A Hybrid Recommender System Based on Bayesian Network Model

Yaonan Zhong, Xiaowei Zhang, Yihao Du
School of Electrical Engineering and Computer Science
Winter 2015 CS536 Final Project

Abstract

In this project, we used Bayesian network model to deal with the recommendation problem combining content-based and collaborative information. An efficient weighted sum mechanism is used to estimate the conditional probability distribution. The model performance is demonstrated using the MovieLens dataset.

1 Introduction

With widely spreading of Internet, great number of web service has been involved in people's life. In order to promote this web service, recommender system is embedded in this web. This system helps the user to easily access information, which is most relevant to their interest. In this way, the user is not necessary to spend great of time in large space. Nowadays, recommender system has be used in a great community of online content, including movies [1], book [2], news [3], friends [4], and jobs [5].

This system greatly enhances the user's experience in the web service. This convenience provided by recommender system will increase the frequency of web service's usage, which is the goal of the majority of companies. Nowadays, more and more companies are aware of the importance of the recommender system, which has been incorporated in their web page. For example, Facebook recommends some other people sharing the same friends with the user. Google recommends some other search words relevant to the original one provided by the user. Amazon recommends some other products sharing the same function of the item in the users' shopping history.

The key problem in the recommender system is how to select the correct item for the users. The algorithm and its implementation, outputting the correct selection for the user, are the key components in this system. Netflix [6], a widely unknown provider of Internet stream media, has supplied 1 Million dollars for a team, who improves the accuracy of predictions about the enjoyable movie. As is described in Campos' paper [7], the rating system in the webpage is a great tool to solve this problem. The recommended items for each user are some products with high rate from this user. It means the recommender system can be established based on the inference for the rate of the products.

2 Related Works

As is mentioned in last section, the key component in this recommender system is to develop an efficient algorithm to select several items highly rated by the users. Generally speaking, there are three kinds of filtering algorithm based on different information.

The first algorithm is content-based algorithm. The recommender system with this algorithm has stored content information of both items and users. The items with great similarity of user's preference, which is indicated by this user's rating history, are recommended to him. In the idea of probabilistic, there is high probability of the item fitting user's preference, to obtain high rate from this user. One important technique to implement this algorithm is to create a probabilistic model for a user based on his rating history. The system can use this model to predict the rate of new item for this user, and recommend an item with possible high rate for him [8].

The second algorithm is collaborative-filtering algorithm. This foundation of this algorithm is information of a group of people who share the same preference with a specific user. The item fitting preference of the community will be recommended to this user. In this algorithm, there are two computational approaches in the collaborative-filtering algorithm [9]: neighborhood methods and latent factor models. The neighborhood method involves user-based computation method or items-based computation method. User-based neighborhood method means to directly predict the score of an item based on the score of user's community. Item-based neighborhood method, also be called item-to-item method [10], is an indirect prediction method. When the rate of one item for a specific user needs to be predicted, the computation is performed based on the corresponding user group's historical rate of other items similar to the target item. Latent factor models perform the inference in terms of the factors from the rating history. For example, in order to predict the score of a book, the factor about category (e.g. scientific VS fiction) may needs to be taken into consideration.

Now, we can make comparison between the two methods. The content-based algorithm aims at establishing a joint distribution of all items, and then uses it to predict rate for the new items. The benefit of this algorithm is unique characterization in user's recommendation, for foundation information of this recommendation is preference information of the user himself. The limitation in content-based algorithm is hardly applicable for the new users, whose information is not great enough for the system to provide this recommendation.

The collaborative-filtering algorithm aims at establishing some conditional models, which is used to predict the score based on the score of the same item or the score of the similar item from other users. For this algorithm, information is great enough for our prediction, but there are still some shortcomings in the output of this algorithm. There is great possibility that some popular items are recommended to the user, but this user is not interested in this item.

A common idea is that greater performance exists in the collaborative-filtering recommender system, for it relies on sufficient data in the rating system. Based on the weakness of this system, the content-based algorithm can be used as an alternative method in the system. In this way, the 3rd algorithm is generated, hybrid approach. This approach takes the advantage of both systems. This is the reason we applies the hybrid recommender system in this project. More details of the hybrid system will be discussed in the next section.

