Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 613612, 10 pages
http://dx.doi.org/10.1155/2015/613612

Research Article

Context-Aware Recommendation via Graph-Based Contextual Modeling and Postfiltering

Hao Wu, Kun Yue, Xiaoxin Liu, Yijian Pei, and Bo Li

School of Information Science and Engineering, Yunnan University, Kunming 650091, China

Correspondence should be addressed to Hao Wu; haowu@ynu.edu.cn

Received 18 December 2014; Revised 17 February 2015; Accepted 27 February 2015

Academic Editor: Houbing Song

Context-aware recommender systems generate more relevant recommendations by adapting them to the specific contextual situation of the user and have become one of the most active research areas in the recommender systems. However, there remains a key issue as how contextual information can be used to create intelligent and useful recommender systems. To assist the development and use of context-aware recommendation capabilities, we propose a graph-based framework to model and incorporate contextual information into the recommendation process in an advantageous way. A contextual graph-based relevance measure (CGR) is specifically designed to assess the potential relevance between the target user and the items further used to make an item recommendation. We also propose a probabilistic-based postfiltering strategy to refine the recommendation results as contextual conditions are explicitly given in a query. Depending on the experimental results on the two datasets, the CGR-based method is much superior to the traditional collaborative filtering methods, and the proposed postfiltering method is much effective in context-aware recommendation scenario.

1. Introduction

Recommender systems [1] are popular tools for assisting customers in navigating through huge archives and alleviating the side effects of information overload in big data era. Most of the recommender systems focus on mining associations between users and items (movies, music, Web information, goods, etc.) and a few works consider the context which the items are associated with [2]. However, in ubiquitous computing scenario, users can access and exchange information at anytime, anyplace, and in any way, and context-awareness is the fundamental objective of ubiquitous computing [3]. User preferences may be changed by switching contexts (such as time, location, surrounding people, emotion, devices, network conditions, etc.) and solely relying on the user-item association does not lead to a valid recommendation. For instance, some users prefer the morning instead of noon to be recommended appropriate news; some users in the tourist are likely to be recommended suitable surrounding restaurants or shopping malls; and some users prefer to watch romantic movies at a theater if they are together with boyfriends/girlfriends. Consequently, the effectiveness of recommendations will be affected if these contextual factors are not considered into the recommender systems. Context-aware recommender systems (CARS) generate more relevant recommendations by adapting them to the specific contextual situation of the user and have gradually become one of the most active research areas in the recommended systems [2].

Nowadays, there are three recognized paradigms for incorporating contextual information into the recommendation process: contextual prefiltering, contextual postfiltering, and contextual modeling [2, 4–6]. Among them, prefiltering is particularly attractive due to a straightforward justification: when context matters, use in the recommendation process only the data acquired in the same contextual situation of the target user, because only this data is relevant for predicting user preferences. However, prefiltering has not always been the best choice. Its major limitation is the difficulty to obtain enough ratings in all possible contextual conditions to build a strong and applicable prediction model [2, 7–9]. On the contrary, postfiltering modeling and contextual modeling can directly exploit available contextual factors to enrich data used for prediction, increasing in turn the

effectiveness of the recommendations. Nevertheless, how to model contextual factors into a recommendation process and how to combine different strategies of CARS are still challenging tasks, and many issues are yet to be taken up. To cope with this issue, we propose a unified graph-based contextual modeling framework to incorporate contextual information in an advantageous way. A contextual graph-based relevance measure and a probabilistic postfiltering strategy are designed to facilitate the development and use of context-aware recommendation capabilities. Experimental results against two real-world datasets show the merits of all the proposed approaches in context-aware recommendation.

The rest of the paper is structured as follows. Section 2 introduces our models and the corresponding recommendation approaches for CARS. In Section 3, experiments against two datasets from different domains are conducted to watch the effectiveness of the proposed methods. Section 4 reviews related works and makes some discussions. Finally, we conclude the works and point to future directions.

2. The Proposed Methods

2.1. Graph-Based Contextual Modeling. CARS address modeling and predicting user tastes and preferences by incorporating existing contextual information into the recommendation process as explicit additional categories of data. These long-term preferences and tastes are generally expressed as ratings and are modeled as the function of not only items and users, but also the context. Adomavicius et al. [2] formally define the recommending process in CARS as $R : User \times Item \times Context \rightarrow Rating$, where *User*, *Item*, and *Rating* are the domains of users, items, and ratings, respectively, and Context specifies the contextual information associated with the application. Once the function R is estimated for the whole $User \times Item \times Context$ space, a recommender system can suggest the highest-rated item for users. Note that context factors indicate the upper-level contextual concepts (e.g., daytype and location), while context conditions represent the combination of contextual concept instances, for example, *Monday, Office Room*.

Distinct from building a prediction model, we examine the context-aware recommendation as a searching problem to find interesting items for a user given a context graph. Formally, suppose $G = \{V, E\}$ is a context graph. The vertex set V are divided into several distinct sets, including a set of users U, a set of items I, a set of attributes A, and a set of contexts C. In particular, we distinguish the contextual information of C from the static information of A of users or items, considering that the attributes basically do not change, while the context factors often change, over different ratings. The edge set E consists of the existing connections of the Cartesian product: $V \times V$. Edges with diverse types have distinct semantics. For example, $U \times A$ connects users and their attributes; $U \times I$ means the users have accessed items; $U \times C$ represents that the users have activities given the contextual condition. The situation for items is equal to that of users. The context graph G can be represented as an adjacent matrix M (as shown in Table 1), where submatrixes are all configured as symmetric (e.g., UI^{T} is the transport matrix of UI).

Table 1: A matrix representation of contextual user-item interaction.

