

A New Approach for Collaborative Filtering based on Bayesian Network Inference

Loc Nguyen ¹, Phung Do ²

¹ Sunflower Soft Company, Ho Chi Minh city, Vietnam. Email: ng_phloc@yahoo.com

² University of Information Technology, Ho Chi Minh city, Vietnam. Email: dtminhphung@yahoo.com

Abstract

Collaborative filtering (CF) is one of the most popular algorithms, for recommendation in cases, the items which are recommended to users, have been determined by relying on the outcomes done on surveying their communities. There are two main CF-approaches, which are memory-based and model-based.

The model-based approach is more dominant by real-time response when it takes advantage of inference mechanism in recommendation task. However the problem of incomplete data is still an open research and the inference engine is being improved more and more so as to gain high accuracy and high speed.

I propose a new model-based CF based on applying Bayesian network (BN) into reference engine with assertion that BN is an optimal inference model because BN is user's purchase pattern and Bayesian inference is evidence-based inferring mechanism which is appropriate to rating database. Because the quality of BN relies on the completion of training data, it gets low if training data have a lot of missing values. So I also suggest an average technique to fill in missing values.

Keywords: collaborative filtering, Bayesian network

1. Introduction

The recommendation system is a system which recommends to the users, all the items which are those among a large number of existing items in database. Items which are to point to anything that users are to considering such as products, services, books, news papers, etc. And there has been also an expectation that the recommended items will be likely the ones that the users would be like the most. Another words, such mentioned items are going to go along with the users' interests.

By those meanings, there are two recommendations systems, found to be with a common trends: content-base filtering (CBF) and collaborative filtering (CF) (Su & Khoshgoftaar, 2009, pp. 3-13) (Ricci, Rokach, Shapira, & Kantor, 2011, pp. 73-139):

- The CBF recommends an item to a user if such item has similarities in contents to other items that he like most in the past (and his rating for such item is high). Note that each item has contents which are their properties and so all items will compose a matrix, called the items content matrix.
- The CF on the other hands, recommends an item to user if his neighbors (mean the other users that are similar to him) are interested in such item. Notes that, user's rating on any item does express his interest on that item. For that reason, all user's ratings which carry out on the items will also composes a matrix, called the rating matrix.

Both of the above mentioned filtering (CBF & CF) do have their own strong, as well weak points. The CBF is the one to focus on the item's contents and user personality's interests. And it is designed to recommend different items to different users. Each user therefore, can receive a unique recommendations; and this is also the strong point of CBF filtering. However CBF doesn't tend towards community like CF does. As the items that users may like "are hidden under" user community, CBF hasn't been capable of discovering such implicit items. Because of this, it is acknowledged as a common weak point of CBF. Moreover, in case the number of users becomes larger at a huge volume, CBF may give a wrong prediction; else the accuracy of CF will get increased.

If there will be a huge contents associated with items, for instance and these items have had various properties then CBF in return, will consume even much more system resources, as well the processing time in order to analyze these items whereas, CF as a matter of fact, doesn't pay any regard to the contents of the meant items. Instead the CF only works on the users' ratings of the items and it is known as the strong point of this CF type. Because of that, CF wouldn't be encountering with problems, such as how to analyze the richness in items' contents. However this is also to reflecting the weak points of CF type as well, simply because CF can also do some unexpected recommendations in some situations, in which items are to be considered suitable to users, but they don't relate to users' profiles in fact. The problem then even turns into more serious trouble when having to facing with too many items which aren't rated. It turns the rating matrix into the spare one which is to containing various missing values. In order to alleviate this weakness of the CF type, there have been two techniques which could be helpful, used for improvements:

- The combinations of the CF and CBF types. This technique is breaking into two stages. First, it applies CBF to setting up a complete rating matrix, and then the next step would be the CF type, which is used to making predictions for recommendations. This mentioned technique will be positively useful to improve the predictions' precision. But it does consuming more time when the first stage plays the role of the filtering step or pre-processing step while the content of items must be fully represented as a requirement. This technique is designed to requiring both, the items' content matrix, and the rating matrix.
- Compressing the rating matrix into a representative model, which then is used to predict all the missing data for recommendations. This is a model-based approach for the CF type. Note that to this CF type, there have been two common approaches, such as the memory-based and the model-based approaches. The model-based approach applies statistical and machine learning methods to mining the rating matrix. The result of this mining task is the above mentioned model.

Although the model-based approach doesn't give result which is as precise as the combination approach, it can solve the problem of huge database and sparse matrix. Moreover it can responds user's request immediately by making prediction on representative model though instant inference mechanism. So this paper focuses on model-based approach for CF based on Bayesian network inference. There are many other researches which apply Bayesian network (BN) into CF. Authors (Miyahara & Pazzani, 2000) propose the Simple Bayesian Classifier for CF. Suppose rating values range in the integer interval $\{1, 2, 3, 4, 5\}$, there is a set of 5 respective classes $\{c_1, c_2, c_3, c_4, c_5\}$. The Simple Bayesian Classifier uses Naïve Bayesian classification method (Miyahara & Pazzani, 2000, p. 4) to determine which class a given user belongs to. Mentioned in (Su & Khoshgoftaar, 2009, p. 9), the NB-ELR algorithm is an improvement of Simple Bayesian Classifier, which combines Naïve Bayesian classification and extended logistic regression (ELR). ELR is a gradient-ascent algorithm, which is a discriminative parameter-learning algorithm that maximizes log conditional likelihood (Su & Khoshgoftaar, 2009, p. 9). NB-ELR algorithm gains high classification accuracy on both complete and incomplete data. Author (Langseth, 2009) assumes that there is a linear mapping from the latent space of users and items to the numerical rating scale. Such mapping which conforms the full joint distribution over all ratings constructs a BN. Parameters of joint distribution are learned from training data, which are used for predicting active users' ratings. According to (Campos, Fernández-Luna, Huete, & Rueda-Morales, 2010), the hybrid recommender model is the BN that includes three main kinds