3 Methodologies

In this section we will describe the BN used to represent the hybrid system. This model represents how users, items, and features are related. Focusing on the input data, the content description of the items is usually expressed by means of a sparse binary matrix. Similarly, the ratings are also represented by means of a matrix, where users are represented in the rows and items in the columns. This matrix is usually sparse as users usually rate a low number of items. We denote by R the rating's domain. When a user has not rated a product, the value is 0.

3.1 Elements in a Recommender Context

Since the BN-based system should include information about users $\mathcal{U}$, items $\mathcal{I}$ and features $\mathcal{F}$, we are going to consider the domain of these variables:

- Feature nodes: There will be an attribute node F_k for each feature used to describe a product. Each node has an associated binary random variable, which takes its values from the set $\{f_{k,0}, f_{k,1}\}$, which means that the kth feature is not relevant (not apply), $f_{k,0}$, or is relevant (apply), $f_{k,1}$, for the description of the content of a product.
- Item nodes: There is a node I_j, for each item. The random variable associated with I_j will take its values from the set $\{i_{j,0}, i_{j,1}\}$ meaning that the item is not relevant (not apply) or is relevant (apply), respectively, when it comes to predicting the user's rating.
- User nodes: These U_i nodes will be used to predict the rating for the target item, particularly they should represent how probable is ''the user rates with s an item". The domain of this variable is therefore the set $\mathcal{R} \cup \{0\}$. The additional value, 0, is included to model the lack of knowledge, i.e. the user has no useful information for predicting the target item's rating.

3.2 Topology of the Model

The user profile will be used to predict how the active user would rate an item. In the hybrid approach, we use a BN to represent both content and collaborative components [11]. Then, these components will be integrated in order to complete our hybrid system. We use a user variable A_{CB} gathering the information needed to perform content-based predictions. With respect to the collaborative component, we have to look for users similar to the active user (and therefore, a learning process becomes necessary). In this case, the collaborative information is also gathered in a user variable A_{CF}. To allow the combination [12] of both components we use a variable A_H to encode the active user's predictions at the hybrid level.

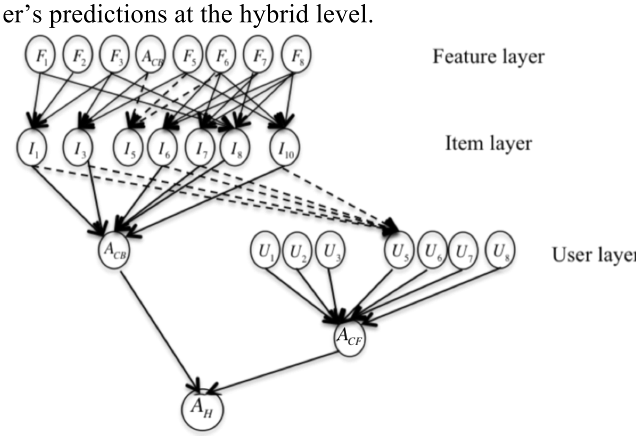

Figure 1 Sub-graph of the hybrid Bayesian network

3.2.1 Content-based Component

We will consider that an item's relevance will depend on the relevance values of the features that define it. Therefore, there will be an arc from each feature node, F_i, to the nodes representing those items, I_j, which have been described with this feature. By directing the links in this way, we allow two items with a common subset of features to be dependent (except when we know the relevance values of these common features). In order to conclude this part, we must connect the nodes representing the items with the node representing the active user's predictions. The basic rule for performing these connections is simple: for each item I_j rated by the active user, add the arc $I_j \rightarrow A_{CB}$ to the graph.

3.2.2 Collaborative-filtering Component

The collaborative component will comprise those people with similar tastes or preferences to the active user, represented by A_{CF}. These relations between users will depend on user ratings and so they must be learnt from the data.