	Users	Items	Contexts	Attributes
Users	UU	UI	UC	UA
Items	UI^{T}	0	IC	IA
Contexts	UC^{T}	IC^{T}	0	0
Attributes	UA^{T}	IA^{T}	0	0

Considering a random search that starts from node i in G, the search iteratively transmits to its neighborhood j with the probability that is proportional to the edge weights S_{ij}. Also, at each step, it has some probability $1 - d$ (the default setting is $d = 0.85$) to return to the node i. The relevance score of node j with respect to node i is described as the steady-state probability; the search will finally stay at node j. The steady probability can be gotten using the power method as in the following equation until convergence:

$$\mathbf{P}^{(n)} = dS^{\mathrm{T}}\mathbf{P}^{(n-1)} + (1 - d)\,\mathbf{e}, \tag{1}$$

where the row-normalized matrix S is the transitivity matrix of random walks and vector $\mathbf{e}$ is a personalized vector that may represent the interests of a particular user. Such a theory is called the random walks with restarts (also known as Personalized PageRank) and has attracted many attentions in various recommendation scenes [10–12].

A contextualized data graph adheres to the schema of G. In the data graph, given the source node i, the target node j and the relationship type L of an edge, the transitivity matrix is initialized as follows:

$$S_{ij} = \begin{cases} w(L)\dfrac{w(i,j,L)}{\sum_k w(i,k,L)} & \text{if } w(i,j,L) > 0, \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$w(L)$ is the weight of the relationship typed L in the schema of G and $w(i,k,L)$ represents the weights of out-edges typed L of i in the data graph. Since the graph G is symmetric, $w(j,i,L) = w(i,j,L)$. Here, we examine the weights of edges in two aspects: (I) edge with different types has unusual semantics; thus, it may affect the selection of nodes during random walks. For instance, when searching items of interest, a user may wish to consult his/her friends, other than viewing the ratings of the items. We use $w(L)$ to impose this influence in the process of the user's random walks. (II) edges must be assigned weights to reflect users usage data. For instance, a user may have contacted an item several times or access separate items under the same context condition. Thus, we further assign a weight of $w(i,j,L)$ to the edges.

With respect to the types of edges L and the weights $w(L)$, they can be defined at different levels of concepts. We first define L at the highest level according to Table 1, resulting in 11 types of L. To make a difference among numerous attributes or context factors, we further extend the definition of L and $w(L)$ to a fine-grained level of concept. For instance, given a set of item attributes, $A = \{A_1, \ldots, A_i, \ldots, A_n\}$, we can

Input: U, I, A, C, d, u, I_u, the empty G, the index of vertex $ind(\cdot)$;
Output: The CGR vector of G;
(1) Add all elements in U, I, A and C as graph nodes to G and assign an unique index for each node;
(2) Add edges to G and assign an type label L for each edge;
(3) **for all** $i \in V$ **do**
(4) **for all** $j \in V$ **do**
(5) **if** There exists an edge from i to j **then**
(6) Get the edge type L and the weight $w(L)$;
(7) $S_{ij} = w(L)\dfrac{w(i, j, L)}{\sum_k w(i, k, L)}$;
(8) **else**
(9) $S_{ij} = 0$;
(10) **end if**
(11) **end for**
(12) **for all** $j \in V$ **do**
(13) $S_{ij} = \dfrac{S_{ij}}{\sum_k S_{ik}}$;
(14) **end for**
(15) **end for**
(16) **for all** $j \in V$ **do**
(17) **if** $j == ind(u)$ **then**
(18) $\mathbf{e}'_j = 1$
(19) **else**
(20) $\mathbf{e}'_j = 0$
(21) **end if**
(22) **end for**
(23) Compute $\mathbf{P_1}^{(n)} = dS^{\mathrm{T}}\mathbf{P_1}^{(n-1)} + (1-d)\mathbf{e}'$;
(24) **for all** $j \in V$ **do**
(25) **if** $j \in I_u$ **then**
(26) $\mathbf{e}''_j = \dfrac{1}{|I_u|}$
(27) **else**
(28) $\mathbf{e}''_j = 0$
(29) **end if**
(30) **end for**
(31) Compute $\mathbf{P_2}^{(n)} = dS^{\mathrm{T}}\mathbf{P_2}^{(n-1)} + (1-d)\mathbf{e}''$;
(32) **return** $(\mathbf{P_1} + \mathbf{P_2})/2$;

Algorithm 1: Contextual graph-based relevance.

partition $IA = \bigcup_i^n IA_i$ and define $w(IA) = \sum_i^n w(IA_i)$, where $w(IA_i)$ is the weight of the ith attribute. By this, we can provide a more scalable framework for the contextual modeling.

In the scenario of recommendation, to estimate the likelihood of an unseen item $i \in I$ to be accessed by $u \in U$, $P(i \mid u)$, we suggest combining two ranking values of i, by biasing different personalized vectors, shown as follows:

$$P(i \mid u) = \frac{\mathbf{P_1}(i) + \mathbf{P_2}(i)}{2}, \tag{3}$$

where $\mathbf{P_1}(i)$ and $\mathbf{P_2}(i)$ is the ranking values of i with respect to $\mathbf{e}'$ and $\mathbf{e}''$, respectively. For the calculation of $\mathbf{P_1}$, we let $\mathbf{e}'_j = 1$, if j is the index of u. For the calculation of $\mathbf{P_2}$, we let $\mathbf{e}''_j = 1/|I_u|$, where $j \in I_u$ and I_u is the set of items has been accessed by u. In both cases, the remaining elements of $\mathbf{e}$ are all settled at zero. Given the contextual graph G, $\mathbf{P_1}$ biases the items interested in the neighborhood of the current user u, while $\mathbf{P_2}$ prefers to the items similar to the set I_u. Such a facile combination has the advantages of both worlds, since it resembles the recommendation combination of a user-based CF and an item-based CF, which have been proved as a better way to increase the accuracy of recommendations. We called the user-item relevance achieved by newly proposed method as CGR (contextual graph-based relevance) to distinguish it from our former work [13]. Details of CGR method are presented as Algorithm 1.

2.2. Context-Aware Postfiltering. With respect to the CGR, all contextual information and attributes of users or items can be taken into account in estimating user-item relevance. However, there remain some significant issues not considered. On one hand, the rating information which explicitly represents users' preferences to items is not exploited. On the other hand, it is very difficult for the CGR model to deal with the query in which context factors are explicitly specified.

