**A New Approach for Collaborative Filtering**

**based on Bayesian Network Inference**

Loc Nguyen

Huong Duong Company, Ho Chi Minh city, Vietnam

**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.

We 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 we also suggest two methods such as average and expectation maximization (EM) so as to fill in missing values.

**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) [1]:

* 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 items 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 one 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. It is the potential approach because it opens a new point of view about recommendation domain. In section 2 we 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.

|  |  |  |  |
| --- | --- | --- | --- |
|  | *item1* | *item2* | *item3* |
| *user1* | r11 = 1 | r12 = 2 | r13 = 1 |
| *user2* | r21 = 2 | r22 = 1 | r23 = 2 |
| *user3* | r31 = 4 | r32 = 1 | r33 = 5 |
| *user4* | r41 = 1 | r42= 2 | r43 = ? (missing) |

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

Let  and  be the normal user vector *i* and the active user *a*, respectively where *rij* 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. We propose a new model-based CF algorithm based on Bayesian network [3] 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 sections.

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

|  |  |  |  |
| --- | --- | --- | --- |
|  | *item1* | *item2* | *item3* |
| *user1* | r11 = 1 | r12 = 3 | r13 = **?** |
| *user2* | r21 = 3 | r22 = **?** | r23 = 5 |
| *user3* | r31 = 4 | r32 = 2 | r33 = 1 |
| *user4* | r41 = **?** | r42 = **?** | r43 = 3 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *user1* | *user2* | *user3* | *user4* |
| *item1* | r11 = 1 | r21 = 3 | r31 = 4 | r41 = **?** |
| *item2* | r12 = 3 | r22 = **?** | r32 = 2 | r42 = **?** |
| *item3* | r13 = **?** | r23 = 5 | r33 = 1 | r43 = 3 |

**Table 2.** Transposinguser-based matrix to item-based matrix

User-based matrix is the original format of rating matrix. 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.

**2.2. Filling in missing values**

The BN learned from complete rating matrix is more adequate than the one learned from sparse matrix. 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, *r41* and *r42* are replaced as following:

* *r41 = r43 / 1 = 3*
* *r42 =* (*r41 + r43*) */ 2 =* (*3+3*) / 2 *= 3*.

Note that *r42* is computed based on the replaced value of *r41*; it is the overlapping. By other way, rating *r42* can be the same to *r41* which is the mean of user vector *user4*. 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 *user4* is empty, its mean is undefined but there is no interruption in replacement process. Therefore, *r42* is assigned by the mean of item-vector *item2*, so *r42 =* (*r12 + r32*) */ 2 = 2.5* and computation will continue. By this way, remaining missing values such as *r22*, *r13* are estimated.

* *r22 =* (*r21 + r23*) */ 2 =* (*3 + 5*) */ 2 = 4*
* *r13 =* (*r11 + r12*) */ 2 =* (*1 + 3*) */ 2 = 2*

Thus, we have completely estimated item-based rating matrix as below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *user1* | *user2* | *user3* | *user4* |
| *item1* | 1 | 3 | 4 | 3 |
| *item2* | 3 | 4 | 2 | 3 |
| *item3* | 2 | 5 | 1 | 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. Another way to fill in missing data is to apply statistical approaches such as expectation maximization (EM). EM algorithm is an efficient iterative procedure to compute the maximum likelihood (ML) estimate in the presence of missing or hidden data. Each iteration of EM algorithm consists of two processes: E-step and the M-step.

* In the expectation, or E-step: missing data is estimated given the observed data and current estimate of statistical parameters. Normally, statistical parameters are expectation and variance.
* In the maximization, or M-step: statistical parameter is estimated by maximizing the likelihood function under the assumption that missing data is known.

EM algorithm is more accurate than average technique because it applies statistical model into rating data so as to estimate missing values according to an iterative process, thus statistical parameters are improved after each iteration.

**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*. Each node in BN has five values {*1, 2, 3, 4, 5*} corresponding to user’s rating values: *5-*most favorite and *1-*most dislike. 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:

* Scored-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 whose each cell gets *1* (*0*) if its value is greater than or equal to *3* (and otherwise).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *user1* | *user2* | *user3* | *user4* |
| *item1* | 0 | 1 | 1 | 1 |
| *item2* | 1 | 1 | 0 | 1 |
| *item3* | 0 | 1 | 0 | 1 |

**Table 4.** Item-based binary rating matrix

Our BN is learned from binary matrix in table *4*. Let *I1, I2* and *I3* denote item *1*, item *2* and item *3*, respectively. Note that *I1, I2* and *I3* 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). The search space includes three Bayesian networks: *BN1, BN2* and *BN3* representing in following figure.