3.2.3 Hybrid Component

Given an active user A, we will have his or her own preferences about the relevance of a new item in node A_{CB} (representing the content-based component), and also the preferences borrowed from similar users in node A_{CF} (collaborative component). These two preferences must be combined in order to obtain the final prediction for the user. This can be easily represented in the BN-based model by including a new node, A_H, which has both content (A_{CB}) and collaborative (A_{CF}) information as its parents.

3.3 Model Modification

Demographic information can be used to identify the types of users that like a certain object.

- Demographic [13] nodes: Similarly, there will be an attribute node D_k for each demographic feature used to describe a user. Each node has an associated binary random variable, which takes its values from the set $\{d_{k,0}, d_{k,1}\}$, which means that the kth demographic feature is not relevant (not apply), $d_{k,0}$, or is relevant (apply), $d_{k,1}$, for the description of the content of a product.

So for our model, instead of using similarity between the preferences of the active user in the CF component, we propose to combine similarity with the new added layer D, which contains the demographic information [14].

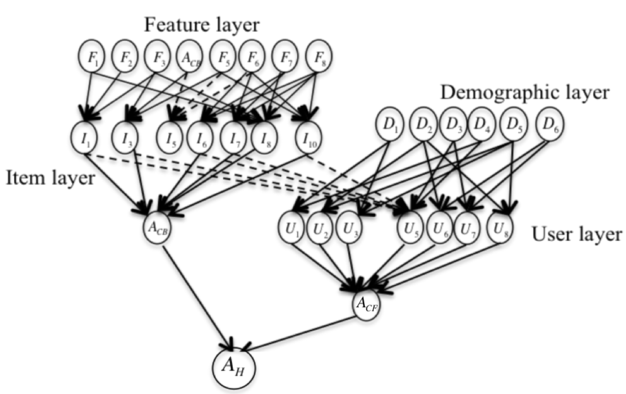

Figure 2 Sub-graph of our model

3.4 Implementation

In order to complete the model's specification, the numerical values for the conditional probabilities must be estimated from the data sets. We factorize the conditional probability tables into a set of weights and use an additive criterion to combine these values. Then we will use the canonical weighted sum to model item and user variables. At last our goal is to compute the predict rating for a given user.

3.4.1 Estimation of Conditional Probability Distribution

The conditional probabilities are represented using the canonical weighted sum (CWS) gate.

- Canonical Weighted Sum:

  Let X_i be a node in a BN, let $Pa\ (X_i)$ be the parent set of X_i, and let Y_k be the kth parent of X_i in the BN. By using a canonical weighted sum, the set of conditional probability distributions stored at node X_i are then represented by means of

$$\Pr(x_{i,j} \mid Pa(X_i)) = \sum_{Y_k \in Pa(X_i)} w(y_{k,l}, x_{i,j}) \qquad (1)$$

  where $y_{k,l}$ is the value that variable Y_k takes in the configuration $Pa(\ X_i)$, and $w(y_{k,l}, x_{i,j})$ are weights (effects) measuring how this lth value of variable Y_k describes the jth state of node X_i. The only restriction that we must impose is that the weights are a set of non-negative values verifying that for each configuration $Pa\ (X_i)$

$$\sum_{j=1}^{r} \sum_{Y_k \in Pa(X_i)} w(y_{k,l}, x_{i,j}) = 1 \qquad (2)$$

The CWS gate has its own limitations since a general probability distribution cannot be represented by means of this gate. It only can represent properly those situations where the joint distribution can be computed by adding the individual's weights. Nevertheless, we believe that its use is appropriate in the recommender framework.

3.4.2 Inference: Computing Recommendation

The aim of the inference process is to estimate the rating of the active user A, given the evidence $Pr\ (A = s|ev)$.

Theorem 1. Let X_a be a node in a BN network, let m_{xa} be the number of parents of X_a, Y_j be a node in $Pa\ (X_a)$, and l_{Yj} the number of states taken by Y_j. If the conditional probability distributions can be expressed under the conditions given by Eq. (2) and the evidence is only on the ancestors of X_a, then the exact posterior probabilities can be computed using the following formula:

$$\Pr(x_{a,s} \mid ev) = \sum_{j=1}^{m_{Xa}} \sum_{k=1}^{l_{Yj}} w(y_{j,k}, x_{a,s}) \bullet \Pr(y_{j,k} \mid ev) \qquad (3)$$

In the first case, on the content-based component, the evidence would comprise a set of features for an item (evidence in the first layer of the Bayesian network), propagation is carried out directly as explained in Theorem 1.

