PAPER

Personalized Recommendation of Item Category Using Ranking on Time-Aware Graphs

Chen CHEN†, Member, Chunyan HOU††(a), Peng NIE†, and Xiaojie YUAN††, Nonmembers

SUMMARY Recommendation systems have been widely used in E-commerce sites, social media and etc. An important recommendation task is to predict items that a user will perform actions on with users’ historical data, which is called top-K recommendation. Recently, there is huge amount of emerging items which are divided into a variety of categories and researchers have argued or suggested that top-K recommendation of item category could be very beneficial for users to make better and faster decisions. However, the traditional methods encounter some common but crucial problems in this scenario because additional information, such as time, is ignored. The ranking algorithm on graphs and the increasingly growing amount of online user behaviors shed some light on these problems. We propose a construction method of time-aware graphs to use ranking algorithm for personalized recommendation of item category. Experimental results on real-world datasets demonstrate the advantages of our proposed method over competitive baseline algorithms.

key words: recommendation system, personalization, time-aware graphs

1. Introduction

With the huge volume of information available on the Web, it is necessary to help users filter overloaded data and recommend relevant information. As a result, recommendation systems have been widely used in E-commerce, social media and etc. An important recommendation task is to predict items that a user will perform a certain action on with user’s historical events, which is also called top-K recommendation in some literatures [2]. There are so many emerging items in E-commerce which are divided into a variety of categories. Researchers have argued or suggested that the recommendation of item category could be very beneficial. For example, Taobao Mall is a platform of Business-to-Consumer (B2C) online retail to sell a variety of brands. Based on the dataset of Taobao Mall, AliData Discovery Challenge in 2014 is associated with the brand recommendation. The brand can be regarded as a kind of category and the brand recommendation is a kind of recommendation of item category. The recommendation of item category begins to gain much attention and becomes hot research topic because item recommendation encounters some problems in practical applications. First, it is difficult to make the item recommendation in terms of the vague needs of users. Usually, users do not know exactly which item to buy, but consider which brand or style is necessary and make decisions after comparing many items of this kind. Second, there is huge amount of emerging new items every day and it is difficult to recommend these items because they have very few feedbacks. Users can find these items by their category and choose to buy or not, which can increase the ease of the user enjoyment. The recommendation of item category becomes an alternative. In addition, user behaviors, such as clicking and purchasing an item of the category, are often recorded by the system. By recommending categories, the system has the potential to allow users to choose items gradually, help users make better and faster decisions and improve user experiences. Thus, we focus on personalized top-K recommendation of item category.

One of the most popular recommendation frameworks is Collaborative Filtering (CF) which has achieved great success in Personalized Recommendation System[18]. Generally, CF methods can be divided into neighbor-based approach[15], [17] and model-based approach[7], [8], [11]. The neighbor-based approach, such as user-based method[15] and item-based method[17], is now widely used in practice. In the last few years, model-based approach, especially the latent factor model based on matrix factorization, has attracted a lot of attention in research community and industry due to their good performance. Many matrix factorization methods have been proposed. However, recommendation systems based on these methods have following problems in the personalized top-K recommendation of item category. First, it is difficult for these methods to use duplicated events. A user performs actions on a lot of items which belong to a category. If these historical events are represented on the level of category, they are duplicated, that is, a user performs actions on a category many times. Second, it is difficult to use temporal information. Recommendation system should not be static and should model dynamic changes of user’s interests. Lack of dynamics impacts the ability to help uses make better decisions.

The ranking algorithm on graphs and the increasingly growing amount of online user behaviors shed some light on these problems. We propose a construction method of time-aware graphs to use ranking algorithm for personalized recommendation of item category. First, we use graphs
to model dynamic changes of users behaviors. The historical events are used to generate the individual preference. Then, the public trend and private interest are integrated into a unified model, which are used to generate personalized recommendation of item category.

In the rest of this paper, we first review some related works in Sect. 2 and provide detailed description of our approach from the graph construction to the ranking algorithm in Sect. 3. Then, we introduce the experimental setup and results on real-world datasets to validate the performance of our proposed approach in Sect. 4. Finally, we conclude the study and discuss future research directions in Sect. 5.

