Top-N Recommendation through Belief Propagation

Jiwoon Ha
Dept. of Electronics and Computer Engineering
Hanyang University
Seoul, 133-791, Korea
jiwoonha@hanyang.ac.kr

Soon-Hyoung Kwon
Dept. of Electronics and Computer Engineering
Hanyang University
Seoul, 133-791, Korea
rikar@agape.hanyang.ac.kr

Sang-Wook Kim
Dept. of Electronics and Computer Engineering
Hanyang University
Seoul, 133-791, Korea
wook@hanyang.ac.kr

Christos Faloutsos
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213, USA
christos@cs.cmu.edu

Sunju Park
School of Business
Yonsei University
Seoul, 120-749, Korea
boxenju@yonsei.ac.kr

ABSTRACT

The *top-n recommendation* focuses on finding the top-n items that the target user is likely to purchase rather than predicting his/her ratings on individual items. In this paper, we propose a novel method that provides top-n recommendation by probabilistically determining the target user's preference on items. This method models the purchasing relationships between users and items as a bipartite graph and employs *Belief Propagation* to compute the preference of the target user on items. We analyze the proposed method in detail by examining the changes in recommendation accuracy under different parameter settings. We also show that the proposed method is up to *40%* more accurate than an existing method by comparing it with an *RWR*-based method via extensive experiments.

Categories and Subject Descriptors

H.2.8 [**DATABASE MANAGEMENT**]: Database applications—*Data mining*

General Terms

Algorithms, Human Factors

Keywords

Top-n recommendation, Belief propagation, Data mining

1. INTRODUCTION

As a result of growing popularity in e-commerce, recommender systems have gained its importance as an active research area and as a practical business tool. There exists an extensive amount of prior work on recommender systems [1]. Recommender systems are employed in many online shopping malls, such as Amazon.com, EBay.com, etc., and profits are generated through these recommender systems [3].

Existing recommender systems targeted for the Netflix prize and others have focused on rating prediction [1, 7, 12]. When we recommend items to customers, however, we should rank the items and pick a specific number of items from it. In fact, online shopping malls provide a specific number of items or the ranked list of items. In order to recommend a specific number of items, therefore, rankings of items are more important than predictions of a rating of each item. These systems that recommend a specific number of items are called *top-n recommendation systems*.

The ranking on items is possible based on ratings of individual items computed through existing rating prediction methods. When they are applied to top-n recommendation, however, the rating prediction methods do not show sufficient accuracy, because they do not focus on the prediction of the *relative* preferences of items, but focus on the absolute preferences (rating) of items [5].

There is a top-n recommendation method based on *random walk* or *random walk with restart (RWR)* [4, 6, 9]. As a method for calculating the relevance between two nodes in a graph, the *RWR* is widely used in various domains [4, 6, 9]. In recommender systems, the relevance between the target user node and item nodes could be regarded as the target user's preferences on each item. The existing *RWR*-based methods have outperformed the rating prediction methods in the top-n recommendation [4, 9].

In this paper, we propose a novel method that provides top-n recommendation by probabilistically determining the target user's preference on items based on a graph. When the relationships between users and items are modeled as a graph, we can infer effectively the implicit preference between users and items through transitivity [4, 6, 9]. Th proposed method infers a target user's preference on each item through *Belief Propagation (BP)* on a graph.

By employing the *BP* to a bipartite graph that represents the relationships between users and items, we can calculate the probabilities that each item is preferred by each user and provide a target user with the top-n recommendation that recommends items in the order of his/her preference

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.

probabilities. In order to apply the *BP* to top-n recommendation, graph generating criteria and initial values of the nodes should be finely tuned. In this paper, we discuss how to generate the edges of the graph and determine the node potentials based on the ratings.

We perform top-n recommendation on the *MovieLens* data and evaluate the accuracy of the proposed method. The accuracy is measured by the *precision*, *recall*, and *mean average precision (MAP)*. The results show that the proposed method outperforms the representative *item-based* method [12] and the *RWR*-based method [4, 6, 9].