Theorem 2. Let F_k be a parent node of I_j in a Bayesian network, with the former being a root node in the network. The a posteriori probability of relevance of the feature given the variable I_j playing the role of evidence is then computed as follows:

$\Pr(f_{k,1} \mid i_{j,1}) =$

$$\begin{cases} \Pr(f_{k,1}) & \text{if } F_k \notin Pa(I_j) \\ \Pr(f_{k,1}) + \dfrac{w(f_{k,1}, i_{j,1}) \Pr(f_{k,1})(1-\Pr(f_{k,1}))}{\Pr(i_{j,1})} & \text{if } F_k \in Pa(I_j) \end{cases} \quad (4)$$

$$\Pr(d_{k,1} \mid u_{j,1}) =$$

$$\begin{cases} \Pr(d_{k,1}) & \text{if } D_k \notin Pa(U_j) \\ \Pr(d_{k,1}) + \dfrac{w(d_{k,1}, u_{j,1}) \Pr(d_{k,1})(1-\Pr(d_{k,1}))}{\Pr(u_{j,1})} & \text{if } D_k \in Pa(U_j) \end{cases} \quad (5)$$

In the second case (instantiating items), the evidence would comprise the item itself (in the second layer), the probability must be computed for each feature node linked to the target item. These posterior probabilities can then be incorporated into the propagation process.

1. Content-based propagation: – If $(ev_{cb} = I_j)$// Item instantiation set $Pr(i_{j,1}\|ev) = 1$ Compute $Pr\ (F_k\|ev)$ using Theorem 2 (4) else for each $F_k \in I_j$ set $Pr\ (F_k = 1\|ev) = 1$// – Propagate to items using Theorem 1 – Propagate to A_{CB} and $U_i \in \mathcal{U}_1^-$ using Theorem 1 2. Collaborative propagation: – For each $D_k \in U_j$ set $Pr\ (D_k = 1\|ev_{cf}) = 1$ – Propagate to users using Theorem 2 (5) – For each $U_k \in \mathcal{U}_1^+$ set $Pr\ (U_k = r_{k,j}\|ev_{cf}) = 1$ – Propagate to A_{CF} node using Theorem 1 3. Combine content-based and collaborative likelihoods at hybrid node A_H 4. Select the predicted rating

Table 1 Algorithm to compute $Pr\ (H_a|ev_{cb} \cup ev_{cf})$

We denote those users who rated the target item I in the past as $\mathcal{U}_1^+$ and those who did not rate the target item I in the past as $\mathcal{U}_1^-$.

3.4.3 Estimation of Weights

I/F	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8		
I_1	1	1	1	0	0	0	0	0		
I_2	1	1	1	0	0	0	0	0		
I_3	0	0	1	1	1	0	0	0		
I_4	0	0	0	1	1	1	0	0		
I_5	0	0	0	0	1	1	0	0		
I_6	0	0	0	0	0	1	1	1		
U/I	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}
U_1	5	5	3	1	3	3	0	0	0	0
U_2	5	5	4	1	1	4	0	0	0	0
U_3	4	4	3	2	2	4	0	0	0	0
U_4	1	0	1	0	0	2	1	3	0	5

Table 2 Example of matrices

We will now present various methods for estimating these weights.

- F Feature Variables:

  Starting from the feature nodes (as these do not have parents) it is only necessary to compute the a priori probability distributions of relevance. In this paper, we use a different alternative for estimating these probabilities: RF (relative frequency), i.e.

  $$\Pr(f_{k,1}) = \frac{n_k + 0.5}{m+1}$$

  where l is the size of the set F; n_k the number of times that feature F_k has been used to describe an item, i.e. the column sum of the left-hand side of Table 2, and m the number of items. The value $Pr\ (f_{k,0})$ is obtained as $Pr\ (f_{k,0}) = 1 - Pr\ (f_{k,1})$.

