen to songs at home, then his preferences propagate mainly through node *Home* to other songs. Considering the two intuitive rules, we borrow the idea of TF-IDF from information retrieval area and define $f(j,k)$ as follows.

$$f(j,k) = co\text{-}occu(j,k) \log \frac{\sum_{v \in \Omega(k)} \sum_{t \in N_s(v)} co\text{-}occu(v,t)}{\sum_{t \in N_s(k)} co\text{-}occu(k,t)}, \tag{2}$$

where $\Omega(k)$ is a set of nodes sharing the same node type with node k. For example, $\Omega(Saturday) = \{Saturday, Sunday\}$ since they share the node type $Dayof Week$.

Given node j and $k \in N_d(j)$, the edge weight $w(j,k)$ is calculated as:

$$w(j,k) = \begin{cases} \frac{(1-\alpha)co\text{-}occu(j,k)}{\sum_{t \in N_d(j)} co\text{-}occu(j,t)} & \text{if } |N_s(j)| > 0 \\ \frac{co\text{-}occu(j,k)}{\sum_{t \in N_d(j)} co\text{-}occu(j,t)} & \text{if } |N_s(j)| = 0 \\ 0 & otherwise. \end{cases} \tag{3}$$

Here, α controls the trade-off between intra-layer and inter-layer interactions. The larger α is, the more effect intra-layer interactions have, since the more influence flow to nodes in the same layer.

4 Ranking in MLCG

User, item and decision contexts are characterized in the responding layer respectively in MLCG, and now we present our techniques of modeling the preference propagation for context-aware recommendation. More specifically, we frame this challenge as ranking problem in graph.

4.1 Preliminaries

PageRank [25] is the most popular node ranking methods in many fields, such as information retrieval. The original PageRank score of a node is given by:

$$\mathbf{PR}(t+1) = \mu \cdot \mathbf{M} \cdot \mathbf{PR}(t) + (1-\mu) \cdot \mathbf{d}, \tag{4}$$

where $\mathbf{PR}(t)$ denotes the rank value at the t-th iteration, $\mathbf{M}$ is a transition probability matrix, μ is a damping factor that is normally given as 0.85 and $\mathbf{d}$ is a vector defined as $\mathbf{d}_k = \frac{1}{n}, k = 1, 2, ..., n$, where n is the number of nodes.

However, PageRank is not suitable for the personalized ranking task for the following two reasons:

1. According to (4), PageRank can be considered as a Markov process with restart, and the probability of a random walker jumps to a node after a restart is equal to others. That is, the PageRank algorithm considers all nodes equally without biasing any important nodes. Thus, the original PageRank may be not effective in modeling user preferences.
2. In essence, PageRank ranks nodes solely based on the graph structure, and disregards the different types of nodes and edges. Generally, nodes with many incoming links (i.e., popular items) tend to be highly scored, in which case the result may not be good enough. It is implied that original PageRank may obtain similar performance with popularity-based methods.

To overcome these two issues, we elaborate two random walk-based ranking methods for personalized recommendation in MLCG. More specifically, context-aware personalized random walk (CPRW) captures user preferences by taking into account historic records, and semantic path-based random walk (SPRW) further fuses semantic information of paths into ranking models.

4.2 Context-aware personalized random walk

We extend the PageRank as context-aware personalized random walk for ranking items in MLCG. The proposed ranking method is similar to topic-sensitive PageRank [15]. In [15], a personalized vector is introduced to bias user preference. More specifically, $\mathbf{d}$ is built as a user-specific personalized vector, where $\mathbf{d}_k = 1$ if k-th node represents the active user, otherwise $\mathbf{d}_k = 0$. Then the Personalized PageRank is calculated by (4).

Referring to [13, 33], we extend the PageRank as CPRW which captures both user preferences and current decision contexts. Given a user u and a current decision context $c_d = \left\{c_{d_k} | c_{d_k} \in C_{D_k}, k = 1, 2, ..., |C_D|\right\}$, we define $\Phi = \{i_k | i_k \in I, k = 1, 2, ..., |\Phi|\}$ as a set of items that u accessed before, then we construct $\tilde{\mathbf{d}}$ as follows:

$$\tilde{\mathbf{d}}_j = \begin{cases} \frac{\lambda}{|\Phi|} & \text{if node } j \in \Phi \\ \frac{1-\lambda}{|c_d|+1} & \text{if node } j \in c_d \text{ or node } j = u \\ 0 & otherwise, \end{cases} \tag{5}$$

where λ adjusts the radio of bias between the user's preferences and the current decision context. Meanwhile, we consider the recommendation process as a multi-path random walk process with multiple starting points. Hence $\mathbf{PR}(0)$ is defined as follows.

$$\mathbf{PR}(0)_j = \begin{cases} 1 & \text{if node } j \in c_d \text{ or node } j = u \\ 0 & otherwise. \end{cases} \tag{6}$$

Then, we normalize $\mathbf{PR}(0)$ to ensure that the sum of its non-negative elements is 1. We present the recommendation method in MLCG using CPRW in Algorithm 2.