2. PROPOSED METHOD

We propose a novel top-n recommendation method that probabilistically determines the target user's preference on items based on a bipartite graph that represents the relationships between users and items. To apply the *BP* to the top-n recommendation, the graph and its parameters should be finely tuned. In this section, we describe the *BP* in general and discuss the way to generate the graph and to define its parameters.

The proposed method runs as follows: 1) modeling the users, the items, and the relationships between them as a bipartite graph with the edge threshold, 2) assigning node potentials based on the target user's ratings on items, 3) defining the propagation matrix, 4) passing messages between nodes on the bipartite graph iteratively for a specific number of times or until the change of the message values between iterations is less than some predefined threshold, 5) calculating the belief of each node (i.e., how much the target user is likely to prefer each item), 6) ranking the items based in order of the belief and determining the list of top-n items. When we want to recommend items to many target users, we repeat the above procedure for every target user.

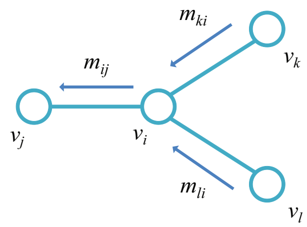
2.1 Belief propagation

To infer the state of a node based on the status of its neighboring nodes, the *BP* is widely used [2, 10, 11]. The information is exchanged between nodes through message passing to infer the state of nodes. The messages sent from one node to another are a vector, called message vector that has possible states as its elements. Each message vector is obtained as Equation 1.

$$m_{ij}(x_c) \leftarrow \sum_{x_d \in X} \phi_i(x_d)\psi_{ij}(x_d, x_c) \prod_{k \in N(i) \setminus j} m_{ki}(x_d) \quad (1)$$

$m_{ij}(x_c)$ denotes the message sent from v_i to v_j. It means v_i's opinion (belief) about v_j's likelihood of being in class x_c. The message from v_i to v_j is made up with the product of the messages from v_i's neighboring nodes except v_j $(N(i) \setminus j)$. $\psi_{ij}(x_d, x_c)$ is the edge potential from the propagation matrix that represents the probability of the v_j being in state x_c when its neighboring node v_i is in state x_d. $\phi_i(x_d)$ is a node potential that represents the probability of v_i being in state x_d. If the state of a node is already known, a higher potential value is assigned to the corresponding state. If the state of the node is unknown, the same potential values are assigned to all of the states.

Figure 1 shows the computation of node v_i's belief about the likelihood of node v_j is in the x_c state, assuming there are two possible node states, x_c and x_d.



$$m_{ij}(x_c) \leftarrow \left\{ \begin{array}{lc} & m_{ki}(x_c) \\ \times & m_{li}(x_c) \\ \times & \psi_{ij}(x_c, x_c) \\ \times & \phi_i(x_c) \end{array} \right\} + \left\{ \begin{array}{lc} & m_{ki}(x_d) \\ \times & m_{li}(x_d) \\ \times & \psi_{ij}(x_d, x_c) \\ \times & \phi_i(x_d) \end{array} \right\}$$

Figure 1: The computation of node v_i's belief about the likelihood of node v_j is in the x_c state.

When the message passing is done on every state, we normalize the sum of every state of each message vector to *1* in order to avoid numerical underflow. The message passing can be performed iteratively on every edge either for a specific number of times or until the change of the message values between iterations is less than some predefined threshold.

After the message passing, belief scores that the v_i's probability of being in class x_c are determined by Equation 2. The belief score vectors have the possible states as the elements, similar to the message vector.

$$b_i(x_c) = k \prod_{j \in N(i)} m_{ji}(x_c) \quad (2)$$

k is the normalizing factor for making the sum of the belief scores of a node to be *1*. The belief scores represent the possibility of the node being in the corresponding state.

When we apply the *BP* to the recommender system, the message from v_i to v_j can be regarded as the v_i's belief that the likelihood of v_j is liked by the target user, and we can infer the likelihood that the target user prefers each item from the messages each node has received.

