

Modeling User Preference from Rating Data Based on the Bayesian Network with a Latent Variable

Renshang Gao¹, Kun Yue^{1(✉)}, Hao Wu¹, Binbin Zhang¹,
and Xiaodong Fu²

¹ Department of Computer Science and Engineering,
School of Information Science and Engineering,
Yunnan University, Kunming, China
kyue@ynu.edu.cn

² Faculty of Information Engineering and Automation,
Kunming University of Science and Technology, Kunming, China

Abstract. Modeling user behavior and latent preference implied in rating data are the basis of personalized information services. In this paper, we adopt a latent variable to describe user preference and Bayesian network (BN) with a latent variable as the framework for representing the relationships among the observed and the latent variables, and define user preference BN (abbreviated as UPBN). To construct UPBN effectively, we first give the property and initial structure constraint that enable conditional probability distributions (CPDs) related to the latent variable to fit the given data set by the Expectation-Maximization (EM) algorithm. Then, we give the EM-based algorithm for constraint-based maximum likelihood estimation of parameters to learn UPBN's CPDs from the incomplete data w.r.t. the latent variable. Following, we give the algorithm to learn the UPBN's graphical structure by applying the structural EM (SEM) algorithm and the Bayesian Information Criteria (BIC). Experimental results show the effectiveness and efficiency of our method.

Keywords: Rating data · User preference · Latent variable · Bayesian network · Structural EM algorithm · Bayesian information criteria

1 Introduction

With the rapid development of mobile Internet, large volumes of user behavior data are generated and many novel personalized services are generated, such as location-based services and accurate user targeting, etc. Modeling user preference by analyzing user behavior data is the basis and key of these services. Online rating data, an important kind of user behavior data, consists of the descriptive attributes of users themselves, relevant objects (called items) and the scores that users rate on items. For example, MovieLens data set given by GroupLens [2] involves attributes of users and items, as well as the rating scores. The attributes of users include sex, age, occupation, etc., and the attributes of items include type (or genre), epoch, etc. Actually, rating data reflects user preference (e.g., type of items), since a user may rate an item when he is preferred

to this item. Moreover, the rating frequency and corresponding scores w.r.t. a specific type of item also indicate the degree of user preference to this type of item.

In recent years, many researchers proposed various methods for modeling user preference by means of matrix factorization or topic model [13, 17–19, 21]. However, these methods were developed upon the given or predefined preference model (e.g., the topic model is based on a fixed structure), which is not suitable for describing arbitrary dependencies among attributes in data. Meanwhile, the inherent uncertainties among the scores, attributes of users and items cannot be well represented by the given model. Thus, it is necessary to construct a preference model from user behavior data to represent arbitrary dependencies and the corresponding uncertainties.

Bayesian network (BN) is an effective framework for representing and inferring uncertain dependencies among random variables [15]. A BN is a directed acyclic graph (DAG), where nodes represent random variables and edges represent dependencies among variables. Each variable in a BN is associated with a table of conditional probability distributions (CPDs), also called conditional probability table (CPT) to give the probability of each state when given the states of its parents. Making use of BN’s mechanisms of uncertain dependency representation, we are to model user preference by representing the arbitrary dependencies and the corresponding uncertainties.

However, latent variables for describing user preference implied in rating data cannot be observed directly, i.e., hidden or latent w.r.t. the observed data. Fortunately, BN with latent variables (abbreviated as BNLV) [15] are extensively studied in the paradigm of uncertain artificial intelligence. This makes it possible to model user preference by introducing a latent variable into BN to describe user preference and represent the corresponding uncertain dependencies. For example, we could use the BNLV ignoring CPTs shown in Fig. 1 to model user preference, where U_1 , U_2 , I , L and R is used to denote user’s sex, age, movie genre, user preference and the rating score of users on movies respectively. Based on this model, we could fulfill relevant applications based on BN’s inference algorithms.