of nodes such as feature nodes, item nodes, and user nodes. Each feature node represents an attribute of item. Active users' ratings are dependent on these nodes.

In general, other researches focus on classification based on BN, discovering latent variables, and predicting active users' ratings while this research focuses on using BN to model users' purchase pattern and taking advantages of inference mechanism of BN. It is the potential approach because it opens a new point of view about recommendation domain. In section 2 I propose an idea for the model-based CF algorithm based on Bayesian network. Section 3 tells about the enhancement of our method. Section 4 is the evaluation and its results. Section 5 is the conclusion.

Note that in this paper, terms such as rating matrix, rating database, training data, and training data set have the same meaning. Suppose we have a rating matrix in which rows indicate users and columns indicate items and each cell is the rating which user gave to item. Each row represents a user vector or rating vector that models a user; so these vectors are considered as user profiles. The user to whom we recommend items is called active user. The vector of active user is called as active user vector. An example of rating matrix is shown in table 1.

	<i>item</i> ₁	<i>item</i> ₂	<i>item</i> ₃
<i>user</i> ₁	$r_{11} = 1$	$r_{12} = 2$	$r_{13} = 1$
<i>user</i> ₂	$r_{21} = 2$	$r_{22} = 1$	$r_{23} = 2$
<i>user</i> ₃	$r_{31} = 4$	$r_{32} = 1$	$r_{33} = 5$
<i>user</i> ₄	$r_{41} = 1$	$r_{42} = 2$	$r_{43} = ?$ (missing)

Table 1. Rating matrix (user 4 is active user)

Let $\vec{u}_i = (r_{i1}, r_{i2}, \dots, r_{in})$ and $\vec{a} = (r_{a1}, r_{a2}, \dots, r_{an})$ be the normal user vector i and the active user a , respectively where r_{ij} is the rating of user i to item j . The question mark (?) indicates missing values and the goal of all CF approaches is to predict or estimate such values. Rating matrix in above table is called user-based matrix because each row is user vector. Otherwise, item-based matrix contains item vector.

2. A new CF algorithm based on Bayesian network

The basic idea of model-based CF is to try to find out an optimal inference model which can give real-time response. Besides, *sparse matrix* and *black sheep* are considered as important problems which need to be solved. I propose a new model-based CF algorithm based on Bayesian network (Neapolitan, 2003, p. 40) inference so as to gain high accuracy and solve the problem of sparse matrix. In general, our method aims to build up Bayesian network (BN) from rating matrix. Each node of such BN represents an item and each arc expresses the dependence relationship between two nodes. Whenever recommendation task is required, the inference mechanism of BN will determine which items are recommended to user, based on the posterior probabilities of such items. The larger the posterior probability of an item is, the higher it's likely that this item is bought by many users. So such item has high frequency and it should be recommended to new users. If the rating matrix is sparse, we try to replace missing values by estimated values so that it is easy and efficient to build up BN from complete matrix instead of from sparse matrix. The technique of how to estimate missing values is discussed later. New algorithm includes 4 steps:


1. Transposing user-based matrix to item-based matrix.
2. Filling in missing values.
3. Learning BN from item-based matrix.
4. Performing recommendation task.

Steps 1, 2, 3 are offline-mode processes and so they don't affect the ability of real-time response in step 4. These steps are described in following sub-sections 2.1, 2.2, 2.3, and 2.4.

2.1. Transposing user-based matrix to item-based matrix

User-based matrix is the original format of rating matrix (see table 1). Each row in user-based matrix is ratings that a concrete user giving to many items. Otherwise, for item-based matrix, each row is ratings that a concrete item receiving from many users. User-based matrix transposed into item-based matrix in this step is considered as simple pre-processing step which is simple but very important because BN is constituted of item nodes. In real context, the number of customers is unlimited and increased much more than the number of items. We use item-based matrix in order to keep the size of BN in small so that the speed of inference is improved in recommendation task (see step 4). Table 2 is an example of transposing user-based matrix to item-based matrix. This example is used throughout this paper.