2. Related Work

In this section, we will review some existing works on personalized recommendation. One of the most popular recommendation frameworks is Collaborative Filtering (CF). Generally, CF can be divided into neighbor-based and model-based. On one hand, the neighbor-based model is the classical models in CF. Two early representatives are user-based and item-based models, which calculate the similarity between users or items. On the other hand, model-based methods have achieved great success in Personalized Recommendation System [10]. Latent Factor Models (LFM) based on Matrix Factorization (MF) have gained great popularity because they usually have achieved state-of-the-art performance in some benchmark datasets [18]. A variety of MF algorithms have been proposed, such as Singular Value Decomposition (SVD) [7], Non-negative Matrix Factorization (NMF) [8], Max-Margin Matrix Factorization (MMMF) [14], Probabilistic Matrix Factorization (PMF) [11], and Local Matrix Factorization (LMF) [9]. They aim at learning latent factors from user-item matrices to make recommendation. However, their latent characteristic makes it difficult to make accurate recommendation on the level of category because some important information, such as time, is ignored.

The most related work to our study is the random walk method for recommendation. There have many models and algorithms based on random walk, such as PageRank [12], HITS [6], LexRank [13], Personalized PageRank [4] and etc. Generally, there are two research lines. Some studies employ the random walk on a variety of graphs, such as the trust network [5], item similarity graph [19], contextual graph [1] to refine the prediction performance. Another way is to use the random walk to measure similarities between users or items, which can overcome the sparsity problem compared with measures in neighbor-based models [3], [16]. Our study goes along the former line. The major difference between our method and these methods is that we focus on time-aware graph to model dynamic changes of user behaviors and use ranking algorithms on the graphs. To the best of our knowledge, our approach is the first attempt to use ranking algorithms on graphs for top-K recommendation of item category.

3. Personalized Recommendation

3.1 Problem Definition

Rating prediction has been popular in the research and industry community. Unfortunately, most real-world recommendation systems do not depend on ratings since users are hard to persuade to give explicit feedbacks while other kinds of feedback (i.e., implicit feedback) are easily obtained by the system, such as user actions like selecting or buying an item, visiting or clicking a page. Buying a product or watching a movie is also a positive expression of preferences. However, not buying or watching an item does not necessarily mean a user dislikes the item. Top-K item prediction from the implicit feedback is the task of determining K items that a user will perform a certain action on using past events or other data. Top-K recommendation of item category is similar to this task, but in terms of categories rather than items. We now formally define the problem of top-K recommendation of item category. Let U, I and C be the set of users, items, categories respectively, and T be a time interval. Each item belongs to at least a category. \( a(u,i,t) \) denotes an action performed by user \( u \) on an item \( i \) at time \( t \). We define category set \( C \) as:

\[
C = \{c_1,c_2,\ldots,c_{|C|}\}
\]  

(1)

where each \( c_i \) denotes a category and \( |C| \) indicates the total number of categories. Our entire training data can be written as:

\[
\Omega(T) = \{(u,c,t) | u \in U, c \in C, t \in T, \exists i \in c, a(u,i,t) \text{is known}\}
\]  

(2)

We also define notation \( C(u,T) \) in Eq. (3) to denote the set of all categories on which user \( u \) performs actions in \( T \).

\[
C(u,T) = \{c | u \in U, t \in T, \exists i \in c, a(u,i,t) \text{is known}\}
\]  

(3)

**Definition 1 (Top-K Recommendation of Item Category):**

Given \( \Omega(T1) \), top-K recommendation of item category is to find a ranking of categories in \( C(u,T2) \) \((T2 > T1)\) for user \( u \) such that the ranking has the good performance of recommendation.

3.2 Time-Aware Graph

Let \( G=(V,E) \) be a graph where \( V \) is a finite set of vertices and \( E \) is a finite set of edges. We define an ordered pair \((u,v)\) as an edge from vertex \( u \) to vertex \( v \). When \( G \) is an undirected graph, we have \((u,v)=(v,u)\). When \( G \) is a directed graph, we have \((u,v)\neq(v,u)\). In time-aware graphs, \( V \) refers to a set of categories and \( E \) refers to the relationships between categories. We define the weight of an edge with \( W(u,v) \). In general, \( W(u,v) \) could take any non-negative real values. Then, we cast the task of recommendation of
We give some important definitions and then describe the construction of time-aware graph.