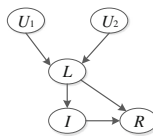


Fig. 1. A BNLV ignoring CPTs

Particularly, we call the BNLV as Fig. 1 as user preference BN (UPBN). To construct UPBN from rating data is exactly the problem that we will solve in this paper. For this purpose, we should construct the DAG structure and compute the corresponding CPTs, as those for learning general BNs from data [12]. However, the introduction of the latent variable into BNs leads to some challenges. For example, learning the parameters in CPTs cannot be fulfilled by using the maximum likelihood estimation directly, since the data of the latent variable is missing w.r.t. the observed data. Thus, we use the Expectation-Maximization (EM) algorithm [5] to learn the parameters and the Structural EM (SEM) algorithm [7] to learn the structure respectively. In this paper, we extend the classical search & scoring method that concerns.

It is worth noting that the value of the latent variable in a UPBN cannot be observed, which derives strong randomness if we learn the parameters by directly using EM and further makes the learned DAG incredible to a great extent. In addition, running SEM with a bad initialization usually leads a trivial structure. In particular, if we set an empty graph as the initial structure, then the latent variable will not have connections with other variables [12]. Thus, we consider the relation between the latent and observed variables, and discuss the property as constraints that a UPBN should satisfy from the perspective of BNLV’s specialties.

Generally speaking, the main contributions can be summarized as follows:

- We propose user preference Bayesian network to represent the dependencies with uncertainties among latent or observed attributes contained in rating data by using a latent variable to describe user preference.
- We give the property and initial structure constraint that make the CPDs related to the latent variable fit the given rating data by EM algorithm.
- We give a constraint-based method to learn UPBN by applying the EM algorithm and SEM algorithm to learn UPBN’s CPDs and DAG respectively.
- We implement the proposed algorithms and make preliminary experiments to test the feasibility of our method.

2 Related Work

Preference modeling has been extensively studied from various perspectives. Zhao et al. [21] proposed a behavior factorization model for predicting user’s multiple topical interests. Yu et al. [19] proposed a user’s context-aware preferences model based on Latent Dirichlet Allocation (LDA) [3]. Tan et al. [17] constructed an interest-based social network model based on Probabilistic Matrix Factorization [16]. Rating data that represents user’s opinion upon items has been widely used for modeling user preference. Matrix factorization and topic model are two kinds of popular methods. Koren et al. [13] proposed the timeSVD ++ model for modeling time drifting user preferences by extending the Singular Value Decomposition method. Yin et al. [18] extended LDA and proposed a temporal context-aware model for analyzing user behaviors. These methods focus on parameter learning of the given or predefined model, but the graph model construction has not been concerned and the arbitrary dependencies among concerning attributes cannot be well described. In this paper, we focus on both parameter and structure learning by incorporating the specialties of rating data.

BN has been studied extensively. For example, Yue et al. [20] proposed a parallel and incremental approach for data-intensive learning of BNs. Breese et al. [4] first applied BN, where each node is corresponding to each item in the domain, to model user preference in a collaborative filtering way. Huang et al. [9] adopted expert knowledge of travel domain to construct a BN for estimating travelers’ preferences. In the general BN without latent variables, user preference cannot be well represented due to the missing of corresponding values.

Meanwhile, there is a growing study on BNLV in recent years. Huete et al. [10] described user’s opinions of one item’s every component by latent variables and

constructed the BNLV for representing user profile in line with expert knowledge. Kim et al. [11] proposed a method about ranking evaluation of institutions based on BNLV where the latent variable represents ranking scores of institutions. Liu et al. [14] constructed a latent tree model, a tree-structured BNLV, from data for multidimensional clustering. These findings provide basis for our study, but the algorithm for constructing BNLV that reflects the specialties of rating data should be explored.

3 Basis for Learning BN with a Latent Variable

3.1 Preliminaries

BIC scoring metric is to measure the coincidence of BN structure with the given data set. The greater the BIC score, the better the structure. Friedman [6] gave the expected BIC scoring function for the case where data is incomplete, defined as follows:

$$BIC(G|D^*) = \sum_{i=1}^m \sum_{X_i} P(X_i|D_i, \theta^*) \log P(X_i, D_i|G, \theta^*) - \frac{d(G)}{2} \log m. \quad (1)$$

where G is a BN, D^* is a complete data obtained by EM algorithm, θ^* is an estimation of model parameter, m is the total number of samples and $d(G)$ is the number of independent parameters required in G . The first term of $BIC(G|D^*)$ is the expected log likelihood, and the second term is penalty of model complexity [12].

As a method to conduct BN's structure learning w.r.t. incomplete data [7], SEM first fixes the current optimal model structure and exerts several optimizations on the model parameter. Then, the optimizations for structure and parameter are carried out simultaneously. The process will be repeated until convergence.