Algorithm 2 Context-aware Personalized Random Walk (CPRW)

Input: Transition probability matrix M for MLCG $\mathcal{G}$, user u
Output: Top-K items.
1: Construct $\tilde{\mathbf{d}}$ for the active user u based on Equation 5
2: Initialize $\mathbf{PR}(0)$ based on Equation 6
3: $t \leftarrow 1$
4: **while** $|\mathbf{PR}(t) - \mathbf{PR}(t-1)| < \varepsilon$ **do**
5: $\quad \mathbf{PR}(t+1) = \mu \cdot \mathbf{M} \cdot \mathbf{PR}(t) + (1-\mu) \cdot \tilde{\mathbf{d}}$
6: **end while**
7: Rank item nodes based on $\mathbf{PR}$ values
8: Return Top-K items

4.3 Semantic path-based random walk

Inspired by the work on path [21, 30], we provide a path-based ranking method, i.e., semantic path-based random walk, which exploits the semantics among paths connecting two nodes, by extending (4) further.

The graph schema, which describes the meta structure of a graph, is defined as:

Definition 5 Graph schema is a directed graph $\mathcal{S} = (T, R)$ derived from MLCG, where $T = \{T_k | k = 1, 2, ..., |T|\}$ denotes the set of node types in MLCG and R denotes the relations between two node types, i.e., $R \subseteq T \times T$.

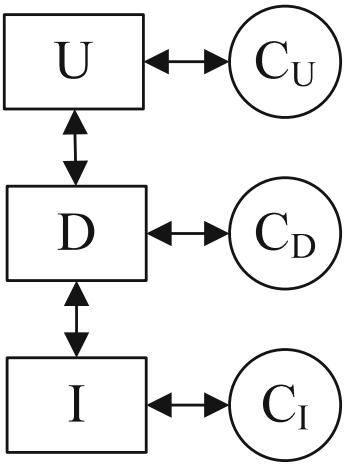


Figure 3 Graph schema of MLCG

For simplifying the problem, we only distinguish high-level node types, that is, we consider entity nodes and their context nodes. Note that we use the variant name as the node type name in the following parts, e.g., U is for user nodes and C_U is for user context nodes. Then the graph schema for MLCG is shown in Figure 3:

Definition 6 Semantic path is defined on the graph schema $\mathcal{S} = (T, R)$, and is denoted in the form of $T_1 \xrightarrow{R_1} T_2 \xrightarrow{R_2} \ldots \xrightarrow{R_l} T_{l+1}$, where $T_k \in T$ represents node type (i.e., U, C_U,

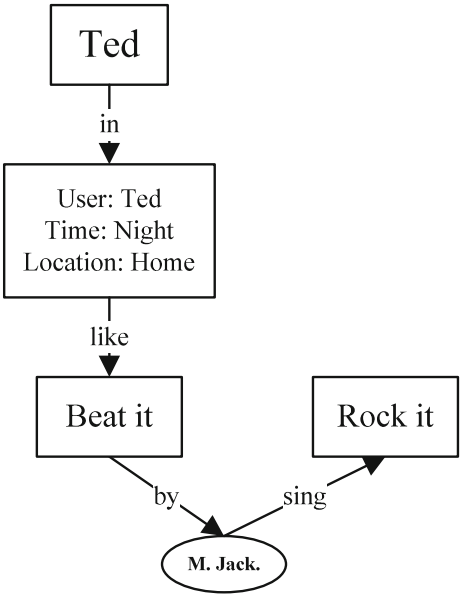


Figure 4 Example of path $U - D - I - C_I - I$

D, C_D, I, and C_I) and $R_k \in R$ represents the semantic relation between node type T_k and T_{k+1}.

For instance, semantic path $\mathcal{P} = (U - D - I)$ represents one path from a user node to an item node via a decision context node. The semantics behind this path represent that the active user prefers to the items he/she has accessed before in the same decision context. Figure 4 shows another path from user nodes to item nodes with semantic relations on edges. Hence, motivated by the intuition that various paths can reflect various semantics, we incorporate paths into random walk model for ranking.

Given a MLCG and user-specified paths $\mathbf{P} = \{\mathcal{P}_k | k = 1, 2, ..., |P|\}$, where $\mathcal{P}_k = (T_1 - T_2 - ... - T_{l_k+1})$ and l_k is the path length, SPRW updates ranking values as follows:

$$\mathbf{PR}(t+1) = \underbrace{\mu \cdot \mathbf{M} \cdot \mathbf{PR}(t) + (1 - \mu - \nu) \cdot \tilde{\mathbf{d}}}_{\text{CPRW}} + \nu \sum_{k=1}^{|\mathbf{P}|} w_k \mathbf{M}_{\mathcal{P}_k} \cdot \mathbf{PR}(t), \tag{7}$$

where $\mathbf{M}_{\mathcal{P}_k}$ is the transition matrix of the path $\mathcal{P}_k$, and $\sum_{k=1}^{|\mathbf{P}|} w_k = 1$. μ and ν are tuneable parameters. As the equation shows, SPRW is obtained by fusing path-constrained random walk into CPRW.