	<i>item</i> ₁	<i>item</i> ₂	<i>item</i> ₃
<i>user</i> ₁	$r_{11} = 1$	$r_{12} = 3$	$r_{13} = ?$
<i>user</i> ₂	$r_{21} = 3$	$r_{22} = ?$	$r_{23} = 5$
<i>user</i> ₃	$r_{31} = 4$	$r_{32} = 2$	$r_{33} = 1$
<i>user</i> ₄	$r_{41} = ?$	$r_{42} = ?$	$r_{43} = 3$



	<i>user</i> ₁	<i>user</i> ₂	<i>user</i> ₃	<i>user</i> ₄
<i>item</i> ₁	$r_{11} = 1$	$r_{21} = 3$	$r_{31} = 4$	$r_{41} = ?$
<i>item</i> ₂	$r_{12} = 3$	$r_{22} = ?$	$r_{32} = 2$	$r_{42} = ?$
<i>item</i> ₃	$r_{13} = ?$	$r_{23} = 5$	$r_{33} = 1$	$r_{43} = 3$

Table 2. Transposing user-based matrix to item-based matrix

2.2. Filling in missing values

The BN learned from complete rating matrix is more adequate than the one learned from sparse matrix. Some methods can learn BN from incomplete data while other methods require complete data. In case of requirement of complete data, the simplest way to fill in incomplete data is to replace missing values by average values. An average value is an estimate of missing value. The replacement is iterative and overlap procedure, which is considered as estimation process:

- Replacement is done via many iterations. Replacing missing values with average values in next iteration is based on estimated values in previous iteration.
- Average value is calculated as the mean of user vector. If user vector is empty then the mean of item vector becomes an estimate of average value.

Given item-based matrix in step 1, for example, r_{41} and r_{42} are replaced as follows:

- $r_{41} = r_{43} / 1 = 3$
- $r_{42} = (r_{41} + r_{43}) / 2 = (3+3) / 2 = 3$

Note that r_{42} is computed based on the replaced value of r_{41} ; it is the overlapping. By other way, rating r_{42} can be the same to r_{41} which is the mean of user vector *user*₄. If so, the speed of algorithm gets much more faster because the mean is computed only one time. However, this overlapping gets more efficient and solid. If vector *user*₄ is empty, its mean is undefined but there is no interruption in replacement process. Therefore, r_{42} is assigned by the mean of item-vector *item*₂, so $r_{42} = (r_{12} + r_{32}) / 2 = 2.5$ and computation will continue. By this way, remaining missing values such as r_{22} , r_{13} are estimated.

- $r_{22} = (r_{21} + r_{23}) / 2 = (3 + 5) / 2 = 4$
- $r_{13} = (r_{11} + r_{12}) / 2 = (1 + 3) / 2 = 2$

Thus, we have completely estimated item-based rating matrix shown in table 3.

	<i>user</i> ₁	<i>user</i> ₂	<i>user</i> ₃	<i>user</i> ₄
<i>item</i> ₁	$r_{11} = 1$	$r_{21} = 3$	$r_{31} = 4$	$r_{41} = \mathbf{3}$
<i>item</i> ₂	$r_{12} = 3$	$r_{22} = \mathbf{4}$	$r_{32} = 2$	$r_{42} = \mathbf{3}$
<i>item</i> ₃	$r_{13} = \mathbf{2}$	$r_{23} = 5$	$r_{33} = 1$	$r_{43} = 3$

Table 3. Completely estimated item-based rating matrix

This average technique is fast but not accurate because replaced values don't reflect the real values that users rate on an item. Learning methods which can undertake incomplete data in order to construct BN are recommended but they go beyond this research.

2.3. Learning BN from item-based matrix

Building up the BN from the complete item-based matrix created in step 1 and step 2. In general case, each node in BN has five values $\{1, 2, 3, 4, 5\}$ corresponding to user's rating values: 5-most favorite and 1-most disliked. Every node is associated with conditional probability table (CPT) which defines prior probabilities.

BN is built up by machine learning techniques. There are two BN learning approaches:

- Score-based approach: given scoring criterion δ assigned to every BN, which BN gains highest δ is the best BN. This criterion δ is computed as the posterior probability over whole BN given training data set such as item-based matrix.
- Constraint-based approach: given a set of constraints, which BN satisfies over all such constraints is the best BN. Constraints are defined as rules relating to Markov condition.

The basic idea of such learning approaches is to find out the most adequate BN structure. Each approach has a lot of algorithms which aren't described here because they go beyond this paper. In general, learning structure algorithm is the most important in our method because the BN structure influences most on the recommendation result. Now we apply scored-based approach into complete item-based rating matrix (see table 3) as simple example for learning BN. For convenience, item-based rating matrix in table 3 is translated into binary matrix (table 4) whose each cell gets 1 (0) if its value is greater than or equal to 3 (and otherwise).

	<i>user₁</i>	<i>user₂</i>	<i>user₃</i>	<i>user₄</i>
<i>item₁</i>	0	1	1	1
<i>item₂</i>	1	1	0	1
<i>item₃</i>	0	1	0	1

Table 4. Item-based binary rating matrix

In general case, rating values range in a pre-defined interval. For example, given the integer interval $\{1, 2, 3, 4, 5\}$, each item will be split into 5 sub-items in accordance with 5 rating values. Each sub-item has binary values 0 and 1. Consequently, the rating matrix is still transformed into binary matrix as shown in table 4 in which each row represents user ratings on a sub-item. \

Suppose BN is learned from binary matrix in table 4. Let I_1 , I_2 and I_3 denote item 1, item 2 and item 3, respectively. Note that I_1 , I_2 and I_3 are binary variables (nodes) whose values are 0 or 1. The essence of scored-based approach is to find out the best BN based on scoring criterion δ among search space that contains a set of possible BN (s). Suppose the search space includes three Bayesian networks: BN_1 , BN_2 and BN_3 shown in figure 1. Note that figure 1 followed by computational formulas is only used for illustrating how to learn Bayesian network. There is really the huge number of possible BN (s) in case of big training data. So effective scored-based algorithms which reduce the search space are applied into learning BN in practice. Concretely, the research applies the K2 learning algorithm built in the Elvira system (Serafin, Carmelo, Pedro, & Francisco, 2003) into constructing BN. The software package Elvira is the fruit of a research project supported by the CICYT (a Spanish national research agency) and the Spanish Ministry of Science and Technology as a join effort of several Spanish universities, in collaboration with two Mexican researchers.