- D Demographic Feature Variables:

  Similarly with the F feature variables, we have:

  $$\Pr(d_{k,l'}) = \frac{n_k^{'} + 0.5}{m^{'} + 1}$$

  where l' is the size of the set D; n_k' the number of times that feature D_k has been used to describe an user, and m' the number of users. The value $Pr\ (d_{k,0})$ is obtained as $Pr\ (d_{k,0}) = 1 - Pr\ (d_{k,1})$.

- I Item Variables:

  With respect to the item nodes, $I_j \in \mathcal{I}$, as these represent a binary variable, the only weights to be defined are those needed to compute $Pr\ (i_{j,1}|Pa\ (I_j))$, since $Pr\ (i_{j,0}|Pa\ (I_j)) = 1 - Pr\ (i_{j,1}|Pa\ (I_j))$.

  In order to assess these values we will consider the following idea: Assume that F_1 and F_2 are two features describing an item I_j, with F_1 being a common feature (in the sense that it has been used to describe many items) and F_2 a rare feature (it appears in few items). It is natural to think that when both features are relevant ($F_1 = f_{1,1}$ and $F_2 = f_{2,1}$) the contribution of F_2 on the I_j's relevance degree will be greater than the contribution of F_1. Particularly, the concept of inverted document frequency (*idf*) [15] is used to measure the term's importance. Therefore, using an *idf*-based approach we use the expression $log\ ((m/n_k)\ +\ 1)$ to measure the importance of a feature in the entire database. Obviously, when a feature is not relevant its weight is set to zero. Therefore, the weights will be computed as

  $$w(f_{k,1}, i_{j,1}) = \frac{1}{M(I_j)} \log((\frac{m}{n_k}) + 1),\ w(f_{k,0}, i_{j,1}) = 0 \quad (6)$$

  with $M(I_j)$ being a normalizing factor computed as

  $$M(I_j) = \sum_{F_k \in Pa(I_j)} \log((\frac{m}{n_k}) + 1) \quad (7)$$

  For example, consider the item I_6 in Table 2 we have that

  $$w(f_{6,1}, i_{6,1}) = 0.3, w(f_{7,1}, i_{6,1}) = 0.4,\quad w(f_{8,1}, i_{6,1}) = 0.3,$$

  thus using the CWS we have that

  $$\Pr(i_{6,1} \mid f_{6,1}, f_{7,1}, f_{8,1}) = 1, \Pr(i_{6,1} \mid f_{6,0}, f_{7,1}, f_{8,1}) = 0.7$$

  $$\Pr(i_{6,1} \mid f_{6,0}, f_{7,1}, f_{8,0}) = 0.4$$

- U User Variables:

  Here we have to distinguish those variables representing the content-based predictions (having items as their parents), the variable that combines collaborative information (having users as their parents).

Content-based predictions: In this case, we must consider the influence of an item in the rating pattern of the user. To assess these weights we will consider two criteria: Firstly, for a given user $U_{CB} \in \{A_{CB}\} \cup \mathcal{U}_1^-$, whenever he or she rated an item I_k with the value s, then all the probability mass should be assigned to the same rating s at the user level. On the other hand, we will assume that all the items are equally important for predicting the active user's rating. Thus, taking into account these two ideas, and depending on whether the item I_k appears as relevant or not in the configuration $Pa\ (U_{CB})$, these weights might be defined as follows:

$$\begin{aligned} &w(i_{k,1}, u_{cb,s}) = \frac{1}{2\,|\,I(U_{CB})\,|}, \\ &w(i_{k,1}, u_{cb,t}) = 0, \quad \text{if } t \neq s, 0 \leq t \leq \#r \\ &w(i_{k,0}, u_{cb,0}) = \frac{1}{2\,|\,I(U_{CB})\,|}, \\ &w(i_{k,0}, u_{cb,t}) = 0, \;\; 0 \leq t \leq \#r, \end{aligned} \tag{8}$$

For example,

$w(i_{6,1}, u_{4,2}) = 0.166, w(i_{7,1}, u_{4,1}) = 0.166$,

$\Pr(u_{4,1} \,|\, Pa(U_4)) = 0.166 + 0.166 + 0.166 = 0.5$

A_{CF} Collaborative-based predictions: In this case, we must determine the weights reflecting the contribution of each similarity [16] and demographic D_i in the prediction of the rating for the active user A.