For one path $\mathcal{P} = (T_1 - T_2 - ... - T_{l+1})$, the transition matrix is defined as $\mathbf{M}_{\mathcal{P}} = \mathbf{W}_{T_1T_2}\mathbf{W}_{T_2T_3}...\mathbf{W}_{T_lT_{l+1}}$, where $\mathbf{W}_{T_kT_{k+1}}$ is the transition matrix between node types T_k and T_{k+1}, that is,

$$\mathbf{W}_{T_kT_{k+1}}(i,j) = \begin{cases} \frac{w(i,j)}{\sum_{k \in \Omega(j)} w(i,k)} & \text{if } \phi(i) = T_k \text{ and } \phi(j) = T_{k+1} \\ 0 & otherwise, \end{cases} \tag{8}$$

where function $\phi(x)$ returns the type of node x, e.g., C_U.

Semantic paths between two node types can be obtained by traversing the graph schema, using traversal methods such as BFS algorithm. To realize a context-aware item recommendation, we can enumerate all the semantic paths within a length constraint staring with node type U and ending with node type I.

In summary, we present the semantic path-based random walk method in Algorithm 3.

Algorithm 3 Semantic Path-based Random Walk (SPRW)

Input: Transition probability matrix M for MLCG $\mathcal{G}$, user-specified paths $\mathbf{P}$,user u
Output: Top-K items.
1: Construct transition matrices between node types based on $\mathbf{M}$ and $\mathbf{P}$
2: Construct $\tilde{\mathbf{d}}$ for the active user u based on Equation 5
3: Initialize $\mathbf{PR}(0)$ based on Equation 6
4: $t \leftarrow 1$
5: **while** $|\mathbf{PR}(t) - \mathbf{PR}(t-1)| < \varepsilon$ **do**
6: $\quad \mathbf{PR}(t+1) = \mu \cdot \mathbf{M} \cdot \mathbf{PR}(t) + (1 - \mu - \nu) \cdot \tilde{\mathbf{d}} + \nu \sum_{k=1}^{|\mathbf{P}|} w_k \mathbf{M}_{\mathcal{P}_k} \cdot \mathbf{PR}(t)$
7: **end while**
8: Rank item nodes based on $\mathbf{PR}$ values
9: Return Top-K items

5 Discussion

The proposed MLCG is a generic recommendation framework, which can be extended to include additional contextual information and allows various forms of personalized recommendation.

5.1 Item promotion

So far, we have discussed user-oriented recommendation, whereas item-oriented recommendation is also of practical significance when service providers promote some items in some particular decision contexts. In the example of "Who are the potential consumers for the comedy movie *Turbo* on Monday night", $c_i = \{Genre : Comedy\}$ and $c_d = \{DOW : Monday, Time : Night\}$.

In CPRW, by modifying (5), the proposed framework can address the issue. Specifically, we construct the bias vector $\tilde{\mathbf{d}}$ as follows to capture user preferences on the target item:

$$\tilde{\mathbf{d}}_j = \begin{cases} \frac{\lambda}{|\Phi'|} & \text{if node } j \in \Phi' \\ \frac{1-\lambda}{|c_i|+1} & \text{if node } j \in c_i \text{ or node } j = i \\ 0 & otherwise, \end{cases} \tag{9}$$

where Φ' denotes the set of users who have watched/liked the target item before. In this way, the preferences from the users who like comedy movie or tend to watch movie on Saturday afternoon will propagate to other user nodes.

In SPRW, we only need to specify the paths from item nodes to user nodes, for instance, $\mathcal{P} = (I - C_I - D - U)$ represents that the promotional items can be recommended to the users who like the similar items.

5.2 Similar user recommendation

Distinct from context-aware recommendation, user recommendation focuses on finding the users who share the most similar interests with the active user [24]. To satisfy this need, based on MLCG, we only need to rank user nodes, instead of item nodes, for recommendations (line 7 in Algorithm 2). Also, we can extract paths staring and ending with user nodes for SPRW.

5.3 Group recommendation

By extending (5) to multiple users, the proposed MLCG can capture simultaneously preferences of one group of users to generate a ranked list of items [14]. For example, MLCG can provide movie recommendations for the active user and his/her girlfriend/boyfriend or family members.

6 Experiments

In this section, we present an experimental study which is conducted on two real-world datasets to demonstrate the effectiveness of our approach.