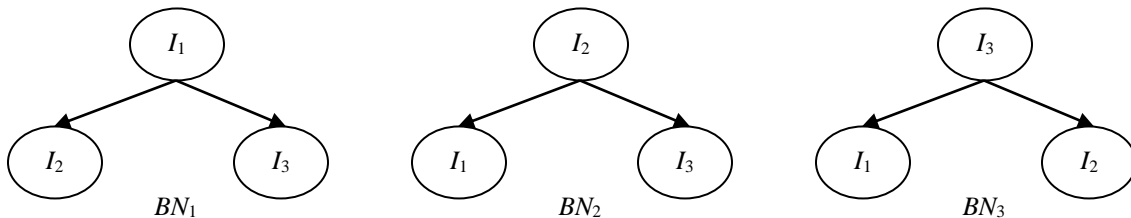


Figure 1. Bayesian networks search space: BN_1 , BN_2 and BN_3

Following is the illustration of scored-based approach. Let D denote binary rating matrix in table 4. Now the CPT of every node is calculated based on following formulas:

$$P(I_k = a) = \frac{\text{Number of users giving value } a \text{ to item } I_k}{\text{Total number of users}}$$

$$P(I_v = b | I_u = a) = \frac{\text{Number of users giving values } a \text{ and } b \text{ to item } I_u \text{ and item } I_v, \text{ respectively}}{\text{Number of users giving value } a \text{ to item } I_u}$$

Where a and b are binary values $\{0, 1\}$. Absolutely, we have results:

- $P(I_1 = 1) = 3/4, P(I_1 = 0) = 1/4$
- $P(I_2 = 1) = 3/4, P(I_2 = 0) = 1/4$
- $P(I_3 = 1) = 1/2, P(I_3 = 0) = 1/2$
- $P(I_2 = 1 | I_1 = 1) = 2/3, P(I_2 = 0 | I_1 = 1) = 1/3, P(I_2 = 1 | I_1 = 0) = 1, P(I_2 = 0 | I_1 = 0) = 0$
- $P(I_3 = 1 | I_1 = 1) = 2/3, P(I_3 = 0 | I_1 = 1) = 1/3, P(I_3 = 1 | I_1 = 0) = 0, P(I_3 = 0 | I_1 = 0) = 1$
- $P(I_1 = 1 | I_2 = 1) = 2/3, P(I_1 = 0 | I_2 = 1) = 1/3, P(I_1 = 1 | I_2 = 0) = 1, P(I_1 = 0 | I_2 = 0) = 0$
- $P(I_3 = 1 | I_2 = 1) = 2/3, P(I_3 = 0 | I_2 = 1) = 1/3, P(I_3 = 1 | I_2 = 0) = 0, P(I_3 = 0 | I_2 = 0) = 1$
- $P(I_1 = 1 | I_3 = 1) = 1, P(I_1 = 0 | I_3 = 1) = 0, P(I_1 = 1 | I_3 = 0) = 1/2, P(I_1 = 0 | I_3 = 0) = 1/2$
- $P(I_2 = 1 | I_3 = 1) = 1, P(I_2 = 0 | I_3 = 1) = 0, P(I_2 = 1 | I_3 = 0) = 1/2, P(I_2 = 0 | I_3 = 0) = 1/2$

Let g_1 , g_2 and g_3 be joint probability functions of BN_1 , BN_2 and BN_3 , respectively. We have $g_1(I_1, I_2, I_3) = P(I_1)P(I_2|I_1)P(I_3|I_1)$, $g_2(I_1, I_2, I_3) = P(I_2)P(I_1|I_2)P(I_3|I_2)$ and $g_3(I_1, I_2, I_3) = P(I_3)P(I_1|I_3)P(I_2|I_3)$. Let δ_1 , δ_2 and δ_3 be scoring criterions of BN_1 , BN_2 and BN_3 , respectively. These scoring criterions represent posterior probabilities of g_1 , g_2 and g_3 given D being binary rating matrix in table 4. Applying Bayes' theorem, the posterior probability of Bayesian network BN_i given D is determined:

$$P(BN_i | D) = g_i(I_1, I_2, I_3 | D) = \frac{g_i(D | I_1, I_2, I_3) g_i(I_1, I_2, I_3)}{g_i(D)}$$

Suppose that the prior probabilities $g_i(I_1, I_2, I_3)$ (s) are the same and the probability over training data $g_i(D)$ is constants, it is possible to remove $g_i(I_1, I_2, I_3)$ and $g_i(D)$ from the posterior probability of Bayesian network BN_i . No loss of generality, scoring criterion δ_i is defined as the likelihood function $g_i(D | I_1, I_2, I_3)$.

