Ranking in Context-Aware Recommender Systems

Minsuk Kahng, Sangkeun Lee, Sang-goo Lee
School of Computer Science and Engineering
Seoul National University
Seoul, South Korea
{minsuk,liza183,sglee}@europa.snu.ac.kr

ABSTRACT

As context is acknowledged as an important factor that can affect users' preferences, many researchers have worked on improving the quality of recommender systems by utilizing users' context. However, incorporating context into recommender systems is not a simple task in that context can influence users' item preferences in various ways depending on the application. In this paper, we propose a novel method for context-aware recommendation, which incorporates several features into the ranking model. By decomposing a query, we propose several types of ranking features that reflect various contextual effects. In addition, we present a retrieval model for using these features, and adopt a learning to rank framework for combining proposed features. We evaluate our approach on two real-world datasets, and the experimental results show that our approach outperforms several baseline methods.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information filtering, Retrieval models

General Terms

Algorithms, Experimentation

Keywords

context-aware recommender systems, recommender systems, collaborative filtering, learning to rank, ranking in information retrieval, context, usage log

1. INTRODUCTION

The goal of *recommender systems* is to estimate a user's preference and deliver a list of items that might be preferred by the given user. As it is recognized that preferences can be affected by the user's *context*, *context-aware recommender systems* aim to provide recommendation of better quality by utilizing available contextual information (e.g. time, location) of the user [1, 4].

However, incorporating context into recommender systems is not a simple task in that contextual effects on the users' preferences can be diverse depending on the application and domain. For example, some *contextual variables* (e.g. weather) can affect the preferences independent of the given user in some applications. On the other hand, there can be user-dependent contextual effects [5]. Furthermore, a set of contextual variables can be more important than any one individual contextual variable [8]. Other varied forms of contextual effects can be possible in the real world.

To tackle the problem, we propose a novel method for context-aware recommendation, which incorporates several features into the ranking model. We are motivated by the ranking process in search engines, which employs various features [6]. We first formulate the problem of context-aware recommendation as searching the most suitable items with respect to a given user and his context. By decomposing a query, we propose five types of features that reflect various contextual effects. We present a retrieval model for using these features [7], and adopt a ranking framework called *learning to rank* [6, 3] for combining proposed features.

Our approach has two major advantages. First, service providers can improve the quality of recommendation by simply adding appropriate features into the ranking model. With the help of learning to rank, the system can construct an optimal function for the application. In addition, it supports flexibility in that service providers can choose features suitable for their specific purposes.

2. PROBLEM FORMULATION

The user's behavior, such as accessing a webpage and listening to a song, is stored in an item usage log. An *item usage log* has a set of events $\mathcal{E} = \{E_1, \ldots, E_n\}$. An *event* E_i consists of *user* u_i, *context* $\mathbf{c}_i$, and *item* d_i. We define *context* as a composition of *contextual values* v_{it}, where v_{it} is an instance of t-th *contextual variable*. Table 1 shows an example of an item usage log.

We define the problem of *context-aware recommendation* as follows: Given a query q, which consists of a user u and his current context $\mathbf{c}$, the system ranks all candidate items $d \in D$, and recommends the top-k suitable items. In other words, for each candidate item d, $p(d|u, \mathbf{c})$ is calculated, and then the system produces a list of k items whose probabilities are the highest.

Table 1: Example of Item Usage Log

User	Context			Item
Username	Time of Day	Location	Weather	Song ID
bluesky	morning	office	sunny	340
bobsmith	evening	coffee shop	rainy	370
musiclover	evening	home	cloudy	370

WWW 2011, March 28–April 1, 2011, Hyderabad, India.
ACM 978-1-4503-0637-9/11/03.

3. RANKING ITEMS USING CONTEXT

3.1 Ranking Features

We propose five types of features that reflect various contextual effects. To obtain them, we decompose a query that consists of a user u and a set of contextual values v.

The first type of feature does not consider any component of the query; items are recommended based on their global popularity. The probability of an item d given a user u and context $\mathbf{c}$ can be written as:

$$p(d|u,\mathbf{c}) \propto p(d).$$

The second type of feature, similar to the traditional *collaborative filtering* [2], considers the *user* for recommending items. The equation becomes:

$$p(d|u,\mathbf{c}) \propto p(d|u).$$

The third type of feature considers one individual *contextual variable*. Preferences are dependent on individual contextual values regardless of the user as shown below. For example, some song items are preferred more on rainy days.

$$p(d|u,\mathbf{c}) \propto p(d|v).$$

The fourth type of feature pairs up two of the components: the user and any one contextual variable. It models user-dependent contextual effects [1, 5]. For instance, some users like certain songs in the morning, while others do not.

$$p(d|u,\mathbf{c}) \propto p(d|u,v).$$

The last type of feature considers all components of the query, which can be thought of as abstracted context [8].

$$p(d|u,\mathbf{c}) \propto p(d|u,v_1,\ldots,v_T),$$

where T is the number of contextual values.