Table 4 Semantic paths used in our experiments

Semantic Path	Weight
$U - D - I$	0.5
$U - D - I - C_I - I$	0.2
$U - D - I - C_D - I$	0.3

6.1 Dataset

The first dataset is Last.fm[1] which contains 19,150,868 music listening logs of 992 users (till May, 4th 2009). We extract the logs from April to May and remove these songs which were listened to less than 10 times, then the final dataset contains 992 users, 12,286 songs and 264,446 logs. Finally, we use logs in April as a training set, and randomly choose 1000 logs from 1st to 4th May 2009 as a test set.

The second dataset is CiteULike[2] which provides logs on *who* posted *what* and *when* the posting occurred. By removing users who posted less than 5 papers and papers which were posted by less than 5 users, we obtain a subset of original dataset which contains 1,299 users, 5,856 items and 40,067 user-items pairs from January to May 2007. Finally, we use logs in the two final weeks as a test set, and others as a training set.

Note that the two datasets used in our experimental study are for two scenarios. For Last.fm dataset, our experiments provide a ranked list of songs, which might have been listened to before by the given user, since users generally listen to the same song more than once. For the experiments on CiteULike dataset, we recommend papers that have not been posted by the active user.

6.2 Experiment setup

Context On Last.fm dataset, the user context only includes domains of COUNTRY and AGE, while item context includes a domain of ARTIST. We notice that the original log tuples only consist of USER, SONG and TIMESTAMP domains. By transforming TIMESTAMP into different temporal features, we obtain several decision contexts, such as Day of Week, Weekend/Weekday, Month, Quarter and Time Slice (i.e., each time slice lasts for 6 hours). There is no user context in CiteULike dataset because of the lack of user information. Similarly, we transform TIMESTAMP into several temporal features, and TAGs of papers are considered as item contexts.

Paths For SPRW, as shown in Table 4, semantic paths specified empirically between users and items up to length 4 are presented in Table 4 , as well as the weight of each path.

Parameters λ in each experiment is set to 0.5. α in each layer is a tunable parameter. For Last.fm dataset, $\alpha_{user} = 0.005$, $\alpha_{decision} = 0.2$ and $\alpha_{item} = 0.01$. For CiteULike, $\alpha_{user} = 0.0$ (user context is unavailable), $\alpha_{decision} = 0.05$ and $\alpha_{item} = 0.01$. μ and ν in SPRW are all set to 0.4 on Last.fm and CiteULike.

[1] http://www.last.fm.com

[2] http://www.citeulike.org

6.3 Evaluation metrics

We use HitRatio@K [20], Mean Reciprocal Rank (MRR@K) [31] and Recall@K to evaluate the performance of our top-K recommendation.

Given a test case $< u, c_d, i >$ in test set $TEST$, where a user u accessed an item i in the decision context c_d, the recommendation method generates a ranked list of items $R(u, c_d)$, where $|R(u, c_d)| = K$. Then HitRatio@K is defined as follows:

$$HitRatio@K = \frac{1}{|TEST|} \sum_{<u,c_d,i>} I(i \in R(u, c_d)), \tag{10}$$

where $I(\cdot)$ is an indicator function.

We also measured the MRR@K for evaluating the rank of the target item i:

$$MRR@K = \frac{1}{|TEST|} \sum_{<u,c_d,i>} \frac{1}{rank(i)}, \tag{11}$$

where $rank(i)$ refers to the rank of target item i in $R(u, c_d)$.

Furthermore, we use Recall@K to evaluate the overall relevancy performance of recommendation methods:

$$Recall@K = \frac{1}{|TEST|} \sum_{<u,c_d>} \frac{|T(u, c_d) \cap R(u, c_d)|}{|T(u, c_d)|}, \tag{12}$$

where $T(u, c_d)$ is the set of items the user u accessed in context c_d.

As the definitions show, a larger value of HitRatio, MRR and Recall indicates a better performance.

6.4 Comparison methods

We evaluate the effectiveness of our method through comparing it with other following existing methods:

Frequency based (FreMax): A user independent method, which ranks the items by the times they were accessed. That is, FreMax generates the same list of items for any user.

User-based Collaborative Filtering (UserCF): A N-neighbor user-based collaborative filtering method, which uses Pearson Correlation Coefficient [6] as the user similarity measurement. The optimal value of N is 10 in experiments on CiteULike dataset, and N is 1 on Last.fm dataset.

ItemRank [13]: A random walk-based item scoring algorithm, which is performed on an item graph. The item graph is constructed by connecting two items if they were rated by at least one user.

GFREC [22]: A contextual bipartite graph-based method, which defines a recommendation factor set F, to transform a given log table into a bipartite graph. In our experiments, we use one of the best settings of F according to [22].

To validate the effectiveness of our ranking methods, we also include the following ranking methods for comparison, and the suffix 'M' means that the ranking method is performed in a MLCG:

PageRank-M (PR-M): Original PageRank algorithm is performed in MLCG for recommendation. In PR-M, all nodes are considered equally without bias.

CPRW-M: The proposed extended PageRank algorithm, which is presented in Section 4.2, to capture user preference by introducing a bias vector.