3.2 Properties of BNLV

Let X_1, X_2, \dots, X_n denote observed variables that have dependencies with the latent variable respectively. Let Y denote the set of observed variables that have no dependency with the latent variable, and L denote the latent variable. There are three possible forms of local structures w.r.t. the latent variable in a BNLV, shown as Fig. 2, where the dependencies between observed variables are ignored.

Property 1. The CPTs related to the latent variable can fit data sets by EM if and only if there is at least one edge where the latent variable points to the observed variable, shown as Fig. 2 (a).

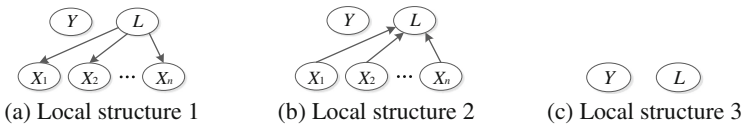


Fig. 2. Local structure related to the latent variable

For the situation in Fig. 2 (a), the CPTs related to the latent variable will be changed in the EM iteration, while the CPTs related to the latent variable will be the same as the initial state in the EM iteration by mathematical derivation of EM for the situations in Fig. 2 (b) and (c). For space limitation, the detailed derivation will not be given here. Accordingly, Property 1 implies that a BNLV must contain the substructure shown in Fig. 2 (a) if we are to make the BNLV fully fit the data set.

4 Constraint-Based Learning of User Preference Bayesian Network

Let $U = \{U_1, U_2, \dots, U_n\}$ denote the set of user's attributes. Let I denote the type of an item, and $I = c_j$ means that the item is of the j th type c_j . Let latent variable L denote user preference to an item, described as the type of the preferring item (i.e., $L = l_j$ means that a user has preference to the item whose type is c_j). Similarly, let R denote the rating score on items. Following, we first give the definition of UPBN, which is used to represent the dependencies among the latent and observed variables.

Definition 1. A user preference Bayesian network, abbreviated as UPBN, is a pair $S = (G, \theta)$, where

- (1) $G = (V, E)$ is the DAG of UPBN, where $V = U \cup \{L\} \cup \{I\} \cup \{R\}$ is the set of nodes in G . E is the directed edge set representing the dependencies among observed attributes and user preference.
- (2) θ is the set of UPBN's parameters constituting the CPT of each node.

4.1 Constraint Description

Without loss of generality, we suppose a user only rates the items that he is interested in. The rating frequency and the corresponding scores for a specific type of items indicate the degree of user preference. Accordingly, we give the constraints to improve the effectiveness of model construction, where constraint 1 means that the initial structure of UPBN learning should be the same as the structure shown in Fig. 3 and constraint 2 means that the CPTs corresponding to I and R should satisfy the inequality for random initialization.

Constraint 1. The initial structure of UPBN is shown as Fig. 3. This constraint demonstrates that the type of a rated item is dependent on user preference and the corresponding rating score is dependent on the type of itself and user preference.

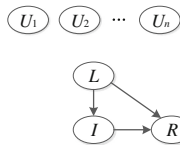


Fig. 3. The initial structure of UPBNs

Constraint 2. Constraint on the initial CPTs:

- (1) $P(I = c_i | L = l_i) > P(I = c_j | L = l_i, i \neq j)$, namely the probability of the users rate c_i will be greater than that of they rate c_j if the user preference value takes l_i .
- (2) If R takes the rating values such as $R \in \{1, 2, 3, 4, 5\}$, then R_1 and R_2 will take values from $\{4, 5\}$ and $\{1, 2, 3\}$, respectively. This means that the users tend to rate high score (4 or 5) instead of rate low score (1, 2, or 3) when their preferences are consistent with the type of items, represented by the following two inequalities:

$$\begin{aligned} P(R = R_1 | I = c_i, L = l_i) &> P(R = R_2 | I = c_i, L = l_i) \text{ and} \\ P(R = R_2 | I = c_i, L = l_j, i \neq j) &> P(R = R_1 | I = c_i, L = l_j, i \neq j) \end{aligned}$$

4.2 Parameter Learning of UPBN

UPBN's parameter learning starts from an initial parameter θ^0 randomly generated under Constraint 2 in Sect. 4.1 and we apply EM to iteratively optimize the initial parameter until convergence.