**Definition 2 (Activate):** If an item is performed action on by user \( u \) for a period of time, we call that its category is activated by user \( u \) during that time.

**Definition 3 (Category Set of User for \( k \)-th Time):** Given the time unit \( t \), the set of categories which are activated by user \( u \) in \( t \), is numbered sequentially from the first. The \( k \)-th category set of user \( u \) is defined as the category set of user \( u \) for \( k \)-th time, which is represented as \( C(u,k) \).

**Definition 4 (Time-aware Graph):** Let \( G = (V,E) \) be a directed graph where \( V \) is a set of vertices and \( E \) is a set of edges. \( G \) is called a time-aware graph if \( V(p,q) \in E \), the weight of this edge is

\[
W(p,q) = \sum_{u \in U} \sum_{v \in K(u,p,q)} [T(u,q,k+1) - T(u,p,k)]^\beta \quad (4)
\]

\[
K(u,p,q) = \{|k|p \in C(u,k) \land q \in C(u,k+1)\} \quad (5)
\]

where \( U \) is the set of users, \( T(u,p,k) \) denotes the date when user \( u \) activates category \( p \) for the \( k \)-th time, \( \beta \) is a parameter called the delay factor here. Note that the \( k \)-th and \((k+1)\)-th time are consecutive for user \( u \), but the interval between them maybe several time units.

With respect to Definition 3, when one category is activated frequently in one time unit, redundant categories are removed and the unique category is remained in a category set. An example is shown in Fig. 1. With day as the time unit, user \( u \) activates the category \( C1, C2 \) and \( C3 \) for the \( k \)-th time, and category \( C2, C4, C5 \) and \( C6 \) for the \((k+1)\)-th time. Thus, \( C(u,k) = C1, C2, C3 \) and \( C(u,k+1) = C2, C4, C5, C6 \). There are 12 directed edges. There are 5 days between the \( k \)-th and \((k+1)\)-th time, and \( T(u,C2,k+1) - T(u,C1,k) = 5 \).

If we set \( \beta = -1 \), each of 12 directed edge has the weight of \( 5^{-1} = 0.2 \), that is, \( W(C1,C2) = 0.2 \). User behavior is the indication of user interest. With Definition 4, an edge with a larger weight implies that an interest transition is more likely to happen from a category to another for a user. Time-aware graphs model the dynamics of users’ interests in terms of categories. We can model the dynamics of all users’ interests with time-aware graphs by summing the weight of edges up. Intuitively, the longer interval between the \( k \)-th and \((k+1)\)-th time is, the fewer possibility the transition has.

### 3.3 Ranking Algorithms

A family of ranking techniques uses the stationary distribution of a random walk on graphs. A random walk defines a Markov chain in a given directed or undirected graphs, where each vertex represents a state and a walk transits from a state to another based on a transition probability. In other words, a random walk on graphs is defined by a transition probability function \( f : V \times V \rightarrow [0,1] \). Let \( p_T(u) \) be the probability that the walk is at state \( u \) at time \( T \). A standard random walk can be defined as

\[
p_T(u) = \sum_{u(v,u) \in E} f(u,v)p_{T-1}(u) \quad (6)
\]

If the Markov chain is irreducible and aperiodic, all states are positive recurrent and thus ergodic, and \( p_T(u) \) converges to a stationary distribution \( \pi(v) \) which is usually used to measure the importance of vertices.

Most existing random walk models assume that the transition probability \( f(u,v) \) can be estimated using the topological structure of the graph or the prior knowledge about the process. In the time-aware graph, \( f(u,v) \) is estimated by

\[
f(u,v) = \begin{cases} 
(1 - \lambda) \times p(v) + \lambda \times \frac{\sum_{c \in C(u)} W(u,c)}{\sum_{v \in V} W(u,v)}, & \text{if } W(u,v) > 0 \\
\lambda, & \text{otherwise}
\end{cases} \quad (7)
\]

where \( \lambda \) is the damping factor and \( W(u,v) \) is the weight of edge \((u,v)\). The Markov chain defined by Eq. (7) is ergodic. The stationary distribution of this random walk is the well-known PageRank score.