To cope with these problems in CARS, we further submit a postfitering strategy. Given an instance of the context factors $C = \{c_1, \ldots, c_i, \ldots, c_n\}$, the likelihood of an unseen item $i \in I$ to be accessed by $u \in U$ can be estimated as follows:

$$P(i \mid u, C) = P(i \mid C)\, P(i \mid u) \tag{4}$$

if the random variables u and C are assumed to be independent. For the estimation of $P(i \mid u)$, we can use CGR-based method or collaborative filtering approaches, or their combinations. Let $P_{\mathrm{CGR}}(i \mid u)$ and $P_{\mathrm{CF}}(i \mid u)$ be the CGR-based and CF-based relevance of u and i, respectively; $P_{\mathrm{CGR}}(i \mid u)$ can be estimated using (3), and $P_{\mathrm{CF}}(i \mid u)$ can be given in $r(u,i)/\max r(\cdot)$, and $\max r(\cdot)$ is the maximal value of ratings in a dataset (e.g., 5). $r(u,i)$ is the predicted rating based a CF algorithm. $P(i \mid C)$ can be estimated as follows:

$$P(i \mid C) \propto \sum_{c_k \in C} P(i \mid c_k)\, P(c_k \mid C) = \frac{1}{|C|} \sum_{c_k \in C} \frac{\mathrm{freq}(i, c_k)}{\sum_j \mathrm{freq}(j, c_k)}, \tag{5}$$

where $\mathrm{freq}(i, c_k)$ is the occurrence frequency of item i given the context condition c_k, and $P(c_k \mid C)$ takes $1/\|C\|$ for simplicity. This probabilistic-based method reranks a candidate list of items that has been made by a recommendation model. Thus, it can be viewed as a postfiltering model of CARS.

3. Experiments

3.1. Datasets. We use two datasets from different domains to evaluate the presented methods. The first one is the LDOS-CoMoDa, which is specially designed for context-aware personalization research [14, 15]. The dataset comprises 2296 rating records from 121 users to 1232 items, accompanying contextual factors. Content items are movies, while the item consumption device is a personal computer with a web-based application to acquire the user's context [14]. Each record contains 30 variables including a user, an item, a rating, 12 context factors, 4 attributes of users, and 11 attributes of items. Context factors associated with ratings include *time*, *daytype*, *season*, *location*, *weather*, *social*, *endEmo*, *dominantEmo*, *mood*, *physical*, *decision*, and *interaction*. Also each context factor may have multiple discrete values, amounting to 49 context conditions. For example, there are three values (*Working day*, *Weekend,* and *Holiday*) as for the *daytype* conditions and seven values (*Alone*, *My partner*, *Friends*, *Colleagues*, *Parents*, *Public*, and *My family*) with respect to the *social* condition. Other variables are general user attributes (age, sex, city, and country) and movie attributes (director, movieCountry, movieLanuage, movieYear, genre1, genre2, genre3, actor1, actor2, actor3, and budget).

The second dataset is scripted from http://www.tripadvisor.com/ which is a tourism website that advises trips, locations, and activities for users and contains a social component, which allows lots of elements to be reviewed [16]. The dataset is about hotel reviews and consists of 4669 ratings from 1202 users to 1890 hotels [9]. The attributes of users are *state* and *timezone* and the attributes of hotels are *city*, *state,* and *timezone*. And there is only one context, *trip type*, assigned to each rating showing the types of trips that the user suggests for this hotel. For this context, the user can select a subset of five possible values: *Family*, *Couples*, *Business*, *Solo travel*, and *Friends*. The statistics of both datasets are presented in Table 2, where we can see that the TripAdvisor dataset is considerably sparser than the LDOS-CoMoDa dataset.

Table 2: The statistics of datasets.

Dataset	LDOS-CoMoDa	TripAdvisor
Users	121	1202
Items	1232	1890
Ratings	2296	4669
Rating scales	1–5	1–5
Context factors	12	1
User attributes	4	2
Item attributes	7	3
Sparsity	98.46%	99.79%

3.2. Performance Metrics. Along with the progress of recommendation techniques, various metrics have been applied to measure the accuracy of recommendations, including statistical accuracy metrics and decision-support measures. Since we focus on recommending top-N items instead of rating prediction of items, Precision@N, Recall@N, and nDCG@N are selected to evaluate the recommendation accuracy.

Given a rank list of recommended items, Precision@N is the fraction of recommended items that are relevant in the top-N position as follows:

$$P@N = \frac{\left|\{relevant\ items\} \cap \{top\text{-}N\ items\}\right|}{N}, \tag{6}$$

while Recall@N is the fraction of relevant items returned in the top-N position, to the true number of relevant items that should have been returned

$$R@N = \frac{\left|\{relevant\ items\} \cap \{top\text{-}N\ items\}\right|}{|\{relevant\ items\}|}. \tag{7}$$

Discounted cumulative gain is a measure that gives more weight to highly ranked resources and incorporates different relevance levels through different gain values. One popular variant is described as

$$\mathrm{DCG@}N = \sum_{i=1}^{N} \frac{2^{rel(i)} - 1}{\log_2(i+1)} \mathrm{IDCG@}N = \sum_{i=1}^{K} \frac{1}{\log_2(i+1)}. \tag{8}$$

Here, the relevance level of ith item $rel(i)$ depends on just a binary notion of relevance, whether the items are relevant or not to the target user. We use $rel(i) = 1$ for relevant items if $r(u,i) \geq 3$ and $rel(i) = 0$ for irrelevant items if $r(u,i) < 3$. With the definition of DCG@N, we can find the ideal DCG@N (IDCG@N) as (8), where K is the number of relevant items inside the top-N recommendations. We use the normalized DCG@N (nDCG@N = DCG@N/IDCG@N) to evaluate the recommendation accuracy [13].

Table 3: The statistics of training and test datasets.