Suppose that we have conducted t times of iterations and obtained the estimation value θ^t , then the $(t + 1)$ th iteration process will be built as the following E-step and M-step, where there are m samples in data set D , and the cardinality of the variable denoting user preference L is c (i.e., c values of user preference, l_1, l_2, \dots, l_c).

E-step. In light of the current parameter θ^t , we calculate the posterior probability of different user preference value l_j by Eq. (2), $P(L = l_j | D_i, \theta^t)$ ($1 \leq j \leq c$) for every sample D_i ($1 \leq i \leq m$) in D , making data set D complete as D^t . Then we obtain expected sufficient statistics by Eq. (3).

$$P(L = l_j | D_i, \theta^t) = \frac{P(L = l_j, D_i | \theta^t)}{\sum_{j=1}^c P(L = l_j, D_i | \theta^t)}. \quad (2)$$

$$m_{ijk}^t = \sum_{l=1}^m P(V_i = k, \pi(V_i) = j | D_l^t). \quad (3)$$

M-step. Based on the expected sufficient statistics, we can get the new greatest possible parameter θ^{t+1} by Eq. (4).

$$\theta_{ijk}^{t+1} = \frac{m_{ijk}^t}{\sum_{k=1}^{r_i} m_{ijk}^t}. \quad (4)$$

To avoid overfitting and ensure the convergence efficiency of the EM iteration, we give a method to measure parameter similarity. The parameter similarity between θ_1 and θ_2 of a UPBN is defined as the follows:

$$\text{sim}(\theta_1, \theta_2) = |\log P(D | G, \theta_1) - \log P(D | G, \theta_2)| \quad (5)$$

UPBN's parameter learning will converge if $\text{sim}(\theta^{t+1}, \theta^t) < \delta$.

For a UPBN structure G' and data set D , we generate initial parameter randomly under Constraint 2 and make D become the complete data set D^0 . We use Eq. (3) to calculate the expected sufficient statistics and obtain parameter estimation θ^1 by Eq. (4). Then, we use θ^1 to make D become the complete data set D^1 again. By repeating the process until convergence or stop condition is met, the optimal parameter θ will be obtained. The above ideas are given in Algorithm 1.

Algorithm 1. $\text{EM}(G', D, \theta^0, \delta, T)$

Input: G' , UPBN structure

D , rating data

θ^0 , the initial parameter randomly generated under the constraint 2

δ , the threshold of convergence

T , the number of iteration

Output: θ , the optimal parameter of UPBN

Steps:

1: $G \leftarrow G'$

2: $\theta \leftarrow \theta^0$

3: FOR $t = 0$ TO T DO

4: E-step: make D become the complete dataset D^t

5: $m_{ijk}^t = \sum_{l=1}^m P(V_i = k, \pi(V_i) = j | D_i^t)$

6: M-step: $\theta^{t+1} \leftarrow \theta_{ijk}^{t+1} = \frac{m_{ijk}^t}{\sum_{k=1}^r m_{ijk}^t}$

7: IF $\text{sim}(\theta^{t+1}, \theta^t) < \delta$ THEN

8: RETURN θ^{t+1}

9: END IF

10: END FOR

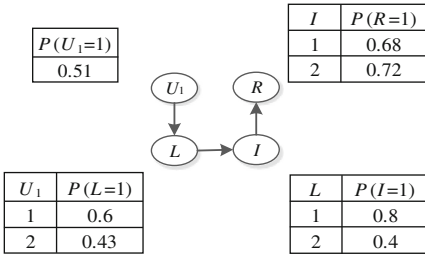


Fig. 4. Current UPBN and θ^1

Table 1. Dataset D

Sample	U_1	I	R	L	Count
D_1	1	1	1		271
D_2	1	1	2		69
D_3	1	2	1		99
D_4	1	2	2		67
D_5	2	1	1		139
D_6	2	1	2		125
D_7	2	2	1		186
D_8	2	2	2		44

Example 1. The current UPBN structure and data set D is presented in Fig. 4 and Table 1 respectively, where *Count* is to depict the number of the same sample. By the E-step in Algorithm 1 upon the initial parameter, we make D become the complete data set D^0 and use Eq. (3) to compute expected sufficient statistics. Then, we obtain parameter θ^1 by Eq. (4), shown in Fig. 4.