*BN3*

*BN2*

*BN1*

**Figure 1.** Bayesian networks search space: *BN1, BN2* and *BN3*

Let *D* denote binary rating matrix in table *4*. Now the CPT of every node is calculated based on following formulas:

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

* *P*(*I1 = 1*) *= 3/4, P*(*I1 = 0*) *= 1/4*
* *P*(*I2 = 1*) *= 3/4, P*(*I2 = 0*) *= 1/4*
* *P*(*I3 = 1*) *= 1/2, P*(*I3 = 0*) *= 1/2*
* *P*(*I2 = 1| I1 = 1*) *= 2/3, P*(*I2 = 0| I1 = 1*) *= 1/3, P*(*I2 = 1|I1 = 0*) *= 1, P*(*I2 = 0|I1 = 0*) *= 0*
* *P*(*I3 = 1| I1 = 1*) *= 2/3, P*(*I3 = 0| I1 = 1*) *= 1/3, P*(*I3 = 1|I1 = 0*) *= 0, P*(*I3 = 0|I1 = 0*) *= 1*
* *P*(*I1 = 1| I2 = 1*) *= 2/3, P*(*I1 = 0| I2 = 1*) *= 1/3, P*(*I1 = 1|I2 = 0*) *= 1, P*(*I1 = 0|I2 = 0*) *= 0*
* *P*(*I3 = 1| I2 = 1*) *= 2/3, P*(*I3 = 0| I2 = 1*) *= 1/3, P*(*I3 = 1|I2 = 0*) *= 0, P*(*I3 = 0|I2 = 0*) *= 1*
* *P*(*I1 = 1| I3 = 1*) *= 1, P*(*I1 = 0| I3 = 1*) *= 0, P*(*I1 = 1|I3 = 0*) *= 1/2, P*(*I1 = 0|I3 = 0*) *= 1/2*
* *P*(*I2 = 1| I3 = 1*) *= 1, P*(*I2 = 0| I3 = 1*) *= 0, P*(*I2 = 1|I3 = 0*) *= 1/2, P*(*I2 = 0|I3 = 0*) *= 1/2*

Let *g1, g2* and *g3* be joint probability functions of *BN1, BN2* and *BN3*, respectively. We have *g1*(*I1, I2, I3*) *= P*(*I1*)*P*(*I2|I1*)*P*(*I3|I1*) , *g2*(*I1, I2, I3*) *= P*(*I2*)*P*(*I1|I2*)*P*(*I3|I2*) and *g3*(*I1, I2, I3*) *= P*(*I3*)*P*(*I1|I3*)*P*(*I2|I3*). Let *δ1*, *δ2* and *δ3* be scoring criterions of *BN1*, *BN2* and *BN3*, respectively. These scoring criterions represent posterior probabilities of *g1, g2* and *g3* given *D* beingbinary rating matrix in table *4*. Applying Bayes’ theorem, the posterior probability of Bayesian network *BNi* given *D* is determined:

Suppose that prior probabilities *gi*(*I1, I2, I3*) (s) are the same and probability over training data *gi*(*D*) is constants, it is possible to remove *g*(*I1, I2, I3*) and *g*(*D*) from the posterior probability of Bayesian network *BNi*. No loss of generality, scoring criterion *δi* is defined as the likelihood function *gi*(*D|I1, I2, I3*)

If each column (corresponding to user) of *D* is considered as a *case* of training set, there are four cases: *c1 =* (*I1 = 0, I2 = 1, I3 = 0*), *c2 =* (*I1 = 1, I2 = 1, I3 = 1*), *c3 =* (*I1 = 1, I2 = 0, I3 = 0*) and *c4 =* (*I1 = 1, I2 = 1, I3 = 1*). Suppose cases are mutually independent, scoring criterion is product of posterior probabilities in flavor of cases.

For instance, by substituting the joint probability of *BN1*, namely *g1*(*I1, I2, I3*) *= P*(*I1*)*P*(*I2|I1*)*P*(*I3|I1*), into the formula of scoring criterion, we have:

Scoring criterions of *BN2* and *BN3* are calculated in the similar way.

It is easy to recognize that *δ3* = 0.0039 is the maximum score and so *BN3* is chosen as the best Bayesian network learned from item-based rating matrix.

*P*(*I3=1*) = 1/2

*P*(*I3=0*) = 1/2

*I3=1 I3=0*

*I2 =1* 1 1/2

*I2 =0* 0 1/2

*I3=1 I3=0*

*I1 =1* 1 1/2

*I1 =0* 0 1/2

**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 thatactive 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 *I1* becomes a binary evidence node whose value is *1*. Following is the target BN whose evidence is shaded.