	LDOS-CoMoDa	TripAdvisor
Splitting ratio	80% : 20%	70% : 30%
Ratings for training	1833	2887
Ratings for test	228	1339
Users for test	60	963
Relevant items per query	3.80	1.39

3.3. Evaluated Methods. UPCC and IPCC: the methods are the classical user-based collaborative filtering and the item-based collaborative filtering [1] and adopt the Pearson correlation coefficient (PCC) to define the similarity between two users or two items.

SVD++ [17] and CAMF [18]: SVD++ is a well-known latent factor model that comprises an alternative approach to collaborative filtering with the more holistic goal to uncover latent features that explain observed ratings. CAMF is an extension of classical matrix factorization (MF) approach for taking into account contextual information in the rating prediction. Similar to SVD++, it is also induced by SVD on the user-item ratings matrix. Learning the prediction models for both of them is solved using stochastic gradient descent. We fix dimension of latent factors as 20; other parameters are optimized to realize the best predicting performance.

CGR. Candidate items are recommended to the user according to their CGR-based relevances. All available information in contextual graph G is considered; however, for simplicity, we let $w(T) = 1$ for all the weights of relationship types in the schema of G. Note that, for all collaborative filtering based models (UPCC, IPCC, SVD++, and CAMF), candidates are sorted by their predicted ratings. For CGR-based models, the candidates are ranked according to their relevance scores.

3.4. Experimental Results on Contextual Modeling. In this section, we conduct experiments to observe the recommendation performance of graph-based contextual modeling approach. Under this case, each query has no prescribed context conditions to prefilter or postfilter the recommended candidates. The LDOS-CoMoDa dataset is randomly divided into the training dataset and the test dataset by a ratio of 80% : 20% and the TripAdvisor dataset is divided into two corresponding parts by the ratio of 70% : 30%. Since we do not focus on the cold-start problem of recommendation, new users and new items in the test dataset are not considered during the recommendation process. Statistics of training dataset and refined test dataset are presented in Table 3.

Firstly, we observe the recommendation accuracy of the CGR using five separate schemes of context graph. By this, we can identify which of the components work best for the task of top-N item recommendation. The results are presented in Figures 1 and 2. *ALL* simultaneously considers contextual information, attributes of users and items and use data of users to items. *ALL-UC-IC* corresponds the case where contextual information are not considered; namely, $w(UC) = 0$, $w(UC^{\mathrm{T}}) = 0$, $w(IC) = 0$, and $w(IC^{\mathrm{T}}) = 0$. Similarly, *ALL-UI*, *ALL-UA,* and *ALL-IA,* respectively, exclude the direct connections of users-items, attributes of users, and attributes of items. According to Figure 1, *ALL-UA* performs in a way much better than the other cases; it shows that the user attributes have negative effects on the LDOS-CoMoDa datasets. The reason lies in that users' attributes (age, sex, city, and country) in the LDOS-CoMoDa dataset are not discriminative; thus, the neighborhoods of users found by a random walk through the links of users' attributes are not consistent in their spectrums of interest. As for the TripAdvisor dataset, all contextual information, attributes of users, and items seem to be bringing positive effects to recommendations. Further, by combining experimental results from both datasets, we find that the connections of users-items are a key factor for finding relevant recommendations, followed by the attributes of items which seem to be more important than the attributes of users and the contextual information.

Next, we compare the CGR with other baseline methods. We are focused on top-10 and top-20 recommendations, as users are more concerned about the candidates being at the front of the recommendation list. Experimental results on both datsets are presented in Figure 3. For the LDOS-CoMoDa dataset, the CGR model significantly outperforms UPCC, IPCC, SVD++, and CAMF in all accurate metrics, and it improves them by at least 80% in all performance metrics. Similar trends are observed in the TripAdvisor dataset, where the CGR model obviously beats all the others. In particular, it outperforms them by at least 200% in all performance indicators. Since the TripAdvisor dataset is much sparser than the LDOS-CoMoDa dataset, we think that CGR-model is stronger than traditional CF-like recommenders and more suitable to cope with the problem of data sparsity. On one hand, CGR-based models perform in a more superior way when data is sparse. On the other hand, it offers an easier way to integrate various contextual factors to improve recommendation performance, which is still thought as a challenging task in other ways.

3.5. Experimental Results on Context-Aware Postfiltering. In this section, we did the experiment with the postfiltering model for CARS. For this, we reuse the training and the test datasets provided in Table 3. Each query in this experiment is linked to a context condition, and the context factors associated with a pertinent item must match the context condition given in the query. Context factors considered for the LDOS-CoMoDa dataset are *daytype* and *social*, which are two important factors affecting a user's choice for watching a movie. The context factor took into account for the TripAdvisor dataset is only *TripType*. By splitting items using selected context factors, we further obtain 225 queries in the LDOS-CoMoDa test dataset and 1102 queries in the TripAdvisor test dataset. The average numbers of relevant items per query are correspondingly 1.01 and 1.22. Obviously, every context-aware query only has about one relevant item. We study recommendation effectiveness of evaluated methods with or without postfiltering strategies. The results are presented in Table 4, where the best values for every indicator are underlined. Since the number of relevant items per query is