$$\delta_i = g_i(D | I_1, I_2, I_3)$$

If each column (corresponding to user) of D is considered as a *case* of training set, there are four cases: $c_1 = (I_1 = 0, I_2 = 1, I_3 = 0)$, $c_2 = (I_1 = 1, I_2 = 1, I_3 = 1)$, $c_3 = (I_1 = 1, I_2 = 0, I_3 = 0)$ and $c_4 = (I_1 = 1, I_2 = 1, I_3 = 1)$. Suppose cases are mutually independent, scoring criterion is product of posterior probabilities in flavor of cases.

$$\delta_i = g_i(D | I_1, I_2, I_3) = \prod_{j=1}^4 g_i(c_j | I_1, I_2, I_3)$$

For instance, by substituting the joint probability of BN_1 , namely $g_1(I_1, I_2, I_3) = P(I_1)P(I_2/I_1)P(I_3/I_1)$, into the formula of scoring criterion, we have:

$$\begin{aligned}
\delta_1 &= \prod_{j=1}^4 g_1(c_j | I_1, I_2, I_3) \\
&= g_1(I_1 = 0, I_2 = 1, I_3 = 0) * g_1(I_1 = 1, I_2 = 1, I_3 = 1) * g_1(I_1 = 1, I_2 = 0, I_3 = 0) * g_1(I_1 = 1, I_2 = 1, I_3 = 1) \\
&= P^3(I_1 = 1) * P(I_1 = 0) * P^2(I_2 = 1 | I_1 = 1) * P(I_2 = 1 | I_1 = 0) * P(I_2 = 0 | I_1 = 1) * P^2(I_3 = 1 | I_1 = 1) \\
&\quad * P(I_3 = 0 | I_1 = 1) * P(I_3 = 0 | I_1 = 0) \\
&= \left(\frac{3}{4}\right)^2 * \frac{1}{4} * \left(\frac{2}{3}\right)^2 * 1 * \frac{1}{3} * \left(\frac{2}{3}\right)^2 * \frac{1}{3} * 1 = 0.0031
\end{aligned}$$

Scoring criterions of BN_2 and BN_3 are calculated in the similar way.

$$\begin{aligned}
\delta_2 &= \prod_{j=1}^4 g_2(c_j | I_1, I_2, I_3) \\
&= g_2(I_1 = 0, I_2 = 1, I_3 = 0) * g_2(I_1 = 1, I_2 = 1, I_3 = 1) * g_2(I_1 = 1, I_2 = 0, I_3 = 0) * g_2(I_1 = 1, I_2 = 1, I_3 = 1) \\
&= P^3(I_2 = 1) * P(I_2 = 0) * P^2(I_1 = 1 | I_2 = 1) * P(I_1 = 0 | I_2 = 0) * P(I_1 = 0 | I_2 = 1) * P^2(I_3 = 1 | I_2 = 1) \\
&\quad * P(I_3 = 0 | I_2 = 1) * P(I_3 = 0 | I_2 = 0) \\
&= \left(\frac{3}{4}\right)^3 * \frac{1}{4} * \left(\frac{2}{3}\right)^2 * 0 * \frac{1}{3} * \left(\frac{2}{3}\right)^2 * \frac{1}{3} * 1 = 0
\end{aligned}$$

$$\begin{aligned}
\delta_3 &= \prod_{j=1}^4 g_3(c_j | I_1, I_2, I_3) \\
&= g_3(I_1 = 0, I_2 = 1, I_3 = 0) * g_3(I_1 = 1, I_2 = 1, I_3 = 1) * g_3(I_1 = 1, I_2 = 0, I_3 = 0) * g_3(I_1 = 1, I_2 = 1, I_3 = 1) \\
&= P^2(I_3 = 1) * P^2(I_3 = 0) * P^2(I_1 = 1 | I_3 = 1) * P(I_1 = 1 | I_3 = 0) * P(I_1 = 0 | I_3 = 0) * P^2(I_2 = 1 | I_3 = 1) \\
&\quad * P(I_2 = 1 | I_3 = 0) * P(I_2 = 0 | I_3 = 0) \\
&= \left(\frac{1}{2}\right)^2 * \left(\frac{1}{2}\right)^2 * 1^2 * \frac{1}{2} * \frac{1}{2} * 1^2 * \frac{1}{2} * \frac{1}{2} = 0.0039
\end{aligned}$$

It is easy to recognize that $\delta_3 = 0.0039$ is the maximum score and so BN_3 is chosen as the best Bayesian network learned from item-based rating matrix. Figure 2 depicts the best Bayesian network learned from item-based rating matrix.

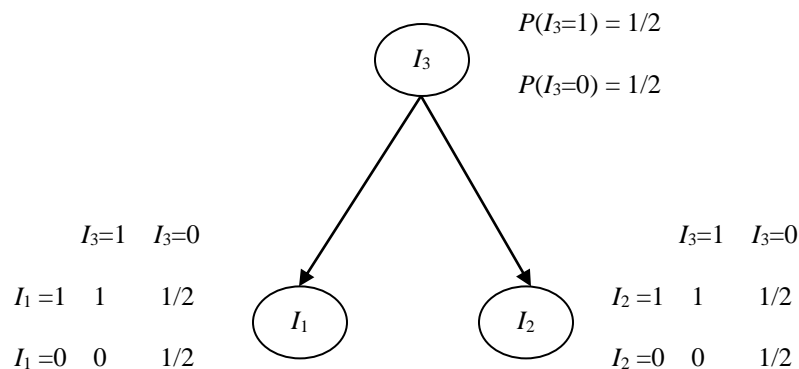


Figure 2. Best Bayesian network learned from item-based rating matrix and its CPT