*P*(*I3=1*) = 1/2

*P*(*I3=0*) = 1/2

*I3=1 I3=0*

*I2 =1* 1 1/2

*I2 =0* 0 1/2

*I3=1 I3=0*

*I1 =1* 1 1/2

*I1 =0* 0 1/2

**Figure 3.** Target Bayesian network with evidence *I1 = 1*

Let *PoP*(*I2*) and *PoP*(*I3*) be the posterior probabilities of nodes *I2* and *I3*, respectively and let g(*I1, I2, I3*) *= P*(*I3*)*P*(*I1|I3*)*P*(*I2|I3*) be the joint probability function of target BN, we have:

Expending *PoP*(*I2*) and *PoP*(*I3*), we have:

It is easy to calculated g(*I1, I2, I3*) *= P*(*I3*)*P*(*I1|I3*)*P*(*I2|I3*) by applying the CPT of target BN when variables *I1, I2* and *I3* arespecified by concrete values. So posterior probabilities *PoP*(*I2*) and *PoP*(*I3*) are totally determined:

Because *PoP*(*I3*) 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. Next section will mention enhancement technique which overcomes such complexity.

**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 such 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-means, K-centroid, 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:
   1. Training data set is pruned. Namely, for each row of rating matrix, columns which are not corresponding to items in this group are removed.
   2. 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 we 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 *BN0, BN1, BN2* and *6* nodes such as *A, B, C, D, E, F, G*, *H*. The example MT is described as following:

|  |  |
| --- | --- |
| 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 *BN0*.
* Nodes *C* and *D* belong to *BN1*.
* Nodes *E* and *F* belong to *BN2*.
* Nodes *G* and *H* belong to *BN1* and *BN2*, 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 *BN0* is *0*, *t0* = 0because *BN0* doesn’t any node rated by active user.
* The total number of nodes contained in *BN1* is *0*, *t1* = *2* because *BN1* contains *G* and *H*.
* The total number of nodes contained in *BN2* is *0*, *t2* = *4* because *BN2* contains *E, H, G* and *H*.

Because *t4* is maximal, *BN2* is the right individual BN.

**4. Evaluation**

Database *Movielens* [2] 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. Our CF 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.

There are 7 metrics used in this evaluation: *MAE, MSE, precision, recall, F1, ARHR* [11] and *time*. Note that 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. Following table is the evaluation result.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Our method** | **Green Fall** | **Item-based** | **User-based** | **SVD** |
| MAE | *0.6127* | *0.8529* | *0.5222* | *0.9319* | *0.5363* |
| MSE | *0.9023* | *1.6304* | *0.6675* | *2.1664* | *1.1734* |
| Precision | *0.1334* | *0.1058* | *0.0245* | *0.0014* | *0.0041* |
| Recall | *0.0513* | *0.0404* | *0.0092* | *0.0005* | *0.0015* |
| F1 | *0.0731* | *0.0576* | *0.0131* | *0.0008* | *0.0021* |
| ARHR | *0.0454* | *0.0341* | *0.0040* | *0.0003* | *0.0012* |
| Time | *1.8169* | *0.0050* | *8.5607* | *7.5558* | *0.0207* |

**Table 6.** Evaluation result

Our 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.

**6. 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 replies on the adequacy of the BN structure, there are two ways to get best structure:

* Choosing appropriate learning algorithm (not discussed in this research)
* 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 Ms. Phung, Do Thi Minh – University of Information Technology, Vietnam National University and Mr. Dong, Vu Ngoc who gave me valuable comments and advices. These comments help me to improve this research.

**Reference**

1. [Su, Khoshgoftaar 2009]. Xiaoyuan Su and Taghi M. Khoshgoftaar. A Survey of Collaborative Filtering Techniques. Hindawi Publishing Corporation, Advances in Artificial Intelligence, Volume 2009, Article ID 421425, 19 pages, doi:10.1155/2009/421425.
2. [Movielens dataset 2011]. Home page is http://www.movielens.org. Download dataset from http://www.grouplens.org/node/12.
3. [Richard 2003]. Richard E. Neapolitan. Learning Bayesian Networks. Northeastern Illinois University Chicago, Illinois 2003.
4. [Ungar, Foster 1998]. L. H. Ungar and D. P. Foster. Clustering Methods for Collaborative Filtering. Proceedings of the Workshop on Recommendation Systems, AAAI Press, 1998.
5. [Han, Kamber 2006]. Jiawei Han and Michelline Kamber. Data Mining: Concepts and Techniques. Second Edition. © 2006 by Elsevier Inc.
6. [Breese, Heckerman, Kadie 1998]. John S. Breese, David Heckerman and Carl Kadie. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. Technical Report MSR-TR-1998. Microsoft Research, Microsoft Corporation, One Microsoft Way, Redmon, WA 98052.
7. [Heckerman, Chickering, Meek, Rougnthwaite 2000]. David Heckerman, David Maxwell Chickering, Christopher Meek, Robert Rougnthwaite and Carl Kadie. Dependecy Networks for Inference, Collaborative Filtering and Data Visualization. Journal of Machine Learning Research 1 (2000) 49-75.
8. [Hofmann 2004]. Thomas Hofmann. Latent Semantic Models for Collaborative Filtering. ACM Transactionson Information Systems, Vol.22, No.1, January 2004, Pages 89-115.
9. [Shani, Heckerman, Brafman 2005]. G. Shani, D. Heckerman, and R. I. Brafman. An MDP-based Recommender System. Journal of Machine Learning Research, vol. 6, pp. 1265-1295, 2005.
10. [Rennie 2004]. Jason D. M. Rennie (jrennie@csail.mit.edu). A Short Tutorial on Using Expectation-Maximization with Mixture Models. MIT University, March 3, 2004
11. [Herlocker, Konstan, Terveen, Riedl 2004]. Jonathan. L. Herlocker, Joseph. A. Konstan, Loren G. Terveen, and John. T. Riedl. Evaluating Collaborative Filtering Recommender Systems. ACM Transactions on Information Systems, vol. 22, no. 1, pp. 5–53, 2004.