SPRW-M: The proposed ranking method, which incorporates semantics among paths in MLCG into random walk model. SPRW-M is provided in Section 4.3.

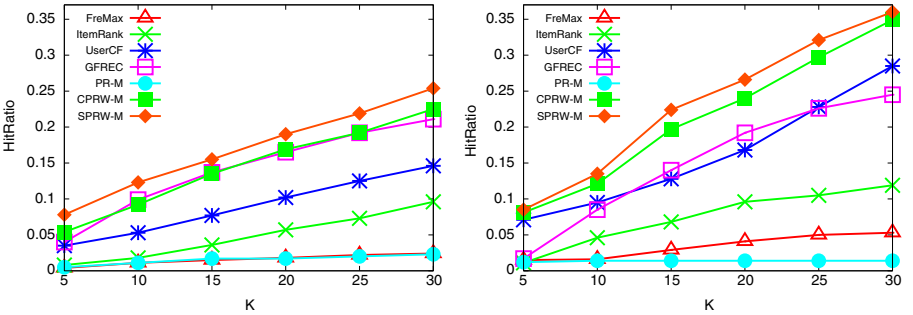
We do not consider item-based CF, since the number of items is much greater than users and user-based CF methods can provide more accurate recommendation. In addition, as [17] indicates, SVD methods only achieve slight improvement on implicit feedback datasets, thus, we do not compare our method with SVD-based methods. We transform our training data into a pseudo rating matrix by considering the normalized access count of a user on an item as the user's pseudo rating on the item, since UserCF and ItemRank require explicit feedback data.

6.5 Performance analysis

HitRatio@K analysis Figure 5 illustrates the HitRatio@K of our experiments in the two datasets. Lines with solid points represent different ranking methods in MLCG.

On Last.fm dataset, we summarize the following observations: (1) SPRW-M consistently performs better than other approaches, including CPRW-M, GFREC and UserCF, which demonstrates the effectiveness of the proposed framework. (2) Generally, the proposed framework (i.e., SPRW-M and CPRW-M) shows better/close performance than GFREC, in terms of HitRatio. In particular, the HitRatio of SPRW-M is 13-95 % higher than that of GFREC. (3) As expected, incorporating semantics of paths can contribute to higher HitRatio, as demonstrated by SPRW-M and CPRW-M. (4) PR-M obtains very close HitRatio with FreMax. It is because that PR-M works like FreMax since PR-M also prefers to the item nodes with more incoming links, i.e., popular songs tend to be ranked highly. The lowest HitRatio of FreMax and PR-M reveals the fact that users have their own preferences on songs, and would not be affected by popularity.

On CiteULike dataset, it is clear that our methods significantly outperform counterpart methods. Among the context-aware methods, SPRW-M achieves perceptible improvement, where HitRatio of SPRW-M is 4.0 times (K=5) and 60.0 % (K=15) higher than that of GFREC. It should be noticed that Top-K recommender systems benefit more from higher accuracy when K is small, such as user experience.



(a) Last.fm (b) CiteULike

Figure 5 Comparing the HitRatio@K of MLCG against *baseline* approaches

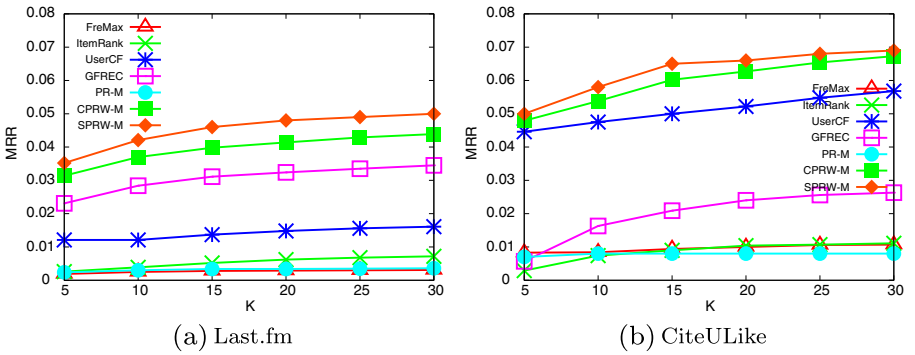


Figure 6 Comparing the MRR@K of MLCG against *baseline* approaches

MRR analysis MRR measures how highly ranked in the list is the target item. It is a particularly important measure of recommendation quality for domains that usually provide users with only few but valuable recommendations (i.e., the less-is-more effect [8]). SPRW-M and CPRW-M significantly outperform other methods in MRR as shown in Figure 6, that is, in the proposed framework, target items appear at the higher ranked positions, which is the desired behavior in a ranked list.

On Last.fm dataset, context-aware methods (i.e., SPRW-M, CPRW-M and GFREC) substantially excel over context-free methods, such as FreMax, ItemRank and UserCF. This verifies the importance of contextual information in improving recommendation accuracy. Further, among SPRW-M, CPRW-M and GFREC, we notice that SPRW-M generates the best result.