4.3 Structure Learning of UPBN

UPBN's structure learning starts from the initial structure and CPTs under the constraints given in Sect. 4.1. First, we rank the order of nodes of the UPBN and make the initial model be the current one. Then, we execute Algorithm 1 to conduct parameter learning of the current model and use BIC to score the current model. Following, we modify the current model by edge addition, deletion and reversal to obtain a series of candidate models which should satisfy Property 1 for the purpose that the candidate ones will be fully fit to the data set.

For each candidate structure G' and the complete data set D^{t-1} , we use Eq. (3) to calculate the expected sufficient statistics and obtain maximum likelihood estimation θ of parameter by Eq. (4) for model selection by BIC scoring metric. The maximum likelihood estimation is presented as Algorithm 2.

Algorithm 2. SEM_ML (G', D^{t-1})

Input: G' , candidate structure of UPBN
 D^{t-1} , the complete dataset derived from EM algorithm.
Output: θ , the maximum likelihood estimation of the current UPBN parameter

Steps:

- 1: $G \leftarrow G'$
- 2: $m_{ijk}^{t-1} = \sum_{l=1}^m P(V_i = k, \pi(V_i) = j | D^{t-1})$
- 3: $\theta \leftarrow \theta_{ijk}^t = \frac{m_{ijk}^{t-1}}{\sum_{k=1}^t m_{ijk}^{t-1}}$
- 4: RETURN θ

By comparing the current model with candidate ones, we adopted that with the maximum BIC score as the basis for the next time of search, which will be made iteratively until the score is not increased. The above ideas are given in Algorithm 3.

Algorithm 3. Learn_UPBN**Input:** G_0 , the initial structure of UPBN D , rating data T , the number of parameter optimization between two times of adjacent structure optimization δ , the threshold of convergence in EM algorithm.**Output:** the optimal UPBN**Steps:**

```

1:  $G \leftarrow G_0$ 
2:  $\theta \leftarrow \theta^{0, 0}$  //the initial parameter  $\theta^{0, 0}$ 
3:  $\theta', D^{T-1} \leftarrow EM(G, D, \theta, \delta, T)$ 
4:  $oldScore \leftarrow BIC(G, \theta' | D^{T-1})$ 
5: WHILE true DO
6:   FOR  $i = 1$  TO  $n$  DO //The order search from each node
7:      $newScore \leftarrow -\infty$ 
8:     FOR each candidate model  $G'$  DO
9:       //Candidate model  $G'$  derived from an edge addition,
10:      edge deletion, or edge reversal on  $G$ 
11:       $\theta' \leftarrow SEM\_ML(G', D^{i, T-1})$ 
12:       $tempScore \leftarrow BIC(G', \theta' | D^{i, T-1})$ 
13:      IF  $tempScore > newScore$  THEN
14:         $G^{i+1} \leftarrow G'; \theta^{i+1, 0} \leftarrow \theta'; newScore \leftarrow tempScore$ 
15:      END IF
16:    END FOR
17:    IF  $newScore > oldScore$  THEN
18:       $\theta^{i+1, T}, D^{i+1, T} \leftarrow EM(G^{i+1}, D, \theta^{i+1, 0}, \delta, T)$ 
19:       $G \leftarrow G^{i+1}; \theta \leftarrow \theta^{i+1, T}$ 
20:       $oldScore \leftarrow BIC(G, \theta | D^{i+1, T})$ 
21:    ELSE
22:      RETURN ( $G, \theta$ )
23:    END IF
24:  END FOR
25: END WHILE

```

Example 2. For the data set D in Table 1 and initial structure of UPBN in Fig. 5(a), we first conduct parameter learning of the initial structure and compute the corresponding BIC score by Algorithm 1. We then execute the three operators on U_1 and obtain three candidate models, shown in Fig. 5(b). Following, we estimate the parameters of the candidate models by Algorithm 2 and compute the corresponding BIC scores by Eq. (1). Thus, we obtain the optimal model G_3' as the current model G . Executing these three operators on other nodes and repeating the process until convergence, an optimal structure of UPBN can be obtained, shown in Fig. 5(c).