**Một cách tiếp cận mới cho lọc cộng tác dựa trên suy diễn mạng Bayesian**

Nguyễn Phước Lộc

Công ty TNHH MTV Lập trình Hướng Dương, Tp. Hồ Chí Minh city, Việt Nam

**Tóm tắt**

Lọc cộng tác (collaborative filtering – CF) là một trong những giải thuật khuyến nghị phổ biến, mỗi mặt hàng giới thiệu cho một người dùng dựa vào kết quả khảo sát các đánh giá từ cộng đồng nhiều người dùng. Có hai cách tiếp cận cho CF: nạp ức (memory-based) và mô thức (model-based).

Cách tiếp cận mô thức vượt trội hơn với khả năng đáp ứng thời gian thực do tận dụng cơ chế suy diễn cho tác vụ khuyến nghị. Tuy nhiên vấn đề dữ liệu thưa và dữ liệu không đầy đủ vẫn còn là nghiên cứu mở cần được cải tiến để cơ chế suy diễn ngày càng chính xác hơn với tốc độ cao hơn.

Chúng tôi đề xuất một giải thuật CF mô thức mới dựa trên mạng Bayesian khi mô hình suy diễn mạng Bayesian được xác nhận là ưu việt do cơ chế suy diễn dựa trên bằng chứng – rất thích hợp đối với dữ liệu đánh giá. Do chất lượng mạng Bayesian phụ thuộc vào sự đầy đủ của dữ liệu huấn luyện, chất lượng kém đi nếu dữ liệu chứa nhiều giá trị bị thiếu. Chúng tôi đề xuất hai phương thức để ước lượng giá trị bị thiếu là giải thuật trung bình và giải thuật cực đại hóa kỳ vọng (EM).

**1. Giới thiệu**

Hệ thống khuyến nghị (recommendation system) là hệ thống giới thiệu những mặt hàng mà người dùng có khả năng yêu thích nhất trong một cơ sở dữ liệu lớn gồm rất nhiều mặt hàng. Mặt hàng có thể là sản phẩm, dịch vụ, sách, tạp chí và tất cả những gì mà người dùng với vai trò khách hàng có thể mua. Những mặt hàng được giới thiệu đó thể hiện sở thích (interests) hay mối quan tâm của người dùng.

Hệ thống khuyến nghị được xây dựng theo hai hướng: lọc nội dung (content-based filtering – CBF) và lọc cộng tác (collaborative filtering – CF) [1]:

* CBF giới thiệu một mặt hàng với người dùng nếu mặt hàng này tương tự về nội dung với những mặt hàng mà họ đã yêu thích với ngữ nghĩa rằng mặt hàng yêu thích là mặt hàng có đánh giá cao. Nội dung của một mặt hàng là những thuộc tính của nó và tập nội dung tất cả các mặt hàng tạo thành ma trận, gọi là ma trận nội dung (content matrix).
* Với tiếp cận khác, CF giới thiệu một mặt hàng với người dùng nếu những láng giềng của họ cũng thích mặt hàng đó với định nghĩa láng giềng là những người giống nhau về sở thích. Đánh giá của một người dùng lên một mặt hàng thể hiện sở thích của họ đối với mặt hàng đó. Tất cả đánh giá người dùng trên các mặt hàng tạo thành ma trận, gọi là ma trận đánh giá (rating matrix).

Cả hai giải thuật CBF và CF đều có ưu nhược điểm riêng. CBF tập trung vào nội dung mặt hàng và sở thích cá nhân người dùng nên mục tiêu của nó là khuyến nghị những mặt hàng khác nhau cho những người dùng khác nhau. Điểm mạnh của CBF là khả năng đưa ra những khuyến nghị đặc thù nhưng CBF không hướng đến cộng đồng như CF. Khi đó những mặt hàng mà người dùng có thể thích bị ẩn trong cộng đồng, không thể phát hiện được bằng CBF. Đây là nhược điểm của CBF, một nhược điểm khác, CBF có thể dự đoán sai sở thích người dùng nếu số lượng người dùng quá lớn.