On CiteULike dataset, we notice that the MRR of GFREC is lower than that of UserCF. The reason is that superfluous combinations of multidimensional data bring connections, as well as noises, making GFREC much sensitive to contexts. Since one contextual value can have plenty of duplicates, tinny changes of it can significantly affect the recommendation result of GFREC. The proposed CPRW-M and SPRW-M address this issue by grouping

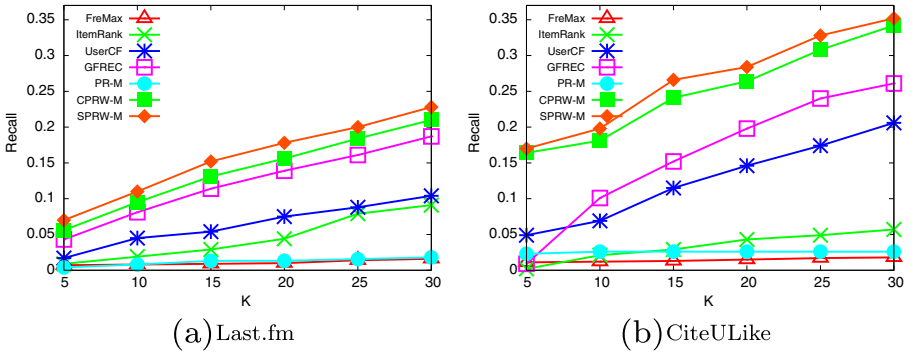


Figure 7 Comparing the Recall@K of MLCG against *baseline* approaches

contexts into the three layers to explore their relationships, and thus obtain an improvement of MRR by up to 30.0 % (K=15), compared to UserCF.

Recall analysis We give the results of recall in Figure 7. It can be seen that by incorporating contextual information, the proposed framework and GFREC have higher recall than other methods. And the improvements of CPRW-M and SPRW-M over the other comparison algorithms on both datasets are still clear. More specifically, while GFREC in general performs the best of the baseline methods, SPRW-M outperforms it with recall of up to 62.7 % (K=5) and 17.9 times (K=5) greater on Last.fm and CiteULike, respectively. Higher recall indicates that the algorithm returns more relevant results based on the instant situation and the active user. So, having a better understanding of the context of the active user, SPRW-M and CPRW-M provide more personalized recommendations than other methods.

6.6 Summary

In this subsection, we summarize the key conclusions we observe from the experimental results as follows:

- The proposed framework (i.e., SPRW-M and CPRW-M) outperforms other comparison methods, including GFREC, in both two scenarios, in terms of HitRatio, MRR and Recall. Specifically, this indicates: our methods are effective in capturing user preferences in various situations, obtaining most relevant items and highly ranking them as well.
- Between SPRW-M and CPRW-M, SPRW-M generates better recommendation. This coincides with the intuition that incorporating varying semantics of paths in MLCG can help provide promising recommendation performance.
- Being easily affected by tiny changes of contexts, GFREC can has a significant robustness failure rate in scenarios where users express relative stable preferences, such as article recommendation. By introducing historic behaviors and semantics, however, the proposed methods exhibit a more stable performance than GFREC.

7 Related work

The work in this paper closely relates to three research areas: recommendation with only implicit feedback data, context-aware recommendation and graph-based recommendation. In this section, we provide a brief review of the most related work in each of them.

7.1 Recommendation with implicit feedback

A typical recommendation approach to dealing with implicit feedback is introduced in [17]. It utilizes a least squares loss function and a weighted strategy to handle with observed and unobserved feedback. Further, Rendle et. al. [27] assume observed feedback as positive examples, and unobserved feedback as negative examples, and further propose Bayesian Personalized Ranking (BPR), which adopts a pair-wise loss function to learn user pair-wise preferences. By extending BPR, Pan et al. [26] develop Group Bayesian Personalized Ranking (GBPR) to consider group level preferences. However, all these methods consider only implicit user-item interaction information, while neglecting the contextual information.

7.2 Context-aware recommendation

Early work in context-aware recommender utilize contextual information for pre-processing or post-processing [1, 16]. Recent work has focused on integrating contextual information with the user-item relations [3, 28, 34].

In [16], they use contextual information about the user's task to improve the recommendation accuracy. Particularly, they concludes that the inclusion of knowledge on the user's task into recommendation in certain applications can lead to better performance. However, this method operates only within the traditional 2D $USER \times ITEM$ space. In [1], a reduction-based pre-filtering approach is proposed, which uses the user's prior item preferences to help match the current context for recommending items. Specifically, this approach is performed by the following two steps. First, it filters out ratings that do not match the current context. Then, traditional two-dimension recommendation algorithms are conducted on the reduced dataset. The drawback of this approach is the sparsity problem due to filtering the data. In [23], random decision trees are applied to partition the user-item-context matrix. Then submatrices are factorized for capturing user and item latent preferences.