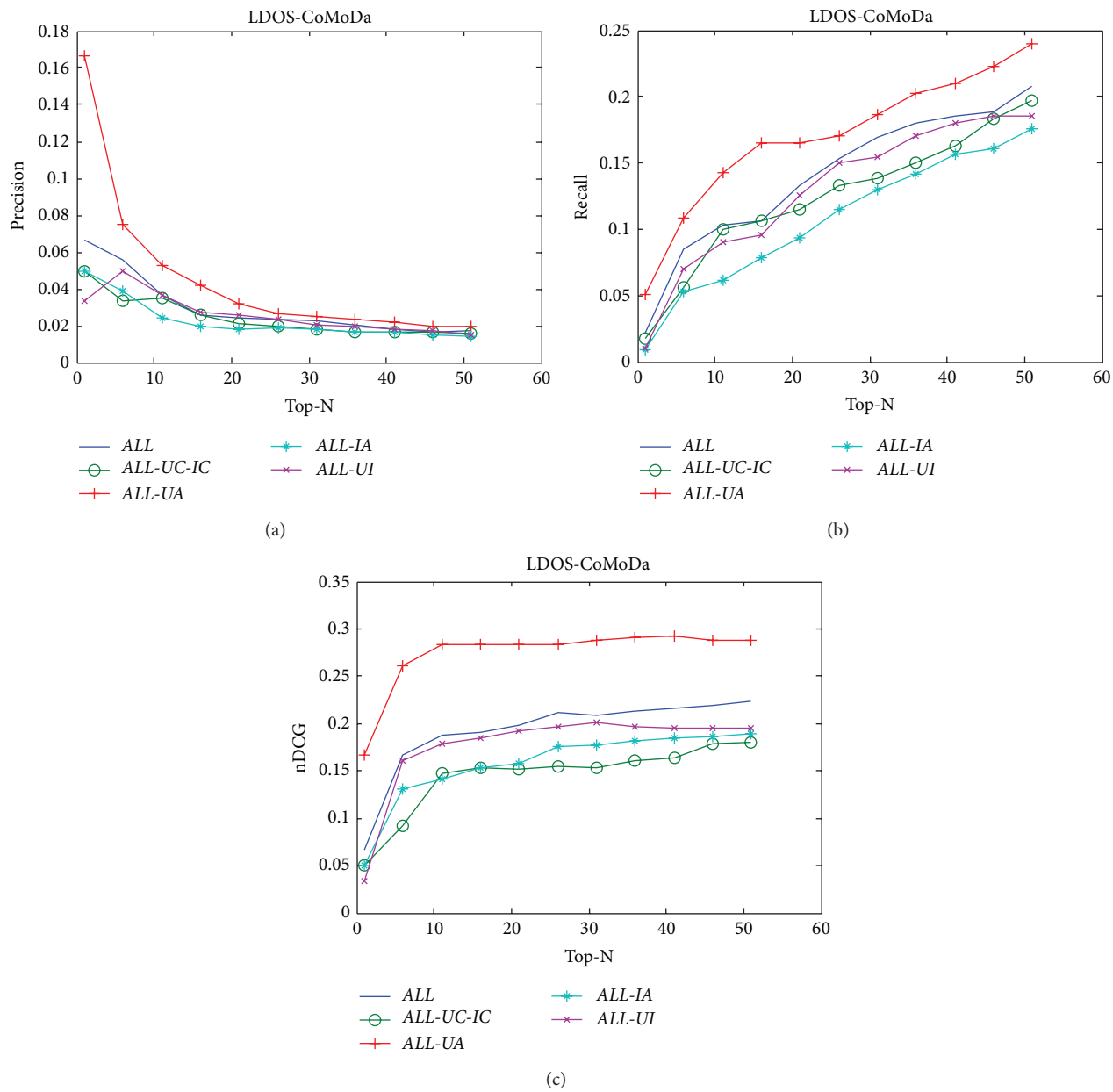

Figure 1: Recommendation accuracy of CGR using different graphic schema on the LDOS-CoMoDa dataset.

extremely small, we consider the recall to evaluate recovery ability of recommenders.

As for the case, without using postfiltering (w/o), we can see that the CGR model largely improve all the other models on both datasets. This outcome is very consistent with prior experimental results (see Figures 1 and 2). UPCC, IPCC, and SVD++ achieve comparable performances on both datasets under this case. It is worth emphasizing that CAMF is much better than UPCC, IPCC, and SVD++ in terms of $R@10$ and $R@20$ on the LDOS-CoMoDa dataset, showing its merits in context-aware recommendations. As for using postfiltering (PoF), significant improvements are observed in four CF-based methods against both datasets. The gains reach to at least 103% across all indicators on the LDOS-CoMoDa dataset and at least 16% in terms of $R@20$ and $R@30$ against the TripAdvisor dataset. However, the case for CGR model is very unique; combining post-filtering in it degrades the recall of recommendations particularly in the TripAdvisor dataset. Therefore, it is not suitable for the CGR to exploit the PoF strategy directly.

3.6. Experimental Results on Hybrid Model. Since directly combining the PoF strategy with the CGR model reduces the quality of recommendations, we suggest a linear combination of a CGR-based and a CF-based relevance measure between u and i, aiming at having both worlds. It is shown as follows:

$$P(i \mid C, u) = \alpha P_{\mathrm{CGR}}(i \mid C, u) + (1 - \alpha) P_{\mathrm{CF}}(i \mid C, u), \quad (9)$$