Nếu nội dung mỗi mặt hàng quá lớn với nhiều thuộc tính, giải thuật CBF tiêu thụ nhiều tài nguyên máy tính và do đó thời gian xử lý sẽ chậm; ngược lại tốc độ không thành vấn đề đối với CF trong trường hợp này do CF không tập trung vào các thuộc tính mặt hàng. Thay vào đó, CF chỉ quan tâm đến đánh giá của người dùng trên mỗi mặt hàng và đây chính là điểm mạnh của CF. Tuy nhiên, mặc dù không đối mặt với vấn đề dữ liệu nội dung các mặt hàng quá lớn, nhược điểm của CF cũng nảy sinh từ đây, bởi vì CF sẽ đưa ra những khuyến nghị ngoài mong đợi trong trường hợp mà một số mặt hàng có thể phù hợp với người dùng nhưng họ chưa đánh giá chúng. Đây là vấn đề dữ liệu thưa, nghĩa là ma trận đánh giá bị thiếu nhiều giá trị hay nói cách khác có quá nhiều mặt hàng chưa được đánh giá. Hai kỹ thuật hữu ích được đề xuất để khắc phục nhược điểm này của CF:

* Kết hợp CBF và CF: kỹ thuật này gồm hai giai đoạn, giai đoạn đầu áp dụng CBF để hoàn chỉnh ma trận đánh giá, giai đoạn sau áp dụng CF để thực hiện khuyến nghị. Kỹ thuật này hữu ích khi cải thiện độ chính xác của khuyến nghị nhưng tốn nhiều thời gian cho giai đoạn đầu đóng vai trò như bước tiền xử lý. Kỹ thuật dùng cả ma trận đánh giá và ma trận nội dung
* Mô hình hóa ma trận đánh giá như một mô hình suy diễn, ước lượng giá trị bị thiếu và thực hiện khuyến nghị dựa trên mô hình suy diễn này. Giải thuật CF theo cách tiếp cận mô hình này được gọi là CF mô thức (model-based CF), còn một cách tiếp cận khác là CF nạp ức (memory-based CF). CF mô thức áp dụng các phương pháp thống kê và máy học để khai thác ma trận đánh giá.

Mặc dù cách tiếp cận mô thức không cho kết quả chính xác như cách tiếp cận kết hợp nhưng nó giải quyết vấn đề dữ liệu khổng lồ và ma trận thưa; hơn nữa nó có thể đáp ứng yêu cầu người dùng một cách tức thời qua cơ chế suy diễn nhanh nhạy. Vì vậy bài báo này tập trung vào CF mô thức với suy diễn mạng Bayesian. Đây là cách tiếp cận đầy tiềm năng bởi vì nó mở ra một hướng mới trong lĩnh vực khuyến nghị. Phần hai trong bài báo mô tả ý tưởng chính của giải thuật CF mô thức dựa trên mạng Bayesian. Phần 3 là cải tiến cho giải thuật đề xuất. Phần 4 mô tả kết quả đánh giá. Phần 5 là kết luận.

Lưu ý trong bài báo các thuật ngữ như ma trận đánh giá (rating matrix), cơ sở dữ liệu đánh giá (rating database), tập huấn luyện (training set) có cùng nghĩa. Mỗi dòng và cột trong ma trận đánh giá tương ứng với người dùng và mặt hàng, mỗi ô là giá trị đánh giá của một người dùng lên một mặt hàng cụ thể. Vì thế mỗi dòng đại diện một vector người dùng hay vector đánh giá; các vector này được xem như hồ sơ người dùng. Người dùng đang yêu cầu những mặt hàng khuyến nghị được gọi là hoạt nhân. Vector tương ứng với hoạt nhân là hoạt vector.

|  |  |  |  |
| --- | --- | --- | --- |
|  | *item1* | *item2* | *item3* |
| *user1* | r11 = 1 | r12 = 2 | r13 = 1 |
| *user2* | r21 = 2 | r22 = 1 | r23 = 2 |
| *user3* | r31 = 4 | r32 = 1 | r33 = 5 |
| *user4* | r41 = 1 | r42= 2 | r43 = ? (bị thiếu) |

**Table 1.** Ma trận đánh giá (người dùng 4 là hoạt nhân)

Giả sử  và  tương ứng là vector người dùng *i* và hoạt vector *a*, với *rij* là giá trị đánh giá của người dùng *i* lên mặt hàng *j*. Dấu hỏi (?) biểu thị giá trị bị thiếu và mục tiêu của giải thuật CF là ước lượng các giá trị bị thiếu. Ma trận đánh giá trong ví dụ trên là ma trận người dùng (user-based matrix) với mỗi dòng là một vector người dùng; ngược lại ma trận mặt hàng (item-based matrix) chứa các vector mặt hàng.