In [3], a regression-based latent factor model is proposed to incorporate features and past interactions. In [28], a context-aware factorization machine is provided to simultaneously take context into account to enhance predictions. Xiong et al. [34] utilize tensor factorization for time-aware recommendation. In [19], High Order Singular Value Decomposition (HOSVD) is applied to factorize user-item-context space. However, these methods are infeasible for scenarios with only implicit feedback data. Shi et. al. [29] propose to directly optimize Mean Average Precision (MAP) for context-aware recommendation. But it fails to consider both user and item contexts into account.

7.3 Graph-based recommendation

Recommendation on a graph comprises two steps: constructing a graph from training data and ranking item nodes for a given user.

In [18], a two-layer graph model is proposed for book recommendation based on similarity among user nodes or item nodes. In [13], a graph is constructed by connecting two item nodes rated by at least one user, then the node scoring algorithm, ItemRank, ranks item nodes according to the active user's preference records. In [11], several Markov-chain model based quantities are considered, which provide similarities between any pair of nodes on a bipartite graph for recommendation. However, all the above approaches consider only $USER$ and $ITEM$ dimensions without additional contextual information.

In [5], the ContextWalk algorithm is provided for movie recommendation, which uses original random walk on a graph by considering the user and item context (e.g., actor, director, genre), but it ignores the decision context where a user chooses to watch a movie. As an example in [2], a user may have significantly different preferences on the genre of movies he/she wants to see when he/she is going out to a theater with his/her boyfriend/girlfriend as opposed to going with his/her parents. In [9], a query-centric random walk method on k-partite graph is provided for multidimensional recommendation. In [33], a graph-based method is presented that aims to improve recommendation quality by modeling user's long-term and short-term preferences. This method can deal with time information, but can not incorporate other types of contextual information, such as location and mood. In [22], a bipartite graph is proposed to model the interactions between users and items based on a recommendation factor set, F, where each recommendation factor $f \in F$ is defined as a combination of multidimensional data. Nodes corresponding to recommendation factors are

connected with item nodes. That is, the method utilizes duplicable dimensions to enhance the interactions, which may bring noises into recommendation. In addition, they do not provide principles to define recommendation factor set F, which is crucial to recommendation performance.

8 Conclusion and future work

In this paper, we investigate how to conduct accurate context-aware recommendation in scenarios where only implicit feedback is available. A graph-based model, Multi-Layer Context Graph (MLCG), is proposed. From implicit feedback data, MLCG utilizes contextual information to construct a layer for each type of contexts respectively, and models the decision making by users. In particular, our model emphasizes that users interact with items in an instant context. Based on MLCG, two novel ranking methods are presented for recommendation: (1) Context-aware Personalized Random Walk (CPRW) fuses user preferences and current decision contexts into random walk model; (2) Semantic Path-based Random Walk (SPRW) incorporates varying semantics among paths in a heterogeneous graph by extending CPRW. Finally, experiments based on two real-world datasets demonstrate that the effectiveness of the proposed method exceeds other existing ones in all evaluation metrics. We also highlight that our proposed framework is a generic framework that can allow various forms of recommendation, such as friend recommendation, item promotion and group recommendation.

Future work includes the distributed and incremental computation of the proposed ranking algorithms for scaling to large-scale data [4, 10].

Acknowledgments This work is partially supported by the National Natural Science Foundation of China (Grant No. 61272480, 71372188) and National Center for International Joint Research on E-Business Information Processing under Grant 2013B01035.

References

1. Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A.: Incorporating contextual information in recommender systems using a multidimensional approach. ACM Trans. Inf. Syst. (TOIS) **23**(1), 103–145 (2005)
2. Adomavicius, G., Tuzhilin, A.: Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Trans. Knowl. Data Eng. **17**(6), 734–749 (2005)
3. Agrawal, D., Chen, B.: Regression-based latent factor models. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge discovery and data mining (2009)
4. Bahmani, B., Chowdhury, A., Goel, A.: Fast incremental and personalized PageRank. Proc. VLDB Endowment **4**(3), 173–184 (2010)
5. Bogers, T.: Movie recommendation using random walks over the contextual graph. In: Proceedings of the 2nd International Workshop on Context-Aware Recommender Systems (2010)
6. Breese, J., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of UAI (1998)
7. Cao, J., Wu, Z., Mao, B., Zhang, Y.: Shilling attack detection utilizing semi-supervised learning method for collaborative recommender system. World Wide Web **16**(5–6), 729–748 (2013)
8. Chen, H., Karger, D.R.: Less is more: probabilistic models for retrieving fewer relevant documents. In: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval. ACM (2006)
9. Cheng, H., Tan, P., Sticklen, J., Punch, W.F.: Recommendation via query centered random walk on k-partite graph. In: Proceedingss of the Seventh IEEE Internation Conference on Data Mining (2007)