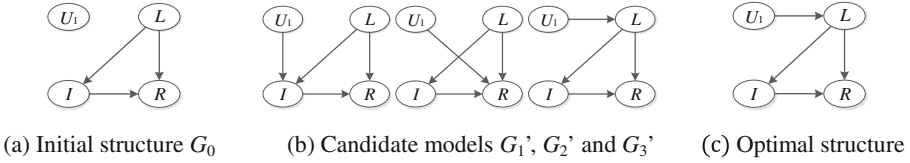


Fig. 5. UPBN's structure learning

5 Experimental Results

5.1 Experiment Setup

To verify the feasibility of the proposed method, we implemented the algorithms for the parameter learning and structure learning of UPBN. The experiment environment is as follows: Intel Core i3-3240 3.40 GHz CPU, 4 GB main memory, running Windows 10 Professional operating system. All codes were written in C++.

All experiments were established on synthetic data. We manually constructed the UPBN shown as Fig. 1 and sampled a series of different scales of data by means of Netica [1]. As for the situation where UPBN contains more than 5 nodes, we randomly generated the corresponding value of sample data. For ease of the exhibition of experimental results, we made use of some abbreviations to denote different test conditions and adopted sign '+' to combine these conditions, where initial CPTs obtained under constraints, initial CPTs obtained randomly, and Property 1 is abbreviated as CCPT, RCPT, P1 respectively. Moreover, we use 1 k to denote 1000 instances.

5.2 Efficiency of UPBN Construction

First, we tested the efficiency of Algorithm 1 for parameter learning with the increase of data size when UPBN contains 5 nodes, and that of Algorithm 1 with the increase of UPBN nodes on 2 k data under different conditions of the initial CPTs, shown in Fig. 6 (a) and (b) respectively. It can be seen that the execution time of Algorithm 1 is increased linearly with the increase of data size. This shows that the efficiency of Algorithm 1 mainly depends on the data size.

Second, we recorded the execution time of Algorithm 1 with the increase of data size and nodes under the condition of CCPT, shown in Fig. 6(c) and (d) respectively. It can be seen that the execution time is increased linearly with the increase of data size no matter how many nodes there are in a UPBN. This means that the execution time is not sensitive to the scale of UPBN.

Third, we tested the efficiency of Algorithm 3 for structure learning with the increase of data size when UPBN contains 5 nodes, and that of Algorithm 3 with the increase of UPBN nodes on 2 k data under different conditions, shown in Fig. 7(a) and (b) respectively. It can be seen from Fig. 7(a) that the execution time of Algorithm 3 is increased linearly with the increase of data size. Moreover, Constraint 2 is obviously beneficial to reduce the execution time under Property 1 when the data set is larger than

6 k. It can be seen from Fig. 7(b) that the execution time of Algorithm 3 is increased sharply with the increase of nodes, and the execution time under CCPT is larger than that under RCPT.