**2. Một giải thuật CF mới dựa trên mạng Bayesian**

Ý tưởng chính của CF mô thức (model-based CF) là xác lập một mô hình suy diễn tối ưu có thể đáp ứng yêu cầu khuyến nghị theo thời gian thực. Bên cạnh đó, CF mô thức cũng giải quyết các vấn đề quan trọng như *ma trận thưa* và *cừu đen*. Chúng tôi đề nghị một giải thuật CF mô thức mới dựa trên suy diễn mạng Bayesian [3] nhằm đạt độ chính xác cao và giải quyết vấn đề ma trận thưa. Nhìn chung giải thuật đề xuất tập trung vào xây dựng mạng Bayesian từ ma trận đánh giá. Mỗi nút trong mạng đại diện một mặt hàng và mỗi cạnh biểu diễn mối quan hệ phụ thuộc giữa hai nút. Bất cứ khi nào yêu cầu khuyến nghị xảy ra, cơ chế suy diễn của mạng Bayesian sẽ quyết định mặt hàng nào cần tư vấn cho người dùng, dựa theo xác suất hậu nghiệm (posterior probabilities) của mặt hàng đó. Xác suất này càng lớn, khả năng người dùng mua mặt hàng càng cao và do đó, mặt hàng này có tần suất mua cao và nên khuyến nghị nó với người dùng. Do xây dựng mạng Bayesian từ ma trận đầy đủ (complete matrix) dễ dàng và hiệu quả hơn từ ma trận thưa, chúng tôi thay thế những giá trị bị thiếu (missing values) trong ma trận thưa bằng các giá trị ước lượng (estimated value) nhằm biến ma trận thưa thành ma trận đầy đủ. Kỹ thuật ước lượng giá trị bị thiếu, sẽ thảo luận sau. Giải thuật đề xuất gồm 4 bước:

1. Chuyển vị ma trận người dùng (user-based matrix) thành ma trận mặt hàng (item-based matrix).
2. Điền những giá trị bị thiếu trong ma trận mặt hàng.
3. Xây dựng mạng Bayesian từ ma trận mặt hàng.
4. Thực hiện thao tác khuyến nghị.

Bước *1*, *2* và *3* được thực hiện ngoại tuyến nên không ảnh hưởng đến khả năng đáp ứng thời gian thực của tiến trình khuyến nghị trong bước *4*.

**2.1. Chuyển vị ma trận người dùng thành ma trận mặt hàng**

|  |  |  |  |
| --- | --- | --- | --- |
|  | *item1* | *item2* | *item3* |
| *user1* | r11 = 1 | r12 = 3 | r13 = **?** |
| *user2* | r21 = 3 | r22 = **?** | r23 = 5 |
| *user3* | r31 = 4 | r32 = 2 | r33 = 1 |
| *user4* | r41 = **?** | r42 = **?** | r43 = 3 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *user1* | *user2* | *user3* | *user4* |
| *item1* | r11 = 1 | r21 = 3 | r31 = 4 | r41 = **?** |
| *item2* | r12 = 3 | r22 = **?** | r32 = 2 | r42 = **?** |
| *item3* | r13 = **?** | r23 = 5 | r33 = 1 | r43 = 3 |

**Bảng 2.** Chuyển vị ma trận người dùng thành ma trận mặt hàng

Ma trận người dùng là dạng ban đầu của ma trận đánh giá. Mỗi dòng trong ma trận người dùng là tập đánh giá của một người dùng cụ thể lên nhiều mặt hàng. Ngược lại, đối với ma trận mặt hàng, mỗi dòng là tập đánh giá từ nhiều người dùng lên một mặt hàng cụ thể. Bước này thực hiện việc chuyển vị từ ma trận người dùng thành ma trận mặt hàng. Đây là bước tiền xử lý, tuy đơn giản nhưng rất quan trọng do mạng Bayesian được cấu thành từ các nút mặt hàng. Do số lượng người dùng không giới hạn và gia tăng rất nhiều so với mặt hàng trong ứng dụng thực tế, chúng ta sử dụng ma trận mặt hàng để giữ kích thước mạng Bayesian luôn nhỏ nhằm nhằm cải thiện tốc độ suy diễn đối với tác vụ khuyến nghị (xem bước 4). Bảng 2 là một ví dụ chuyển vị từ ma trận người dùng thành ma trận mặt hàng.