10. Das Sarma, A., Molla, A.R., Pandurangan, G., Upfal, E.: Fast distributed pagerank computation. Theor. Comput. Sci. (2014)
11. Fouss, F., Pirotte, A., Renders, J.-M., Saerens, M.: Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. IEEE Trans. Knowl. Data Eng. **19**(3), 355–369 (2007)
12. Gori, M., Pucci, A.: Research paper recommender systems: A random-walk based approach. Web Intelligence. IEEE/WIC/ACM Int. Conf. Web Intell., 778–781 (2006)
13. Gori, M., Pucci, A., Roma, V., Siena, I.: Itemrank: A random-walk based scoring algorithm for recommender engines. In: Proceedings of the 20th international joint conference on Artifical intelligence, pp. 2766–2771 (2007)
14. Gorla, J., Lathia, N., Robertson, S., Wang, J.: Probabilistic group recommendation via information matching. In: Proceedings of the 22nd international conference on World Wide Web. International World Wide Web Conferences Steering Committee (2013)
15. Haveliwala, T.H.: Topic-sensitive pagerank. In: Proceedings of the 11th international conference on World Wide Web, pp. 517–526 (2002)
16. Herlocker, J.L., Konstan, J.A.: Content-Independent Task-Focused Recommendation. IEEE Internet Comput. **5**(6), 40–47 (2001)
17. Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: Proceedings of the 8th IEEE International Conference on Data Mining, pp. 263–272 (2008)
18. Huang, Z., Chung, W., Ong, T.-H., Chen, H.: A graph-based recommender system for digital library. In: Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries, pp. 65–73 (2002)
19. Karatzoglou, A., Amatriain, X., Baltrunas, L.: Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In: Proceedings of the fourth ACM conference on Recommender systems, pp. 79–86. ACM Press (2010)
20. Karypis, G.: Evaluation of item-based top-n recommendation algorithms. In: Proceedings of the 10th international conference on Information and knowledge management, pp. 247–254 (2001)
21. Lao, N., Cohen, W.W.: Fast query execution for retriveal models based on path-constrained random walks. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 881–888 (2010)
22. Lee, S., Song, S.-I., Kahng, M., Lee, D., Lee, S.-G.: Random walk based entity ranking on graph for multidimensional recommendation. In: Proceedings of the fifth ACM conference on Recommender systems, pp. 93–100 (2011)
23. Liu, X., Aberer, K.: SoCo A social network aided context-aware recommender system. In: Proceedings of the 22nd International Wrold Wide Web Conference, pp. 787–784 (2013)
24. Lo, S., Lin, C.: Wmr–a graph-based algorithm for friend recommendation. In: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence. IEEE Computer Society (2006)
25. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank Citation Ranking: Bringing Order to the Web. Technical Report (1999)
26. Pan, W., Chen, L.: GBPR: Group preference based bayesian personalized ranking for one-class collaborative filtering. In: Proceedings of the Twenty-Third international joint conference on Artificial Intelligence, pp. 2691–2697 (2013)
27. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit feedback. In: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, pp. 452–461 (2009)
28. Rendle, S., Gantner, Z., Freudenthaler, C., Schmidt-Thieme, L.: Fast context-aware recommendations with factorization machines. In: Proceedings of the 34th International Conference on Reseach and Development in Information, pp. 635–644 (2011)
29. Shi, Y., Karatzoglou, A., Baltrunas, L.: TFMAP: Optimizing MAP for top-n context-aware recommendation. In: Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval. ACM, pp. 155–164 (2012)
30. Sun, Y., Han, J., Yan, X., Yu, P.S., Wu, T.: Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. In: Proceedings of internation conference on very large data base (2011)
31. Vallet, D., Cantador, I., Joemon, J.: Personalizing web search with folksonomy-based user and document profiles. In: Proceedings of the 32nd European conference on advances in information, pp. 420–431 (2010)
32. Wang, C., Blei, D.M.: Collaborative topic modeling for recommending scientific articles. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 448–456 (2011)

33. Xiang, L., Yuan, Q., Zhao, S., Chen, L., Zhang, X., Yang, Q., Sun, J.: Temporal recommendation on graphs via long-and short-term preference fusion. In: Proceedings of the 16th International Conference on Knowledge Discovery and Data mining, pp. 7230–732 (2010)
34. Xiong, L., Chen, X., Huang, T.-Y., Schneider, J., Carbonell, J..: Temporal collaborative filtering with bayesian probabilistic tensor factorization. In: Proceedings of SIAM international conference on data mining, pp. 211–222 (2010)
35. Yao, W., He, J., Huang, G., Cao, J., Zhang, Y.: Personalized Recommendation on Multi-Layer Context Graph. In: Web Information Systems EngineeringWISE 2013, pp. 135–148. Springer, Berlin Heidelberg (2013)
36. Yao, W., He, J., Huang, G., Zhang, Y.: SoRank: incorporating social information into learning to rank models for recommendation. In: Proceedings of the companion publication of the 23rd international conference on World wide web companion, pp. 409–410 (2014)