2.2 Graph

We create a bipartite graph from the users' preference on items. We model both the users and the items as nodes and the users' ratings on items as edges. The states of a node can be either *like* or *dislike* states[1]. In order to reflect the users' preference more accurately, we assume the user prefers the item only when the rating is higher than a threshold. That is, we create an edge only when the rating is higher than the threshold (instead of creating an edge whenever the user rates the item). How to determine the threshold is investigated in the following section. Figure 2 depicts the graph generation process.

2.3 Node potential

For the personalized recommendation, we assign the node potentials to the item nodes based on the target users' ratings on the items. The edge threshold in the graph generation step is related to the all users' preference, not to the target user. In order to generate the list of recommended items

[1]Note that when we finally build a list of the top-n recommendation item, the states of user nodes are meaningless. The message passing, however, can occur between an item node and a user node, so user nodes should have *like* and *dislike* states same as the item nodes.

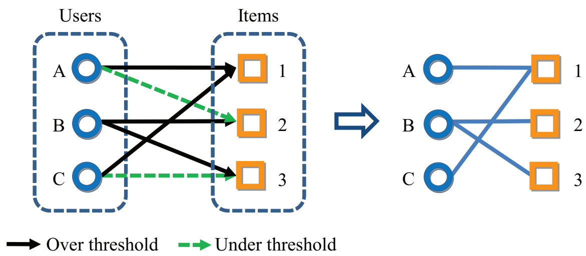


Figure 2: Generating a bipartite graph

personalized to the target user, his/her preference should be reflected in the graph. By assigning the node potentials to nodes based on the target user's ratings and using them in the message passing process, the target user's individual preference can be reflected. There are two node potentials, *like* and *dislike*. Each node potential is allocated as a value between *0.1* and *0.9*. The sum of node potentials of a node should be *1*. Because of matrix multiplication operations during the belief propagation process, the sum of the node potentials should not be *0* or *1*[2]. The higher the rating by the target user is, the higher the *like* node potential is assigned to. The values of node potentials are determined by Equation 3.

$$L_{ij} = 0.5 + \frac{\textit{z-score}(i,j)}{p}, \quad D_{ij} = 0.5 - \frac{\textit{z-score}(i,j)}{p} \qquad (3)$$

L_{ij} and D_{ij} indicate the *like* node potential and the *dislike* node potential, respectively. The *z-score*(i, j) means the normalized value of the target user v_i's rating on item v_j. By using the *z-score*, more accurate relative preference that reflecting the tendency of user's ratings can be acquired. The constant p is a normalizing factor. In the following section, we experimentally show the differences in the accuracy according to the p values. If the node potential value acquired is lower than *0.1* or higher than *0.9*, we replace it with *0.1* or *0.9*. The node potentials of item nodes not rated by the target user and all of user nodes are initialized to be unbiased, i.e., *like* node potential and *dislike* node potential are both *0.5*.

2.4 Propagation matrix

Lastly, we define the propagation matrix related to the edge potential, $\psi_{ij}(x_d, x_c)$ in Equation 1. Table 1 is an example of the propagation matrix, while the possible states are defined as *like* and *dislike*. In $\psi_{ij}(x_d, x_c)$, x_d indicates the row and x_c does the column of the matrix. It means the probability that the target user is likely to prefer the neighboring nodes of a node v_i being in *like* state, $\psi_{ij}(L, L)$ is $0.5 + \alpha$. Generally, in order to avoid numerical underflow, α is defined to be a very small number.

Table 1: Propagation matrix

State	Like	Dislike
Like	0.5 + α	0.5 – α
Dislike	0.5 - α	0.5 + α

[2]This is because *0* makes the multiplication result *0*, and *1* makes another node potential *0*.

3. EXPERIMENTS

The data used in this section is the *MovieLens* data that contains *943* users, *1,682* movies, and *100,000* ratings by users. The ratings are given as an integer between *1* and *5*. We split the data into a training set and a test set in the proportion of *8* to *2*, and we performed a *5*-fold cross validation on them. The answers in the test set are items rated as *5* by the target user. For *alpha* in the propagation matrix, we use *0.0001*.

3.1 Edge threshold

We analyze the accuracy changes according to the edge threshold. There are six criteria when setting the threshold: *1, 2, 3, 4, 5*, and the user's rating average. In this set of experiments, we use parameter p in the node potential calculation as *4*.