3.2 Retrieval Model

We adopt the existing LDA-based document retrieval model proposed by Wei and Croft [7]. Based on the *query likelihood model* they used, we assume $p(d|u,\mathbf{c}) \propto p(u,\mathbf{c}|d)$. For the fourth type of feature, the model can be written as:

$$p(u,v|d) = \lambda\Big(\frac{N_d}{N_d+\mu}p_{ML}(u,v|d) + (1-\frac{N_d}{N_d+\mu})p_{ML}(u,v|C)\Big) + (1-\lambda)p_{LDA}(u,v|d),$$

where p_{ML} is for the maximum likelihood estimation, 'C' represents the whole item collection, and p_{LDA} is for the LDA. The LDA topic model finds similar users or context by extracting the latent topics z from the given data:

$$p_{LDA}(u,v|d) = p(u,v|\theta_d,\beta) = \sum_k p(u,v|z_k,\beta)p(z_k|\theta_d).$$

3.3 Learning to Rank Items

Once we have several features and the retrieval model, we need a ranking function to produce a ranked list of items. Using a machine learned ranking framework called *learning to rank* [6], we obtain a ranked result by combining the features. We choose to use *Ranking SVM*, one of the well known learning to rank algorithms. To apply this pairwise approach, we randomly generate some negative feedback since we have only positive feedback, and we assume that the item selected by the user is more preferred than any other item.

4. EXPERIMENTS

We evaluate our method on two real-world datasets. First, we use a music listening log gathered from the music streaming service called Bugs (`http://www.bugs.co.kr`). We incorporate all five types of features into the model using user and three contextual variables: date, time of day, and weather. Second, we collect a place 'check-in' log from the location-based service named Foursquare (`http://foursquare.com`). We choose user, GPS location, date, and time of day. The system suggests the top-k places for a query.

We use *normalized discounted cumulative gain* (NDCG) as an evaluation measure. We consider each event as a different query, and check if the model produces a list that includes the item in the event [5]. We compare our method to three baseline methods: popularity (type 1), collaborative filtering using LDA (type 2), and reduction-based approach [1].

Table 2 shows a comparison of our method to the baseline methods on two datasets. Our method which uses *Ranking SVM* outperforms all baseline methods. This is because our approach, compared to other methods, can incorporate various types of features into the unified ranking model.

Table 2: NDCG@k comparison on two datasets

Method	Music		Foursquare	
	@5	@10	@5	@10
Popularity	0.0670	0.0902	0.0199	0.0246
User only (CF)	0.0687	0.0946	0.0822	0.0996
Reduction	0.0909	0.1030	0.2368	0.2408
Ranking SVM	0.2161	0.2359	0.4612	0.4890

5. CONCLUSION

In this paper, we propose a novel method for context-aware recommendation by incorporating several features into the ranking model. We propose five types of features that model various contextual effects. We present a retrieval model for using these features, and utilize the learning to rank framework for combining the features. The experimental results show that our approach performs better than some baseline methods.

6. REFERENCES

[1] G. Adomavicius and A. Tuzhilin. Context-aware recommender systems. In *Recommender Systems Handbook*. Springer, 2010.

[2] A. S. Das, M. Datar, A. Garg, and S. Rajaram. Google news personalization: scalable online collaborative filtering. In *WWW '07*. ACM, 2007.

[3] F. Diaz, D. Metzler, and S. Amer-Yahia. Relevance and ranking in online dating systems. In *SIGIR '10*. ACM, 2010.

[4] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *RecSys '10*. ACM, 2010.

[5] D. Lee, S. E. Park, M. Kahng, S. Lee, and S.-g. Lee. Exploiting contextual information from event logs for personalized recommendation. In *Computer and Information Science 2010*. Springer, 2010.

[6] T.-Y. Liu. Learning to rank for information retrieval. *Found. Trends Inf. Retr.*, 3:225–331, 2009.

[7] X. Wei and W. B. Croft. Lda-based document models for ad-hoc retrieval. In *SIGIR '06*. ACM, 2006.

[8] E. Zheleva, J. Guiver, E. Mendes Rodrigues, and N. Milić-Frayling. Statistical models of music-listening sessions in social media. In *WWW '10*. ACM, 2010.