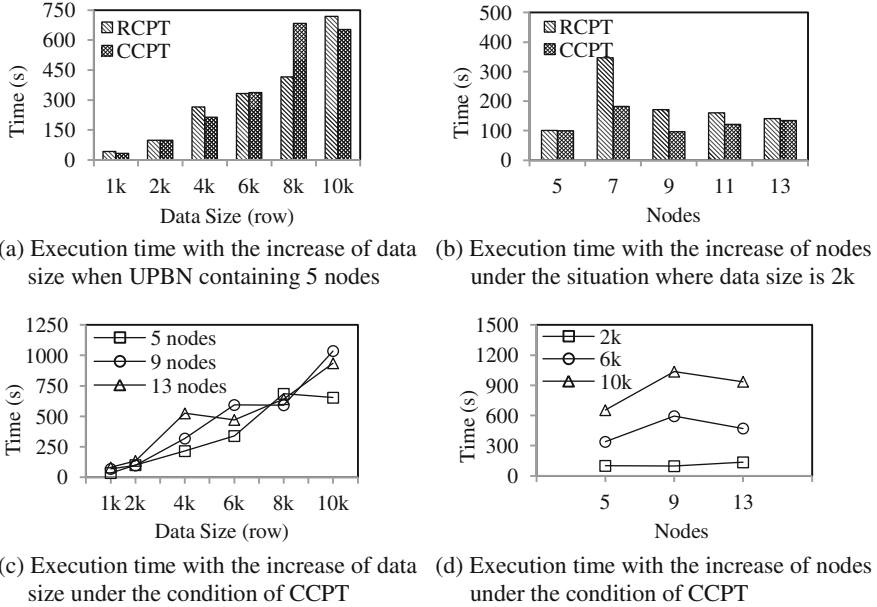


Fig. 6. Execution time of parameter learning

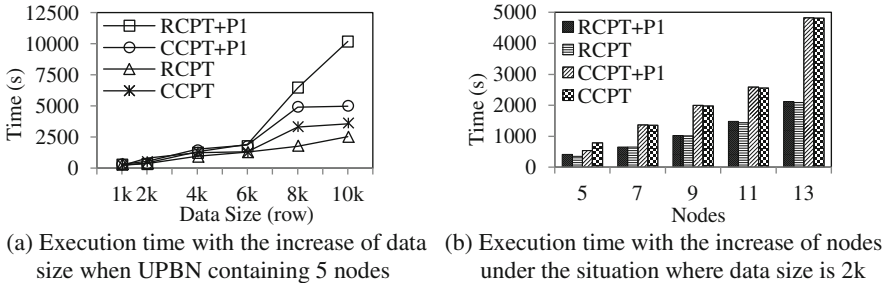


Fig. 7. Execution time of structure learning

5.3 Effectiveness of UPBN Construction

It is pointed out [6] that a BNLV resulted from SEM makes sense under specific initial structures. According to Property 1, a UPBN should include the constraint “ $L \rightarrow X$ ” at least, where L is the latent variable and X is an observed variable. Thus, we introduced the initial structure in Fig. 8 with the least prior knowledge. We constructed 50 UPBNs under the constraint in Fig. 3, denoted as DAG1, and each combination of different

conditions respectively. Meanwhile, we also constructed 50 UPBNs under the constraint in Fig. 8, denoted as DAG2, and each combination of different conditions respectively.

To test the effectiveness of the method for UPBN construction, we constructed the UPBN by the clique-based method [8], shown as Fig. 1. We then compared our constructed UPBNs with this UPBN, and recorded the number of different edges (e.g., no different edges in the UPBN shown in Fig. 1). We counted the number of UPBNs with various number of different edges (0 ~ 8), shown in Table 2. It can be seen that the UPBN constructed upon Fig. 3 is better than that upon Fig. 8 under the same conditions, since the former derives less different edges than the latter. Moreover, the number of the constructed UPBNs with less different edges under CCPT is obviously larger than that under RCPT (e.g., the number of UPBNs with 0 different edges under DAG1 + CCPT is greater than that under DAG1 + RCPT), which means that our constraint-based method is beneficial and better than the traditional method by EM directly in parameter learning for UPBN construction. Thus, our method for UPBN construction is effective w.r.t. user preference modeling from rating data.

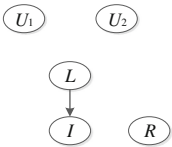


Fig. 8. Initial structure with the least constraint

Table 2. Structures of learned UPBN under different conditions

Condition	The number of different edge								
	0	1	2	3	4	5	6	7	8
DAG1 + CCPT + P1	18		27	3	2				
DAG1 + CCPT	18		27	3	2				
DAG1 + RCPT + P1			2	1	47				
DAG1 + RCPT			2	1	47				
DAG2 + CCPT + P1			12		11	5	13	9	
DAG2 + CCPT			5		10	16	15	2	2
DAG2 + RCPT + P1			1		1	4	17	27	
DAG2 + RCPT			1		1	4	17	18	9

6 Conclusions and Future Work

In this paper, we aimed to give a constraint-based method for modeling user preference from rating data to provide underlying techniques for the novel personalized services in the context of mobile Internet like applications. Accordingly, we gave the property that enables CPTs related to the latent variable to fit data sets by EM and constructed UPBN to represent arbitrary dependencies between user preference and explicit attributes in rating data. Experimental results showed the efficiency and effectiveness. However, only test on synthetic data is not enough to verify the feasibility of our method in realistic situations. So, we will make more experiments on real rating data sets further. As well, modeling preference from massive, distributed and dynamic rating data is what we are currently exploring.