2. 4. Performing recommendation task

Recommendation task is performed according to evidence-based inference in BN. Firstly, it determines posterior probabilities (PoP) of nodes in networks and secondly, recommends which nodes have high PoP to users. Whole BN is considered user's purchase pattern and existing her/his rated items are considered evidences. This method has two advantages:

- Using BN being itself purchase pattern can discover user interests and predict her/his purchase trend in future. So the quality of recommendation is improved.
- Evidence-based inference in BN is a solid and powerful deduction mechanism. This decreases mean square of error when estimating missing ratings.

Suppose that active user U has already rated item 1 with value 3, recommendation system is responsible for determining which items are introduced to active user U . Suppose the best BN learned from item-based rating matrix (see figure 2) is chosen as target BN for recommendation. Because item 1 is rated with value 3, node I_1 becomes a binary evidence node whose value is 1. Figure 3 depicts the target BN whose evidence is shaded.

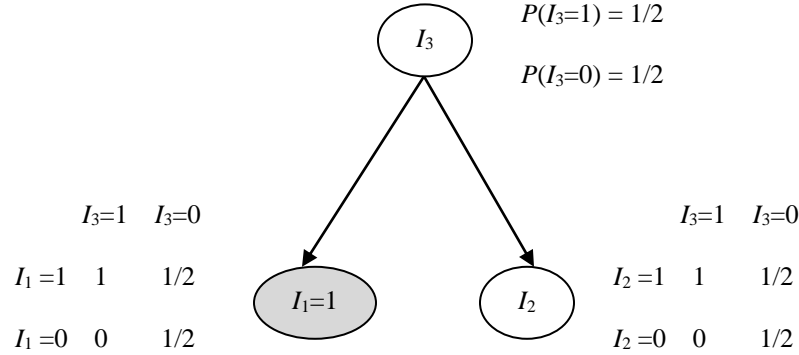


Figure 3. Target Bayesian network with evidence $I_1 = 1$

Let $PoP(I_2)$ and $PoP(I_3)$ be the posterior probabilities of nodes I_2 and I_3 , respectively and let $g(I_1, I_2, I_3) = P(I_3)P(I_1|I_3)P(I_2|I_3)$ be the joint probability function of target BN, we have:

$$PoP(I_2) = P(I_2 = 1 | I_1 = 1) = \frac{\sum_{I_3} g(I_1 = 1, I_2 = 1, I_3)}{\sum_{I_1, I_3} g(I_1, I_2 = 1, I_3)}$$

$$PoP(I_3) = P(I_3 = 1 | I_1 = 1) = \frac{\sum_{I_2} g(I_1 = 1, I_2, I_3 = 1)}{\sum_{I_1, I_2} g(I_1, I_2, I_3 = 1)}$$

Expending $PoP(I_2)$ and $PoP(I_3)$, we have:

$$PoP(I_2) = \frac{g(I_1 = 1, I_2 = 1, I_3 = 1) + g(I_1 = 1, I_2 = 1, I_3 = 0)}{g(I_1 = 1, I_2 = 1, I_3 = 1) + g(I_1 = 1, I_2 = 1, I_3 = 0) + g(I_1 = 0, I_2 = 1, I_3 = 1) + g(I_1 = 0, I_2 = 1, I_3 = 0)}$$

$$PoP(I_3) = \frac{g(I_1 = 1, I_2 = 1, I_3 = 1) + g(I_1 = 1, I_2 = 0, I_3 = 1)}{g(I_1 = 1, I_2 = 1, I_3 = 1) + g(I_1 = 1, I_2 = 0, I_3 = 1) + g(I_1 = 0, I_2 = 1, I_3 = 1) + g(I_1 = 0, I_2 = 0, I_3 = 1)}$$

It is easy to calculate $g(I_1, I_2, I_3) = P(I_3)P(I_1/I_3)P(I_2/I_3)$ by applying the CPT of target BN when variables I_1 , I_2 and I_3 are specified by concrete values. So posterior probabilities $PoP(I_2)$ and $PoP(I_3)$ are totally determined:

$$PoP(I_2) = \frac{\frac{1}{2} + \frac{1}{8}}{\frac{1}{2} + \frac{1}{8} + 0 + \frac{1}{8}} = 0.83$$

$$PoP(I_3) = \frac{\frac{1}{2} + 0}{\frac{1}{2} + 0 + 0 + 0} = 1$$

Because $PoP(I_3)$ is maximal posterior probability, item 3 is strongly recommended to user. The target BN is very small with only three nodes and so the complexity of computation is insignificant and does not affect ability of real-time response of recommendation system but when BN gets huge, it becomes serious problem that needs to be resolved. The Pearl algorithm (Neapolitan, 2003, pp. 126-156) is the classical method to solve this problem by propagating messages over the network according to two different directions. This research applies an inference algorithm – the variable elimination method of propagation built in Elvira system (Serafin, Carmelo, Pedro, & Francisco, 2003) into calculating posterior probabilities.

Besides, next section will mention an enhancement technique which alleviates such complexity of computation.