For the similarity part, we are measuring how the set of rating overlaps:

$$sim(A,U) = abs(PC(A,U)) \times \frac{|\,I(A) \cap I(U)\,|}{|\,I(A)\,|}$$

where I is the set of items rated by the user in the data set.

The particular weights are:

$$\begin{aligned} &w(u_{i,t}, a_{cf,s}) = RSim(U_i, A) \times \Pr^*(A = s \,|\, U_i = t), 1 \leq t, s \leq \#r \\ &w(u_{i,t}, a_{cf,0}) = 0, \quad \text{if } 1 \leq t \leq \#r \\ &w(u_{i,0}, a_{cf,0}) = RSim(U_i, A) \\ &w(u_{i,0}, a_{cf,s}) = 0, \;\; 1 \leq s \leq \#r \end{aligned} \tag{9}$$

where

$RSim(U_i, A) = sim(U_i, A) / \sum_{j \in Pa(A_{CF})} sim(U_j, A)$

And for the demographic part, we have the similar formula with content-based predictions, so the weights are:

$$\begin{aligned} &w(d_{k,1}, u_{cf,s}) = \frac{1}{2\,|\,D(U_{CF})\,|}, \\ &w(d_{k,1}, u_{cf,t}) = 0, \quad \text{if } t \neq s, 0 \leq t \leq \#r \\ &w(d_{k,0}, u_{cf,0}) = \frac{1}{2\,|\,D(U_{CF})\,|}, \\ &w(d_{k,0}, u_{cf,t}) = 0, \;\; 0 \leq t \leq \#r, \end{aligned} \tag{10}$$

- A_H User Variables:

  As AH node has two parents, A_{CB} and A_{CF}, representing the content-based and collaborative predictions, we must assess the conditional probability values $Pr\ (A_H|\ A_{CB}\ , A_{CF})$. These probabilities represent how to combine both types of information when predicting the active user's rating for the item I_j. We use a parameter $\alpha_j = Pr(A_{CF} = 0|ev)^2$, $0 \leq \alpha_j \leq 1$, be used to control the contributions of each component.

$$\begin{aligned} &\Pr(a_{h,s} \,|\, a_{cb,s}, a_{cf,s}) = 1 \\ &\Pr(a_{h,s} \,|\, a_{cb,s}, a_{cf,q}) = \alpha_j, \quad \text{if } q \neq s \\ &\Pr(a_{h,s} \,|\, a_{cb,t}, a_{cf,s}) = 1 - \alpha_j, \quad \text{if } t \neq s \\ &\Pr(a_{h,s} \,|\, a_{cb,t}, a_{cf,q}) = 0 \end{aligned} \tag{11}$$

4 Results

This section presents the experimental results about the performance of the hybrid system. We used the MovieLens dataset collected by the GroupLens Research Project at the University of Minnesota during the seven-month period from September 19th, 1997 through April 22nd, 1998. This data set consists of 100,000 ratings (1-5) from 943 users on 1682 movies. Each user has rated at least 20 movies. Each movie is described by a set of genres. We used 5-fold cross validation to perform parameters training and testing.

For the feature layer in the model, we used the relative frequency as the priori probability distribution of relevance. We use $\alpha_j = \Pr\ (A_{cf} = 0|ev)$ as proposed in the original paper. In order to explore the effect of neighborhood in collaborative component, we consider top-N most similar users, where *N* is 10, 20, 30 and 50.

We selected the most probable rating as the predictive rating from the candidate distribution. In order to reflect the predictive capacity of the model, we used two approaches to measure the system accuracy: the mean absolute error (MAE) and the error rate. MAE measures how close system predictions are to the user's rating for each movie by considering the average absolute deviation between a predicted rating and the user's true rating.

$$MAE = \frac{\sum_{i=1}^{N} abs(p_i - r_i)}{N} \tag{12}$$

where *N* being the number of cases in the test set, *pi* the vote predicted for a movie and *ri* the true rating. The other approach, error rate, shows the capability of exact prediction.