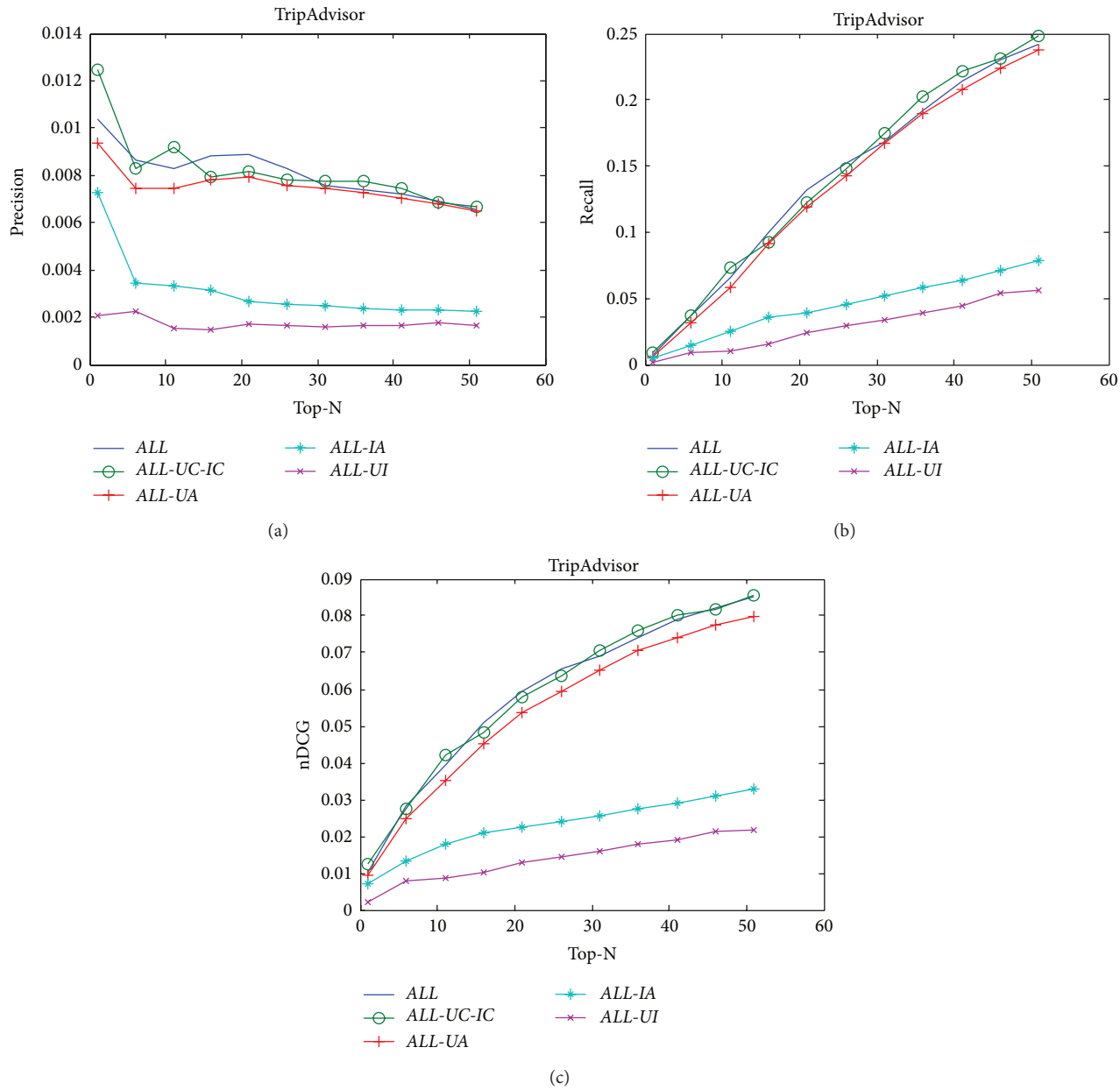

Figure 2: Recommendation accuracy of CGR using different graphic schema on the TripAdvisor dataset.

where α is a coefficient to combine the normalized values of two terms. Both terms are computed using (4)-(5) and scaled using max-min normalization. We perform an experiment against the LDOC-CoMoDa dataset to observe the effectiveness of novel hybrid model, where we take the IPCC as the rating predictor. The results are shown in Figure 4, where a notable improvement can be observed as the coefficient α increases. The gains for R@30 and R@20 can reach about 2.4% and 6.0%, respectively. This confirms our speculation that CGR-based method is useful to enhance recommendation effectiveness along with the additional methods. Since CGR is dominant and much better than counterparts against the TripAdvisor dataset, using a hybrid in this dataset cannot achieve additional profits; we omit the results with respect to the TripAdvisor dataset.

4. Related Works

Context-aware recommender systems represent an increasingly active and highly problem-rich research area, especially for pervasive computing environments, where individual users consuming various products or services are always associated with rich contexts. A lot of existing research on CARS addressed different aspects of the problem, such as general-purpose algorithms, evaluation protocols, and

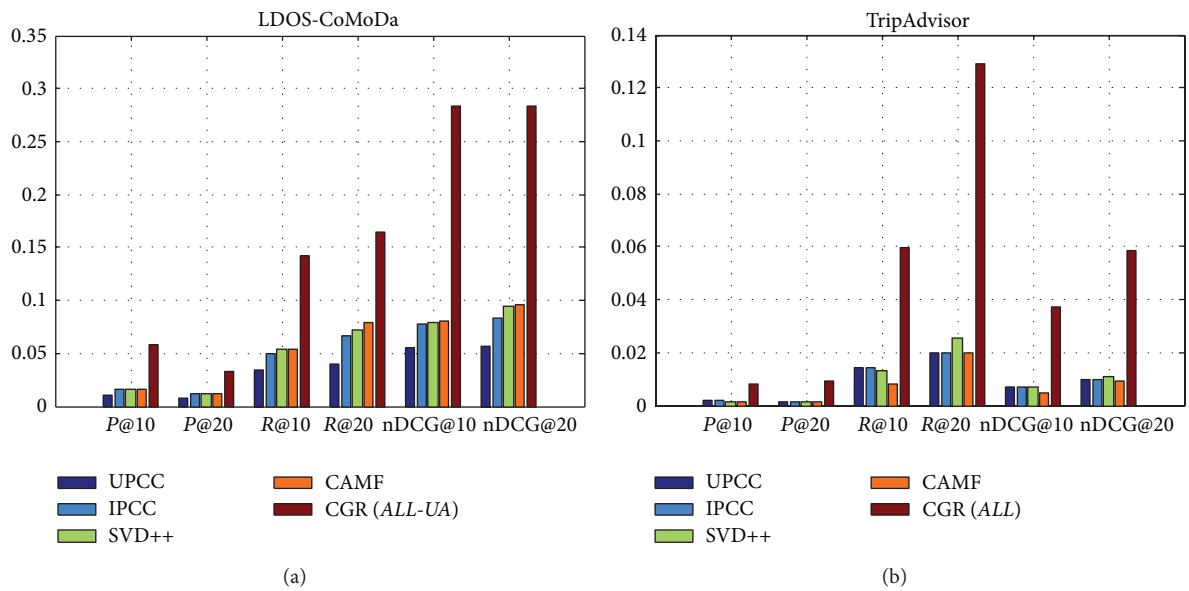

Figure 3: Recommendation performance comparison between the CGR and the baseline methods.

Table 4: Recall performance of different recommenders in combination with or without postfiltering strategy.