There are two methods to estimate the \( f(u,v) \) in different scenarios. If none of prior distribution \( p(v) \) is known, we assume \( p(v) \) is uniformly distributed and substitute \( p(v) \) with \( 1/|C| \) in Eq. (7). Otherwise, we use the laplace smoothing to estimate \( p(v) \) by Eq. (8). The stationary distribution of such a random walk refers to personalized PageRank, or topic-sensitive PageRank.

\[
p(v) = \frac{\#(v \text{ is activated}) + 1}{\#(c \text{ is activated}) + 1} \quad (8)
\]

We apply ranking algorithms on the time-aware graph to recommend top-\( K \) categories for each user. Intuitively, a user has different preferences to each of categories and personal preference can be inferred from his historical data. A category may include many items which are associated with user’s behaviors. For the recommendation of item category,
a prior distribution can be estimated by counting how often a category is activated. The prior distribution \( p(y) \) indicates the personal interest. User's interest changes over time and is affected by the public trend. The time-aware graph can model the dynamic change of user’s behaviors. The edge represents the transition of users behaviors over time. The weight of edge reflects the probability of transition.

4. Experiments

4.1 Datasets

We choose the dataset of TMall† (the abbreviation of Taobao Mall) and Yelp†† for experimentations. Some statistics of the dataset is shown in Table 1. Taobao Mall is a platform of business-to-consumer (B2C) online retail to sell a variety of brands for local Chinese and international businesses. The dataset consists of user behaviors on brands for four months, that is, April, May, June and July. A user can click items, add items to the wish list, put items in the shopping basket or purchase items of a range of brands. The brand is regarded as the category. As shown in Fig. 2 (a), the number of behaviors is usually stable for four months. The Yelp dataset consists of user reviews on the businesses located in the Phoenix City of the US. Every business has the attribute of category which means the class of business such as Food, Restaurant and etc. The number of reviews is sufficient and stable from Jan, 2011 to Dec, 2012. Thus, Yelp dataset consists of 24 month data which is used in experiments. The number of behaviors is shown in Fig. 2 (b).

Two datasets have different taxonomy. In the dataset of TMall, an item just has one category because a product has just one brand and a category consists of many items. In Yelp, an item may have many categories, that is, the number of category is more than one for some items, and a category also has many items. Experiment results can verify the generalization of our proposed method with different kinds of datasets.

4.2 Experimental Setup

Based on the dataset of user behaviors in the past, our goal is to build category preference for each user and recommend categories that a user will most likely to perform a certain action on in the future. Performing a behavior on a product of a category is regarded as a positive expression of preference while no behavior means a negative expression. Thus, top-\( K \) recommendation is based on positive-only implicit feedbacks. For TMall dataset, any behavior is regarded as the positive feedback while the review with 4 or 5 stars as positive feedback for Yelp dataset. In that case, two datasets can be used to evaluate the performance of top-\( K \) recommendation.

We evaluate the performance of methods by separating the data into the train data that is used to train the model, and the test data that is used to evaluate it. The data of the previous month is kept for training while the data of the next month is for testing. All parameters of ranking algorithms for next month are estimated by data of the previous month.

In order to compare the results with previous studies, we use Precision, Recall, F1-Score, Normalized Discounted Cumulative Gain (NDCG) and Mean Average Precision (MAP) to evaluate the performance of different models. Note that F1 score is computed by Precision@10(P@10) and Recall@10(R@10).

We call our method as PPR for abbreviation because it is based on Personalized PageRank. We compare our proposed method with the following baseline algorithms:

**Most Popular (MP)**: Non-personalized recommendation where the items are ranked by the number of behaviors performed by users.

**KNN**: neighbor-based collaborative filter algorithms which are divided into user-based and item-based KNN. They are represented by UserKNN and ItemKNN respectively.

**NMF**: Nonnegative Matrix Factorization algorithm

---

†http://102.alibaba.com/competition/addDiscovery/index.htm
††http://www.yelp.com/dataset_challenge
that is among the best latent factor models in the previous studies.

**BPRMF**: Bayesian Regularized Ranking (BRP) optimization for Matrix Factorization, which is the state-of-the-art algorithm for top-K recommendation.

**History**: we assume that users behavior is stable in a period of time. In other word, a user tends to perform behaviors this month which are similar to the previous month. We predict the ranking of categories based on this assumption.