Acknowledgements. This paper was supported by the National Natural Science Foundation of China (Nos. 61472345, 61562090, 61462056, 61402398), Natural Science Foundation of Yunnan Province (Nos. 2014FA023, 2013FB009, 2013FB010), Program for Innovative Research Team in Yunnan University (No. XT412011), and Program for Excellent Young Talents of Yunnan University (No. XT412003).

References

1. Netica Application (2016). <http://www.norsys.com/netica.html>
2. MovieLens Dataset (2016). <http://grouplens.org/datasets/movielens/1m>
3. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. *J. Mach. Learn. Res.* **3**, 993–1022 (2003)
4. Breese, J., Heckerman, D., Kadie, C.M.: Empirical analysis of predictive algorithms for collaborative filtering. In: *UAI 1998*, pp. 43–52. Morgan Kaufmann (1998)
5. Dempster, A., Laird, N., Rubin, D.: Maximum-likelihood from Incomplete Data via the EM algorithm. *J. Royal Stat. Soc.* **39**(1), 1–38 (1977)
6. Friedman, N.: Learning belief networks in the presence of missing values and hidden variables. In: *ICML 1997*, pp. 452–459. ACM (1997)
7. Friedman, N.: The Bayesian structural EM algorithm. In: *UAI 1998*, pp. 129–138. Morgan Kaufmann (1998)
8. Elidan, G., Lotner, N., Friedman, N., Koller, D.: Discovering Hidden variables: a structure-based approach. In: *NIPS 2000*, pp. 479–485 (2000)
9. Huang, Y., Bian, L.: A bayesian network and analytic hierarchy process based personalized recommendations for tourist attractions over the internet. *Expert Syst. Appl.* **36**(1), 933–943 (2009)
10. Huete, J., Campos, L., Fernandez-luna, J.M.: Using structural content information for learning user profiles. In: *SIGIR 2007*, pp. 38–45 (2007)
11. Kim, J., Jun, C.: Ranking evaluation of institutions based on a bayesian network having a latent variable. *Knowl. Based Syst.* **50**, 87–99 (2013)
12. Koller, D., Friedman, N.: *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, Cambridge (2009)
13. Koren, Y.: Collaborative filtering with temporal dynamics. *Commun. ACM* **53**(4), 89–97 (2010)
14. Liu, T., Zhang, N.L., Chen, L., Liu, A.H., Poon, L., Wang, Y.: Greedy learning of latent tree models for multidimensional clustering. *Mach. Learn.* **98**(1–2), 301–330 (2015)
15. Pearl, J.: Fusion, propagation, and structuring in belief networks. *Artif. Intell.* **29**(3), 241–288 (1986)
16. Salakhutdinov, R., Mnih, A.: Probabilistic Matrix Factorization. In: *NIPS 2007*, pp. 1257–1264 (2007)
17. Tan, F., Li, L., Zhang, Z., Guo, Y.: A multi-attribute probabilistic matrix factorization model for personalized recommendation. In: Dong, X.L., Yu, X., Li, J., Sun, Y. (eds.) *WAIM 2015*. LNCS, vol. 9098, pp. 535–539. Springer, Heidelberg (2015). doi:[10.1007/978-3-319-21042-1_57](https://doi.org/10.1007/978-3-319-21042-1_57)
18. Yin, H., Cui, B., Chen, L., Hu, Z., Huang, Z.: A Temporal context-aware model for user behavior modeling in social media systems. In: *SIGMOD 2014*, pp. 1543–1554. ACM (2014)

19. Yu, K., Zhang, B., Zhu, H., Cao, H., Tian, J.: Towards personalized context-aware recommendation by mining context logs through topic models. In: Tan, P.-N., Chawla, S., Ho, C.K., Bailey, J. (eds.) PAKDD 2012. LNCS (LNAI), vol. 7301, pp. 431–443. Springer, Heidelberg (2012). doi:[10.1007/978-3-642-30217-6_36](https://doi.org/10.1007/978-3-642-30217-6_36)
20. Yue, K., Fang, Q., Wang, X., Li, J., Liu, W.: A parallel and incremental approach for data-intensive learning of bayesian networks. *IEEE Trans. Cybern.* **45**(12), 2890–2904 (2015)
21. Zhao, Z., Cheng, Z., Hong, L., Chi, E.H.: Improving user topic interest profiles by behavior factorization. In: *WWW 2015*, pp. 1406–1416. ACM (2015)