	LDOS-CoMoDa			TripAdvisor		
	R@10	*R*@20	*R*@30	*R*@10	*R*@20	*R*@30
UPCC						
w/o	0.0311	0.0400	0.0577	0.0163	0.0241	0.0318
PoF	0.1055 (+239%)	0.1560 (+290%)	0.1789 (+210%)	0.0150 (−8%)	0.0318 (+32.0%)	0.0422 (+32.7%)
IPCC						
w/o	0.0444	0.0667	0.0948	0.0163	0.0241	0.0318
PoF	0.1238 (+179%)	0.1606 (+141%)	0.1927 (+103%)	0.0150 (−8%)	0.0318 (+32.0%)	0.0422 (+32.7%)
SVD++						
w/o	0.0444	0.0637	0.0908	0.0133	0.0252	0.0350
PoF	0.1174 (+164%)	0.1697 (+166%)	0.1881 (+107%)	0.0177 (+33.1%)	0.0295 (+17.1%)	0.0408 (+16.6%)
CAMF						
w/o	0.0543	0.0765	0.0849	0.0104	0.0204	0.0340
PoF	0.1147 (+111%)	0.1651 (+116%)	0.1972 (+132%)	0.0182 (+75.0%)	0.0295 (+44.6%)	0.0413 (+21.5%)
CGR						
w/o	**0.1556**	**0.1748**	0.1970	**0.0594**	**0.1310**	**0.1638**
PoF	0.1330 (−17.0%)	0.1743 (−0.01%)	**0.2018** (+2.4%)	0.0445 (−33.5%)	0.0717 (−82.7%)	0.0907 (−80.6%)

domain-specific engineering. Instead of providing a detailed survey of CARS, we only think of the algorithmic principles most pertinent to our approaches. As for a recent survey on CARS and the discussion of probable directions, users can refer to the work [2]. For the comparisons of different CARS, users can refer to works [4–6].

From the algorithmic perspective, there are currently three recognized paradigms for incorporating contextual information into the recommendation process: (i) contextual prefiltering: context is used for selecting the relevant set of rating data before computing predictions. For contextual prefiltering, many recommendation models, such as collaborative filtering and matrix factorization, can be directly utilized before or after computing predictions [2]. Prefiltering is particularly appealing due to its straight-forward and flexibility of justification. Some strategies have been proposed, such as splitting ways [7], context relaxation [9], and semantic filtering [8], to acquire more pertinent data for predicting user preferences in the same contextual situation of the target user. (ii) Contextual postfiltering: context is used to adjust predictions generated by a context-free 2D prediction model. Distinct from prefiltering, the reranking strategies are required for the recommendation list already obtained. (iii) Contextual modeling: contextual information is directly incorporated in the prediction model, usually by generalizing the 2D prediction model to an n-dimensional one. As for contextual modeling, more complex algorithms are typically explored. Tensor decomposition [19] aims to

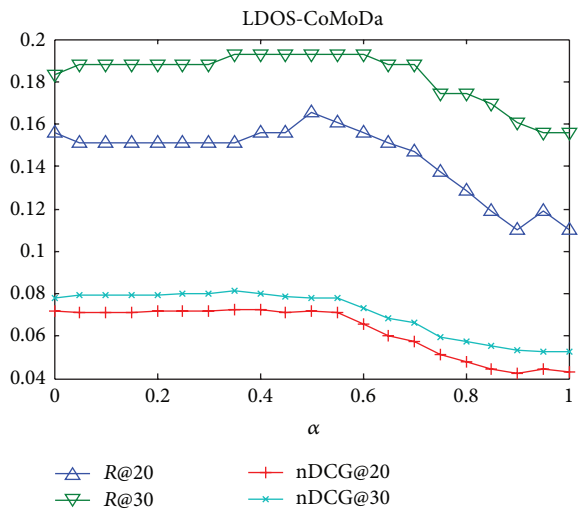

Figure 4: Recall of the hybrid model against LDOS-CoMoDa dataset.

factorize the tensor over user, items, and all categorical context variables directly to improve prediction accuracy. Nonetheless, with the increased contextual factors, efficiency becomes the bottleneck of the method [20]. In contrast, Baltrunas et al. present a novel MF-based context-aware recommendation algorithm [18], which models the interaction of contextual factors with item ratings introducing additional model parameters. The proposed solution provides comparable results to the tensor decomposition based approaches, with the merits of smaller computational cost. However, incorporating contextual factors into existing CF-based or MF-based algorithms is always a challenging task, due to the inherent limits of scalability and flexibility of these algorithms.

Another kind of approach similar to us is graph-based, which incorporates contextual information into recommender systems in a flexible and scalable way. Graph-based methods can mitigate data sparsity problem utilizing rich contextual information [10, 11, 21, 22]. However, presented graph models are not grounding on a unified modeling framework. Also these domain-specific models have not been tested across different scenarios of applications. Reporting to them, our works design a general-purpose framework to facilitate the development and use of context-aware recommendation capabilities. We put forward a unified graph-based contextual modeling framework to incorporate contexts in the prediction model and specially design the CGR measure to estimate the relevance between the target user and the items. Also, we put forward a probabilistic postfiltering strategy to improve the effectiveness of context-aware recommendation. In addition, we proved that the newly proposed methods can work with additional methods to continue to improve the recommendation performance.

5. Conclusions and Future Works

We have proposed a graph-based framework for incorporating contextual information in the recommendation process. Also, we present a probabilistic reranking strategy to filter recommendation consequences given the contextual conditions. Experimental results have illustrated that all the proposed approaches are helpful to facilitate the development and use of context-aware recommendation capabilities. In addition, our proposed model can be viewed as a component and entered into a hybrid model to enhance effectiveness of CARS.

Some issues have yet to be investigated in the future. At first, we want to examine proposed methods on additional context-aware datasets and compare them with more powerful counterparts. We currently utilize the contextual graph in a simplified manner as we do not recognise on the importance of different contextual factors and relationships. In the subsequent phase, we propose using machine-learning algorithms to determine the importance of semantic relations (including contextual factors) in the recommendation and then automatically assign weights to these relations in the contextual graph. In addition, since not all information produces positive effects in context-modeling, feature selection will be used to get rid of weak features.