### 4.3 Results

#### 4.3.1 Results of TMall Dataset

There is TMall dataset for four months, so we report the results in Table 2. The results are on top of Table 2 when April for training and May for testing, and so on.

The traditional item-based method is worst because the number of users is far fewer than the number of categories for TMall dataset. MP, UserKNN, NMF and BPRMF have the similar performance. As a heuristic method, MP is able to get the comparable result. This shows that the public trend is an important factor to determine users behaviors. The history method outperforms these typical methods of collaborative filtering. This observation confirms the hypothesis that the user behaviors are stable. Private interests can provide good recommendation regardless of the public trend. PPR on time-aware graph outperforms other comparative methods consistently. Therefore, incorporating private interests and public trend can improve the performance.

#### 4.3.2 Results of Yelp Datasets

For Yelp dataset, the average of all 24 months’ scores is reported in Table 3. ItemKNN and NMF have the trivial performance. The performance of MP, UserKNN, BPRMF are similar. The history method is better in NDCG, but worse in P@10 than MP and BPRMF. Because NDCG evaluates the ranking rather than the set, history method improve the ranking more than the set. PPR on time-aware graph is best among all methods. With respect to the F1 and NDCG, we depict the result of every month in Fig. 3. The detailed results are consistent with that in Table 2.

#### 4.3.3 Sensitivity of Damping Factor

We study how the performance changes as the damping factor increase in the range of [0, 1] with a step size of 0.1. The average of all months is computed and used in Fig. 4. The larger damping factor is, the less important private behaviors in past are. The time-aware graph is used to model the dynamic changes of public trend in behaviors. Thus, the damping factor can be seen as a tradeoff of the private interest and public trend.

For Taobao Mall dataset, we find that the optimal value for the damping factor is 0.1 in terms of NDCG. Results on other metrics are similar. Damping factor 0.1 means that if a user has a 10% likelihood of choosing a random walk from the state they are currently visiting, and a 90% likelihood of jumping to a state chosen at random from the all states in graphs by historical data. In other word, users behaviors are more determined by the private interest, and less by the public trend.

For Yelp dataset, the damping factor plays a role in different way. As shown in Fig. 4 (b), the performance continues to rise with the increase of damping factor from zero, and tends to be stable when the damping factor equals 0.2. Thus, users behaviors are more determined by the public trend. An item (i.e. a business in Yelp) can belong to many categories. If there is an action performed on this item, its categories are activated and become positive feedbacks. There is an action performed on this item, its categories are activated and become positive feedbacks.

#### 4.3.4 Time-Aware Graphs

When the ranking on graphs is used for personalized recommendation, it is important to study how the performance is affected by time-aware graphs. We investigate whether time-aware graphs built on more data can enhance the recommendation. We set the damping factor to 0.1 in the experiments on TMall dataset while the damping factor to 0.2 on Yelp dataset. We also study how the delay factor has effect on the performance and find that the delay factor plays a trivial role. Thus, we let the delay factor equal to zero in the following experiments.
We build time-aware graphs on the data of more than two months. As shown in Table 4, the first column means which months are kept for training and which month is for testing. For example, 4,5-6 indicates that the data in April and May is kept for training while the data in June is for testing. We observe that the performance becomes bad when more data is used to build graphs. Like to the experiments on TMall dataset, we choose Yelp dataset in Dec. 2012 for testing. In the first column of Table 5, P1 refers to the data of previous one month (i.e., Nov. 2012) which is used for training. Similarly, P2 means the data of previous two months (i.e., Oct. and Nov. 2012).

Experiment results on both datasets means that the noise is introduced into the recommendation procedure when more data is used to build time-aware graphs. The possible reason is that although more data is added, these data is not fresh and graphs cannot exhibit the latest public trend. The observation is consistent with that of TMall dataset. In addition, although the recommendation based on the category alleviates the sparsity problem to some extent, the absolutely few historical data leads to bad recommendation. The sparsity problem accounts for most of failures.