**2.2. Điền những giá trị bị thiếu trong ma trận mặt hàng**

Mạng Bayesian học từ ma trận đánh giá đầy đủ (complete rating matrix) có chất lượng tốt hơn học từ ma trận thưa. Cách đơn giản nhất để hoàn chỉnh dữ liệu không đầy đủ (incomplete data) là thay thế những giá trị bị thiếu bằng các giá trị trung bình (average values). Giá trị trung bình chính là ước lượng (estimate) của giá trị bị thiếu, hay còn gọi giá trị ước lượng. Tiến trình thay thế, còn được gọi tiến trình ước lượng, mang tính chất lặp (iterative) và chồng (overlap):

* Tiến trình thay thế được thực hiện qua nhiều lần lặp. Việc thay thế giá trị bị thiếu tại một lần lặp dựa trên những giá trị được ước lượng trong lần lặp trước đó.
* Giá trị ước lượng trước tiên được xác định như là trung bình (mean) vector người dùng (user vector). Nếu vector người dùng rỗng thì trung bình vector mặt hàng trở thành giá trị ước lượng.

Ví dụ, giả sử với ma trận mặt hàng trong bước *1*, *r41* và *r42* được thay thế như sau:

* *r41 = r43 / 1 = 3*
* *r42 =* (*r41 + r43*) */ 2 =* (*3 + 3*) */ 2 = 3*

Giá trị *r42* được tính theo giá trị thay thế của *r41*, đây là kỹ thuật chồng (overlapping). Bằng cách khác, giá trị *r42* có thể bằng với *r41*, cùng là trung bình (mean) của vector người dùng *user4*. Nếu vậy, giải thuật nhanh hơn do trung bình vector chỉ tính một lần. Tuy nhiên, kỹ thuật chồng hiệu quả và vững chắc hơn. Nếu vector *user4* rỗng thì trung bình của nó không xác định nhưng tiến trình thay thế không gián đoạn. Theo đó, *r42* được gán bằng trung bình (mean) của vector mặt hàng *item2*, ta có *r42 =* (*r12 + r32*) */ 2 = 2.5*. Những giá trị bị thiếu còn lại như *r22*, *r13* được ước lượng như sau:

* *r22 =* (*r21 + r23*) */ 2 =* (*3 + 5*) */ 2 = 4*
* *r13 =* (*r11 + r12*) */ 2 =* (*1 + 3*) */ 2 = 2*

Theo đó, ma trận mặt hàng được ước lượng một cách đầy đủ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *user1* | *user2* | *user3* | *user4* |
| *item1* | 1 | 3 | 4 | 3 |
| *item2* | 3 | 4 | 2 | 3 |
| *item3* | 2 | 5 | 1 | 3 |

**Bảng 3.** Ma trận mặt hàng đầy đủ

Kỹ thuật trung bình nhanh như không chính xác bởi vì các giá trị thay thế chưa phản ảnh đánh giá thật sự của người dùng trên một mặt hàng. Một cách tiếp cận khác để điền giá trị bị thiếu là áp dụng thuật toán cực đại hóa kỳ vọng (expectation maximization – EM). Thuật toán EM là tiến trình lặp nhằm ước lượng giá trị bị thiếu trong dữ liệu ẩn hay không đầy đủ. Mỗi lần lặp của giải thuật EM gồm 2 bước:

* Tại bước kỳ vọng (bước E): dữ liệu bị thiếu được ước lượng trên dữ liệu quan sát và giá trị ước lượng các tham số thống kê. Tham số thống kê thường là kỳ vọng và phương sai.
* Tại bước cực đại hóa (bước M): Tham số thống kê được ước lượng bằng cách cực đại hóa hàm khả năng với giả định rằng đã biết dữ liệu bị thiếu.

Giải thuật EM chính xác hơn giải thuật trung bình do thuật toán EM áp dụng mô hình thống kê vào dữ liệu đánh giá nhằm ước lượng giá trị bị thiếu với một tiến trình lặp mà theo đó, tham số thống kê sau mỗi lần lặp được cải thiện.

**2.3. Xây dựng mạng Bayesian từ ma trận mặt hàng**

Xây dựng mạng Bayesian từ ma trận mặt hàng đầy đủ được tạo ở bước 1 và 2. Mỗi nút trong mạng có năm giá trị {*1, 2, 3, 4, 5*} tương ứng với đánh giá người dùng: *5-*thích nhất và *1-*không thích nhất. Mỗi nút đi kèm một bảng xác suất điều kiện (conditional probability table – CPT) xác định những xác suất tiên nghiệm.

Mạng Bayesian được xây dựng bằng phương pháp máy học, có hai cách tiếp cận:

* Cách tiếp cận điểm tiêu chuẩn (scored-based): giả sử mỗi mạng Bayesian được gán một điểm theo tiêu chuẩn δ, mạng nào có điểm δ cao nhất là mạng tốt nhất. Tiêu chuẩn δ là xác suất hậu nghiệm trên toàn mạng Bayesian với một bộ dữ liệu huấn luyện là ma trận mặt hàng.
* Cách tiếp cận dựa trên các ràng buộc (constraint-based): giả sử có một tập các ràng buộc (constraint), mạng Bayesian tốt nhất sẽ thỏa mãn tất cả các ràng buộc. Các ràng buộc được định nghĩa như các luật liên quan đến điều kiện Markov.