Disclosure

This work is an extended version of the short paper presented in [13].

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the Special Funds for "Middle-aged and Young Core Instructor Training Program" of Yunnan University, the Applied Basic Research Project of Yunnan Province (2013FB009, 2014FA023), the Program for Innovative Research Team in Yunnan University (XT412011), and the National Natural Science Foundation of China (61472345). The authors are grateful to the anonymous reviewers for their constructive comments and suggestions which contribute substantially to the improvement of this paper.

References

[1] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[2] G. Adomavicius, B. Mobasher, F. Ricci, and A. Tuzhilin, "Context-aware recommender systems," *AI Magazine*, vol. 32, no. 3, pp. 67–80, 2011.

[3] M. Baldauf, S. Dustdar, and F. Rosenberg, "A survey on context-aware systems," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 2, no. 4, pp. 263–277, 2007.

[4] P. G. Campos, I. Fernndez-Tobas, I. Cantador, and F. Dez, "Context-aware movie recommendations: an empirical comparison of pre-filtering, post-filtering and contextual modeling

approaches," in *E-Commerce and Web Technologies: Proceedings of the 14th International Conference (EC-Web '13), Prague, Czech Republic, August 27-28, 2013*, vol. 152 of *Lecture Notes in Business Information Processing*, pp. 137–149, Springer, Berlin, Germany, 2013.

[5] U. Panniello and M. Gorgoglione, "Incorporating context into recommender systems: an empirical comparison of context-based approaches," *Electronic Commerce Research*, vol. 12, no. 1, pp. 1–30, 2012.

[6] U. Panniello, A. Tuzhilin, and M. Gorgoglione, "Comparing context-aware recommender systems in terms of accuracy and diversity," *User Modeling and User-Adapted Interaction*, vol. 24, no. 1-2, pp. 35–65, 2014.

[7] L. Baltrunas and F. Ricci, "Experimental evaluation of context-dependent collaborative filtering using item splitting," *User Modeling and User-Adapted Interaction*, vol. 24, no. 1-2, pp. 7–34, 2014.

[8] V. Codina, F. Ricci, and L. Ceccaroni, "Exploiting the semantic similarity of contextual situations for pre-filtering recommendation," in *Proceedings of the 21th International Conference on User Modeling, Adaptation, and Personalization*, pp. 165–177, Springer, 2013.

[9] Y. Zheng, R. Burke, and B. Mobasher, "Differential context relaxation for context-aware travel recommendation," in *Proceedings of the 13th International Conference on Electronic Commerce and Web Technologies (EC-WEB '12)*, pp. 88–99, 2012.

[10] I. Konstas, V. Stathopoulos, and J. M. Jose, "On social networks and collaborative recommendation," in *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '09)*, pp. 195–202, ACM, July 2009.

[11] H. Wu, X. H. Cui, J. He, B. Li, and Y. J. Pei, "On improving aggregate recommendation diversity and novelty in folksonomy-based social systems," *Personal and Ubiquitous Computing*, vol. 18, no. 8, pp. 1855–1869, 2014.

[12] H. Wu, Y. J. Pei, B. Li, Z. Kang, X. Liu, and H. Li, "Item recommendation in collaborative tagging systems via heuristic data fusion," *Knowledge-Based Systems*, vol. 75, pp. 124–140, 2015.

[13] H. Wu, X. X. Liu, Y. J. Pei, and B. Li, "Enhancing contextaware recommendation via a unified graph model," in *Proceedings of the International Conference on Identification, Information and Knowledge in the Internet of Things*, pp. 1–4, IEEE, 2014.

[14] A. Koir, A. Odi, M. Kunaver, M. Tkali, and J. F. Tasi, "Database for contextual personalization," *Elektrotehniki vestnik*, vol. 78, no. 5, pp. 270–274, 2011.

[15] A. Odić, M. Tkalčič, J. F. Tasič, and A. Košir, "Predicting and detecting the relevant contextual information in a movie-recommender system," *Interacting with Computers*, vol. 25, no. 1, pp. 74–90, 2013.

[16] H. A. Lee, R. Law, and J. Murphy, "Helpful reviewers in TripAdvisor, an online travel community," *Journal of Travel & Tourism Marketing*, vol. 28, no. 7, pp. 675–688, 2011.

[17] Y. Koren, "Factor in the neighbors: Scalable and accurate collaborative filtering," *ACM Transactions on Knowledge Discovery from Data*, vol. 4, no. 1, article 1, 2010.

[18] L. Baltrunas, B. Ludwig, S. Peer, and F. Ricci, "Context relevance assessment and exploitation in mobile recommender systems," *Personal and Ubiquitous Computing*, vol. 16, no. 5, pp. 507–526, 2012.

[19] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver, "Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering," in *Proceedings of the 4th ACM Recommender Systems Conference (RecSys '10)*, pp. 79–86, ACM, September 2010.

[20] B. Zou, C. Li, L. Tan, and H. Chen, "GPUTENSOR: efficient tensor factorization for context-aware recommendations," *Information Sciences*, vol. 299, pp. 159–177, 2015.

[21] H. Wu, F. Luo, X. M. Ning, and H. Jin, "A suggested framework for exploring contextual information to evaluate and recommend services," in *Advances in Grid and Pervasive Computing: Proceedings of the 3rd International Conference, GPC 2008, Kunming, China, May 25–28, 2008*, vol. 5036 of *Lecture Notes in Computer Science*, pp. 471–482, Springer, Berlin, Germany, 2008.

[22] S. Lee, S.-I. Song, M. Kahng, D. Lee, and S.-G. Lee, "Random walk based entity ranking on graph for multi-dimensional recommendation," in *Proceedings of the 5th ACM Conference on Recommender Systems (RecSys '11)*, pp. 93–100, ACM, October 2011.