Table 2: The accuracy according to the edge threshold

Threshold	Precision (top 10)	Precision (top 20)	Recall (top 10)	Recall (top 20)	MAP
>=1	0.1245	0.0949	0.2128	0.3064	0.1614
>=2	0.1265	0.0961	0.2175	0.3108	0.1646
>=3	0.1296	0.0984	0.2238	0.3176	0.1685
>=4	0.1348	0.1003	0.2317	0.3202	0.1715
=5	0.1280	0.0968	0.2126	0.3058	0.1643
>=average	**0.1362**	**0.1010**	**0.2349**	**0.3239**	**0.1730**

Table 2 shows the *precision, recall*, and *mean average precision (MAP)* of different threshold value settings. As shown in Table 2, the accuracy is the highest when the threshold value is set to be the user's average rating. This means that each user has own rating tendency, so one should apply different edge thresholds for different users in order to reflect the user's preference more accurately. In the following experiments, we use the average rating as the edge threshold.

3.2 Node potential

We analyze the accuracy of the top-n recommendation of our proposed method according to the node potentials. The node potentials capture the target user's preference. We assign a high *like* node potential value to the item nodes rated by the target user, because the target user's ratings reflect his/her preference. The sum of a node's node potentials should be *1*, so we cannot use the original rating that ranges from *1* to *5*. Thus, a transformation from the original ratings to the node potential value should be made. We compared two transformation methods: transforming the original rating to a real number between *0* and *1* or calculating the node potential value using Equation 3 in Section 2.3.

Table 3 is the summary of the experiments. *Direct matching* represents the case where the original ratings between *1, 2, 3, 4* and *5* are transformed to *0.1, 0.3, 0.5, 0.7*, and *0.9*, respectively. $p = 1$ to $p = 5$ indicate the case where the original rating is transformed by Equation 3, and each number represents the value of parameter p.

As shown in Table 3, there are no significant differences. The most accurate results are obtained from the cases where we apply Equation 3 and set p to *3* or *4*. The result indicates that the normalization of the *z-score* of the ratings with parameter p obtains more accurate range of node potential than the use the *z-score* of the ratings only. The optimal

Table 3: The accuracy according to the node prior

Experiment	Precision (top 10)	Precision (top 20)	Recall (top 10)	Recall (top 20)	MAP
Direct matching	0.1278	0.0971	0.2292	0.3214	0.1698
p=1	0.1299	0.0974	0.2229	0.3116	0.1682
p=2	0.1348	0.1003	0.2311	0.3186	0.1723
p=3	0.1357	**0.1014**	**0.2349**	0.3239	**0.1746**
p=4	**0.1362**	0.1010	0.2349	**0.3239**	0.1730
p=5	0.1343	0.1001	0.2308	0.3201	0.1707

normalizing parameter p may vary according to application domains.

3.3 Comparing with existing methods

We evaluated our proposed method by comparing with existing recommendation methods. Among existing recommendation methods, a traditional *item-based* method[3], which was originally proposed for rating prediction rather than top-n recommendation, and an *RWR*-based method were selected for comparisons [4, 6, 9, 12]. For the *item-based* method, we set the parameter k as *30*, the best found in the existing study for the rating prediction performed on the same *MovieLens* data [12]. For the *RWR*, we performed the *RWR*-based top-n recommendation on a graph where the ratings higher than the user's average rating are regarded as edges. Elements of the restart vector (personalization vector) were set as *1* for the target user and *0* for the others. The restart probability was also set as *0.15*. Table 4 shows *precision, recall* and *MAP* of each method.