# of neighbors	10	20	30	50
MAE	0.83133	0.83337	0.83524	0.83896
Error rate	0.59982	0.59996	0.60019	0.60089

Table 3 Hybrid performances without demographics layer

Stand-alone content-based performance:
MAE = 0.85222, Error rate = 0.61076

Hybrid performance with demographics layer and 10 neighbors:
MAE = 0.8446, Error rate = 0.5991

5 Discussions

In last section, we show the experimental results given by the hybrid model. Table 3 presents the prediction capacity using different number of similar neighbors in the collaborative component without the demographics layer. We can observer that using a small number of neighbors gives us higher prediction accuracy. It is possible that a large number of neighbors can introduce some noise such as irrelevant information that decreases the model performance. Second, comparing the Table 3 and the stand-alone content-based results, we observer that the hybrid model indeed has a better performance than the single content-based component as expected.
Considering the new model involving the demographics information, such as age, gender and occupation, we can observe that it didn't give us a better prediction result than the original hybrid model. Since the belief propagates from the demographics layer to the user layer, it directly affects the content-based rating distribution. This additional layer is an exploratory experiment and we are not sure how this layer leads to a worse result. The reason for adding this layer is based on our intuition that it may provide some hidden information that is related to the behavior of user rating.

6 Conclusions

In this paper, we implemented a hybrid recommender model based on Bayesian network. The network structure combines all the elements in the recommendation problem. We used weighted sum to approximate the conditional probability distribution in parameter learning. The model includes content-based and collaborative components. The experimental results show us how these two components help improve the prediction accuracy. We also explored an additional layer using the demographics information of users and discussed its effect on the model performance.

References

[1]. IMDb. http://www.imdb.com

[2]. Amazon. http://www.amazon.com

[3]. Google news. http://news.google.com

[4]. Facebook. https://www.facebook.com

[5]. Indeed. http://www.indeed.com

[6]. NetflixPrize. http://www.netflixprize.com

[7]. L. Campos, J. Fernandez-Luna, J. Huete, M. Reuda-Mrales. 2010. Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks.

[8]. R.J.Mooney, L.Roy. 2000. Content-based book recommending using learning for text categorization, in: DL'00: Proceedings of the Fifth ACM Conference on Digital Libraries. 195–204. ACM Press.

[9]. Mike Gartrell, Ulrich Paquet, Ralf Herbrich. May 2012. A Bayesian Treatment of Social Links in Recommender Systems, Technical Report, CU-CS-1092-12.

[10]. Linden, G; Smith, B; York, J. January 2003. "Amazon.com recommendations: item-to-item collaborative filtering". IEEE Internet Computing (IEEE) 7 (1): 76–80.

*[11]. Luis M. de Campos, Juan M. Fernández-Luna *, Juan F. Huete, Miguel A. Rueda-Morales. 2010. Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks. International Journal of Approximate Reasoning 51 (2010) 785–799.*

[12]. R. Burke. 2002. Hybrid recommender systems: survey and experiments, User Modeling and User-Adapted Interaction 12 (4) 331–370.

[13]. Marco de Gemmis, Leo Iaquinta, Pasquale Lops, Cataldo Musto, Fedelucio Narducci, and Giovanni Semeraro. 2006. Department of Computer Science University of Bari "Aldo Moro", Italy.

[14]. Michael J. Pazzani. A Framework for Collaborative, Content-Based and Demographic Filtering. Department of Information and Computer Science University of California, Irvine, CA

[15]. C.J.V. Rijsbergen, 1979. Information Retrieval, second ed., Butterworth, London, UK.

[16]. L.M. de Campos, J.M. Fernández-Luna, J.F. Huete, 2008. A collaborative recommender system based on probabilistic inference from fuzzy observations, Fuzzy Sets and Systems 159 (12) 1554–1576.