5. Conclusions and Future Work

In this paper, we formulate a new problem of top-$K$ recommendation of item category, and propose to construct time-aware graphs to rank categories for each user. We build time-aware graphs to model the dynamic change of public trend and use the individual history to model user’s private interests, then incorporate both the public trend and the private interest into a unified ranking framework on graphs. The experimental results show our method significantly outperforms baseline methods in top-$K$ recommendation of item category. In addition, we study how the performance changes as the damping factor varies and find that there are different effects on TMall and Yelp datasets. The time-aware graphs built on more data cannot enhance the recommendation.

In the future, we will compare more ranking algorithms on graphs. Even though our proposed method works well in experiments, it is interesting to investigate way of the random walk to measure similarities between users or items.
Acknowledgments

Research was partially supported by National Natural Science Foundation of China (No.61170184, 61402242, 61402243 and 61402333), the Tianjin Municipal Science and Technology Commission (No.13ZCZDGX02200, 13ZCZDGX01098), the Research Foundation of Ministry of Education and China Mobile (No.MCM20130381).

References

[1] T. Bogers, “Movie recommendation using random walks over the contextual graph,” Proc. 2nd Workshop on Context-Aware Recommender Systems, 2010.

[2] P. Cremonesi, Y. Koren, and R. Turrin, “Performance of recommender algorithms on top-n recommendation tasks,” Proc. Fourth ACM Conference on Recommender Systems, pp.39–46, 2010.

[3] F. Fouss, A. Pirotte, J. Renders, and M. Saerens, “Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation,” IEEE Trans. Knowl. Data Eng., vol.19, no.3, pp.355–369, 2007.

[4] T.H. Haveliwala, “Topic-sensitive PageRank,” Proc. 11th International Conference on World Wide Web, pp.517–526, 2002.

[5] M. Jamali and M. Ester, “TrustWalker: A random walk model for combining trust-based and item-based recommendation,” Proc. 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.397–406, 2009.

[6] J.M. Kleinberg, “Authoritative sources in a hyperlinked environment,” J. ACM, vol.46, no.5, pp.604–632, 1999.

[7] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” Computer, vol.42, no.8, pp.30–37, 2009.

[8] D.D. Lee and H.S. Seung, “Algorithms for non-negative matrix factorization,” Proc. Advances in Neural Information Processing Systems, pp.556–562, 2001.

[9] J. Lee, S. Bengio, S. Kim, G. Lebanon, and Y. Singer, “Local collaborative ranking,” Proc. 23rd International Conference on World Wide Web, pp.85–96, 2014.

[10] J. Liu, M.Z.Q. Chen, J. Chen, F. Deng, H. Zhang, Z. Zhang, and T. Zhou, “Recent advances in personal recommender systems,” Int. J. Information and Systems Sciences, vol.5, no.2, pp.230–247, 2009.

[11] A. Mnih and R. Salakhutdinov, “Probabilistic matrix factorization,” Proc. Advances in Neural Information Processing Systems, pp.1257–1264, 2007.

[12] L. Page, S. Brin, R. Motwani, and T. Winograd, “The PageRank citation ranking: Bringing order to the Web,” Technical Report, Stanford Digital Library Technologies, 1998.

[13] D.R. Radev and G. Erkan, “Lexrank: Graph-based centrality as salience in text summarization,” J. Artif. Intell. Res., vol.22, 2004.

[14] J.D.M. Rennie and N. Srebro, “Fast maximum margin matrix factorization for collaborative prediction,” Proc. 22nd International Conference on Machine Learning, pp.713–719, 2005.

[15] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, “GroupLens: An open architecture for collaborative filtering of netnews,” Proc. 1994 ACM Conference on Computer Supported Cooperative Work, pp.175–186, 1994.

[16] Y. Rong, X. Wen, and H. Cheng, “A Monte Carlo algorithm for cold start recommendation,” Proc. 23rd Conf. World Wide Web, pp.327–336, 2014.

[17] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms,” Proc. 10th International Conference on World Wide Web, pp.285–295, 2001.

[18] X. Su and T.M. Khoshgoftaar, “A survey of collaborative filtering techniques,” Advances in Artificial Intelligence, vol.2009, p.19, 2009.

[19] H. Yildirim and M.S. Krishnamoorthy, “A random walk method for alleviating the sparsity problem in collaborative filtering,” Proc. 2008 ACM Conference on Recommender Systems, pp.131–138, 2008.