3. An enhancement – clustered Bayesian networks

As discussed, however the number of items is limited and not increased as much as the number of users, it is still so large. Obviously, BN consists of connected nodes but it may contain some incoherent or unconnected nodes because it is learned from large data. Such incoherent nodes make the inference mechanism less efficient. This is problem of incoherence among item nodes, which need solved.

Suppose that there are a lot of items in supermarket and they are divided into categories (groups). Clothes items (T-shirts, trousers, jeans, pulls, etc.) in the same category (clothes category) are related together. So they are connected nodes in BN and compose naturally a sub-group of nodes. Nevertheless, other items not related to clothes category will become incoherent nodes which are unconnected to clothes nodes. The inference based on whole BN including incoherent groups of nodes is less precise. This issue is solved by learning one BN for each category, thus, we build up many individual BN (s) and each BN represents a group of related items. In other words, a large and whole BN is decomposed into small and individual BN (s) so that nodes in the same individual BN are more coherent. For example, three individual BN (s) are built up, which correspond with three categories in supermarket: clothes, furniture and electrical goods.

In case that the training data set (rating matrix) doesn't specify explicitly categories, we will apply clustering technique into discovering groups of items. So the improvement in building up BN includes two steps:

1. Applying clustering methods such as k -mean, k -medoid, etc. into grouping items. We can classify items into groups (categories) manually, thus each item can belong to more than one group.
2. For each group (category) of items:
 - a. Training data set is pruned. Namely, for each row of rating matrix, columns which are not corresponding to items in this group are removed.
 - b. BN is learned from pruned training dataset. Such BN is called individual BN.

Note that a node can belong to more than one individual BN. It is a drawback but occurs in commercial context, an item can be classified into more than one category (group).

So every time recommendation task is required (in step 4 of our method), the inference process is executed on individual BN instead of whole BN as before. The speed is improved because the number of nodes in individual BN is much smaller than in whole large BN. But another issue is raised “given active user how to choose a right individual BN in order to perform inference task?”. If we browse over all of individual BN (s) and all their nodes to find out the right BN which contains most items (nodes) of active user, it consumes a lot time and computer resources. So it requires another approach. This is an open research but I also suggest a solution so-called mapping table (MT) technique.

The basic idea of MT technique is to create a mapping table (MT) at the same time to learning BN. Each row of MT is a key-value pair. Key is node’s name. Value is the bit set indicating which individual BN (s) to which this node belongs. Each bit of this bit set represents the occurrence of an individual BN, in other words, whether or not such individual BN contains the node specified by the key. Suppose there are 3 individual BN (s) such as BN_0 , BN_1 , BN_2 and 6 nodes such as A, B, C, D, E, F, G, H . The example MT is described in table 5.

A	100
B	100
C	010
D	010
E	001
F	001
G	011
H	011

Table 5. Mapping table

This MT is interpreted as below:

- Nodes A and B belong to BN_0 .
- Nodes C and D belong to BN_1 .
- Nodes E and F belong to BN_2 .
- Nodes G and H belong to BN_1 and BN_2 , respectively.

Given active user and her/his rated items, for each individual BN, the total number of nodes contained in this BN is counted. Which individual BN has the highest total number is chosen as right one on which inference task will be executed. For instance, given nodes E, F, G and H on which an active user rates, we have:

- The total number of nodes contained in BN_0 is 0, $t_0 = 0$ because BN_0 doesn’t any node rated by active user.
- The total number of nodes contained in BN_1 is 0, $t_1 = 2$ because BN_1 contains G and H .
- The total number of nodes contained in BN_2 is 0, $t_2 = 4$ because BN_2 contains E, H, G and H .

Because t_4 is maximal, BN_2 is the right individual BN.

4. Evaluation

Database *Movielens* (GroupLens, 1998) including 100,000 ratings of 943 users on 1682 movies is used for evaluation. Database is divided into 5 folders, each folder includes training set over 80% whole database and testing set over 20% whole database. Training set and testing set in the same folder are disjoint sets.

The system setting includes: processor Pentium(R) Dual-Core CPU E5700 @ 3.00GHz, RAM 2GB, available RAM 1GB, Microsoft Windows 7 Ultimate 2009 32-bit, Java 7 HotSpot (TM) Client VM. The proposed *BN method* is compared to four other methods: *Green Fall* – model-based CF using mining frequent itemsets technique, neighbor item-based method, neighbor user-based and SVD (Ricci, Rokach, Shapira, & Kantor, 2011, pp. 151-152). Note that the BN method is enhanced by clustering individual BN (s) aforementioned in section 3.

There are 7 metrics (Herlocker, Konstan, Terveen, & Riedl, 2004, pp. 19-39) used in this evaluation: *MAE*, *MSE*, *precision*, *recall*, *F1*, *ARHR* and *time*. Time metric is calculated in seconds. MAE and MSE are predictive accuracy metrics that measure how close predicted value is to rating value. The less MAE and MSE are, the high accuracy is. Precision, recall and F1 are quality metrics that measure the quality of recommendation list – how much the recommendation list reflects user's preferences. ARHR is also quality metric that indicates how well recommendation list is matched to user's rating list according to rating ordering. The large quality metric is, the better algorithm is. Note that these metrics take many arithmetic operations. For example, mean absolute error (MAE) is (Herlocker, Konstan, Terveen, & Riedl, 2004, p. 20):

$$MAE = \frac{1}{n} \sum_{i=1}^n |e_i - v_i|$$

Where n is the total number of recommended items while e_i and v_i are estimated value and rating value of item i , respectively.