Table 4: Comparison of the *BP* with existing methods

Method	Precision (top 10)	Precision (top 20)	Recall (top 10)	Recall (top 20)	MAP
BP	**0.1362**	**0.1010**	**0.2348**	**0.3238**	**0.1730**
Item-based	0.0230	0.0223	0.0254	0.0469	0.0296
RWR	0.0968	0.0780	0.1867	0.2836	0.1286

The *RWR*-based method shows the accuracy higher than the *item-based* method. This is because the *RWR*-based method employs iterative random walks through which it infers implicit relationships among nodes. As a result, it can infer implicit relationships more effectively than the item-based method [9]. The proposed method outperforms the *RWR*-based method because the *RWR* considers *homophily* of nodes only, but the *BP* considers both *homophily* and *heterophily* [8]. The *RWR* computes the relevance (similarity) between two connected nodes, which can be regarded as the computation of the likelihood that a node connected to a node preferred by a target user is also preferred by the target user. On the other hand, the *BP* computes both the likelihood that the target user prefers a node and the likelihood that she does not prefer the node. That is, the following all four cases are considered by the BP: 1) the likelihood that the target user prefers a node neighboring the node preferred by herself, 2) the likelihood that the target user prefers a node neighboring the node not preferred by her, 3) the likelihood that the target user does not prefer a node neighboring the node preferred by her, and 4) the likelihood that the target user does not prefer a node neighboring the node not preferred by her. We believe that, because of this difference, the proposed method outperforms the *RWR*-based method.

[3]We omit the result of the *user-based* method here because it was shown to be worse than that of the *item-based* method.

4. CONCLUSIONS

We have proposed a novel method that provides the top-n recommendation by probabilistically determining the target user's preference on items. Compared with the *RWR*-based method, the proposed method is shown to be up to about *40%* more accurate in the *precision*. We expect that more satisfactory recommendation is provided through the proposed method. Furthermore, the application of the *BP* to the personalized recommendation, suggests a new direction for solving personalization problems in various domains.

5. ACKNOWLEDGEMENTS

This research was supported by (1) the MKE (The Ministry of Knowledge Economy), Korea, under the Convergence-ITRC (Convergence Information Technology Research Center) support program (NIPA-2012-H0401-12-1001) supervised by the NIPA (National IT Industry Promotion Agency), (2) the MKE, Korea and Samsung Electronics Co., Ltd., under IT/SW Creative research program (NIPA-2010-C1810-1003-0007) supervised by the NIPA, and (3) the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (NRF-2011-330-B00076).

6. 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 TKDE*, vol.17, no.6, pp.734-749, 2005.

[2] D. Chau, S. Pandit, and C. Faloutsos, "Detecting Fraudulent Personalities in Networks of Online Auctioneers," *ECML/PKDD*, pp.103-114, 2006.

[3] M. Dias et al., "The Value of Personalised Recommender Systems to E-business: a Case Study," *ACM RecSys*, pp.291-294, 2008.

[4] F. Fouss et al., "Random-walk Computation of Similarities between Nodes of a Graph with Application to Collaborative Recommendation," *IEEE TKDE*, vol.19, no.3, pp.355-369, 2007.

[5] A. Gunawardana and G. Shani, "A Survey of Accuracy Evaluation Metrics of Recommendation Tasks," *JMLR*, vol.10, pp.2935-2962, 2009.

[6] I. Konstas, V. Stathopoulos, and J. M. Jose, "On Social Networks and Collaborative Recommendation," *ACM SIGIR*, pp.195-202, 2009.

[7] Y. Koren, "Collaborative Filtering with Temporal Dynamics," *ACM CACM*, vol.53, no.4, pp.89-97, 2010.

[8] D. Koutra et al., "Unifying Guilt-By-Association Approaches: Theorems and Fast Algorithms," *Machine Learning and Knowledge Discovery in Databases*, vol.6921, pp.245-260, 2011.

[9] N. Liu and Q. Yang, "EigenRank: A Ranking-oriented Approach to Collaborative Filtering," *ACM SIGIR*, pp.83-90, 2008.

[10] M. McGlohon et al., "SNARE: A Link Analytic System for Graph Labeling and Risk Detection," *ACM KDD*, pp.1265-1274, 2009.

[11] S. Pandit et al., "NetProbe: A Fast and Scalable System for Fraud Detection in Online Auction Networks," *WWW*, pp.201-210, 2007.

[12] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-Based Collaborative Filtering Recommendation Algorithms," *WWW*, pp.285-295, 2001.