If the BN representing purchase pattern is binary BN whose nodes are binary variables because binary BN provides more effective inference mechanism then, there is a back transformation. Binary values of recommended items resulted from binary BN are turned back normal values e_i ranging in integer interval $\{1, 2, 3, 4, 5\}$. These values e_i , in turn, are used to calculate evaluation metrics. However, the experiment is evaluated on multivalued BN whose nodes get values in integer interval $\{1, 2, 3, 4, 5\}$.

The evaluation result is shown in table 6 as follows:

	BN method	Green Fall	Item-based	User-based	SVD
MAE	0.6127	0.7241	0.5222	0.9319	0.5363
MSE	0.9023	1.1640	0.6675	2.1664	1.1734
Precision	0.1430	0.1328	0.0245	0.0014	0.0041
Recall	0.0552	0.0523	0.0092	0.0005	0.0015
F1	0.0785	0.0739	0.0131	0.0008	0.0021
ARHR	0.0504	0.0442	0.0043	0.0005	0.0016
Time	1.8780	0.0050	9.3706	8.3831	0.0176

Table 6. Evaluation result

The propose BN method is much more effective than other methods when getting high quality via metrics precision, recall, F1 and ARHR. Its accuracy is approximate to item-based, user-based methods, SVD and better than Green Fall via metrics MAE and MSE. It consumes more time than Green Fall and SVD but less time than item-based and user-based.

5. Conclusion

The essence of our method is to build up a Bayesian network (BN) from rating matrix and to apply such BN inference into recommending items to user. BN is directed acyclic graph (DAG) comprising a set of node and a set of arcs. Each node represents an item and each arc expresses a conditional dependency relationship between two items. Whenever recommendation task is required, the Bayesian inference is executed based on evidences from rating matrix and therefore, items whose posterior probability is highest are recommended to users. Because the recommendation accuracy relies on the adequacy of the BN structure, there are two ways to get best structure:

- Choosing appropriate learning algorithm. The research applies the learning algorithm built in the Elvira system (Serafín, Carmelo, Pedro, & Francisco, 2003) into constructing BN.
- Filling in missing values by estimated values so that the incomplete rating matrix (sparse matrix) becomes complete rating matrix. The BN learned from complete data gets more adequate.

The average and EM techniques are proposed to complete sparse matrix. The average method is faster and simpler but the EM method is more precise.

In the evaluation, our method is compared with other memory-based and model-based methods such as Green Fall, item-based, user-based and SVD. The result shows that BN based algorithm is the most effective method when it gains high quality via precision, recall, F1 and ARHR metrics although Green Fall is also dominant method in real-time response aspect. However the time to model training data in this method is much longer than in Green Fall method. In other words, building up BN takes much more time than mining frequent itemsets. In general BN based algorithm is good in recommendation quality and Green Fall is good in of real-time response.

Acknowledgment

This research is the place to acknowledge Sir Vu, Dong N. who gave me valuable comments and advices. These comments help me to improve this research.

Reference

- Campos, L. M., Fernández-Luna, J. M., Huete, J. F., & Rueda-Morales, M. A. (2010, September). Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks. (T. Denoeux, Ed.) *International Journal of Approximate Reasoning*, 51(7), 785–799. doi:10.1016/j.ijar.2010.04.001
- GroupLens. (1998, April 22). *MovieLens datasets*. (GroupLens Research Project, University of Minnesota, USA) Retrieved August 3, 2012, from GroupLens Research website: <http://grouplens.org/datasets/movielens/>
- Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems (TOIS)*, 22(1), 5-53.
- Langseth, H. (2009). Bayesian Networks for Collaborative Filtering. In A. Kofod-Petersen, H. Langseth, & O. E. Gundersen (Eds.), *Norwegian Artificial Intelligens Symposium (NAIS)* (pp. 67-78). Tapir Akademisk Forlag. Retrieved August 14, 2015, from www.tapironline.no/last-ned/248
- Miyahara, K., & Pazzani, M. J. (2000). Collaborative Filtering with the Simple Bayesian Classifier. In R. Mizoguchi, J. Slaney, R. Mizoguchi, & J. Slaney (Eds.), *PRICAI 2000 Topics in Artificial Intelligence* (pp. 679-689). Springer Berlin Heidelberg. doi:10.1007/3-540-44533-1_68
- Neapolitan, R. E. (2003). *Learning Bayesian Networks*. Upper Saddle River, New Jersey, USA: Prentice Hall.

- Ricci, F., Rokach, L., Shapira, B., & Kantor, P. B. (2011). *Recommender Systems Handbook* (Vol. I). (F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor, Eds.) Springer New York Dordrecht Heidelberg London.
- Serafin, C. M., Carmelo, R. T., Pedro, M. L., & Francisco, V. J. (2003, January). Elvira system. *Elvira project, 0.11*. Spain: National University of Distance Education. Retrieved 2012, from <http://www.ia.uned.es/~elvira>
- Su, X., & Khoshgoftaar, T. M. (2009). A Survey of Collaborative Filtering Techniques. (J. Hong, Ed.) *Advances in Artificial Intelligence, 2009*.