Ý tưởng chính của các phương pháp máy học là tìm ra cấu trúc mạng Bayesian phù hợp nhất. Mỗi phương pháp có nhiều giải thuật không đề cập ở đây vì chúng vượt ra ngoài phạm vi bài báo. Nhìn chung học mạng Bayesian quan trọng nhất trong giải thuật chúng tôi bởi vì cấu trúc mạng Bayesian ảnh hưởng lớn nhất đến kết quả khuyến nghị. Bây giờ chúng ta có một ví dụ đơn giản cho việc học mạng Bayesian bằng cách áp dụng phương pháp điểm tiêu chuẩn vào ma trận mặt hàng đầy đủ (xem bảng 3). Để thuận tiện, chúng ta chuyển ma trận mặt hàng trong bảng 3 thành ma trận nhị phân với mỗi ô được gán *1* nếu giá trị đánh giá tương ứng lớn hơn hay bằng *3*.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *user1* | *user2* | *user3* | *user4* |
| *item1* | 0 | 1 | 1 | 1 |
| *item2* | 1 | 1 | 0 | 1 |
| *item3* | 0 | 1 | 0 | 1 |

**Bảng 4.** Ma trận mặt hàng nhị phân

Mạng Bayesian được học từ ma trận nhị phân trong bảng *4*. Đặt *I1*, *I2* và *I3* đại diện các mặt hàng *1*, *2* và *3* tương ứng. *I1*, *I2* và *I3* là các biến (nút) nhị phân nhận hai giá trị *0* và *1*. Phương pháp điểm tiêu chuẩn sẽ tìm ra mạng Bayesian tốt nhất trong không gian tìm kiếm theo tiêu chuẩn δ. Không gian tìm kiếm là tập các mạng Bayesian, ở đây gồm 3 mạng được ký hiệu *BN1*, *BN2* và *BN3*.

*BN1*

*BN3*

*BN2*

**Hình 1.** Không gian tìm kiếm mạng Bayesian: *BN1, BN2* và *BN3*

Gọi *D* là ma trận nhị phân trong bảng *4*, bảng xác suất điều kiện của mỗi nút được tính theo công thức sau:

Với *a* và *b* là các giá trị nhị phân, chúng ta có các kết quả sau:

* *P*(*I1 = 1*) *= 3/4, P*(*I1 = 0*) *= 1/4*
* *P*(*I2 = 1*) *= 3/4, P*(*I2 = 0*) *= 1/4*
* *P*(*I3 = 1*) *= 1/2, P*(*I3 = 0*) *= 1/2*
* *P*(*I2 = 1| I1 = 1*) *= 2/3, P*(*I2 = 0| I1 = 1*) *= 1/3, P*(*I2 = 1|I1 = 0*) *= 1, P*(*I2 = 0|I1 = 0*) *= 0*
* *P*(*I3 = 1| I1 = 1*) *= 2/3, P*(*I3 = 0| I1 = 1*) *= 1/3, P*(*I3 = 1|I1 = 0*) *= 0, P*(*I3 = 0|I1 = 0*) *= 1*
* *P*(*I1 = 1| I2 = 1*) *= 2/3, P*(*I1 = 0| I2 = 1*) *= 1/3, P*(*I1 = 1|I2 = 0*) *= 1, P*(*I1 = 0|I2 = 0*) *= 0*
* *P*(*I3 = 1| I2 = 1*) *= 2/3, P*(*I3 = 0| I2 = 1*) *= 1/3, P*(*I3 = 1|I2 = 0*) *= 0, P*(*I3 = 0|I2 = 0*) *= 1*
* *P*(*I1 = 1| I3 = 1*) *= 1, P*(*I1 = 0| I3 = 1*) *= 0, P*(*I1 = 1|I3 = 0*) *= 1/2, P*(*I1 = 0|I3 = 0*) *= 1/2*
* *P*(*I2 = 1| I3 = 1*) *= 1, P*(*I2 = 0| I3 = 1*) *= 0, P*(*I2 = 1|I3 = 0*) *= 1/2, P*(*I2 = 0|I3 = 0*) *= 1/2*

Gọi *g1, g2* và *g3* lần lượt là các phân bố xác suất hợp của các mạng *BN1, BN2* và *BN3*, ta có *g1*(*I1, I2, I3*) *= P*(*I1*)*P*(*I2|I1*)*P*(*I3|I1*) , *g2*(*I1, I2, I3*) *= P*(*I2*)*P*(*I1|I2*)*P*(*I3|I2*) và *g3*(*I1, I2, I3*) *= P*(*I3*)*P*(*I1|I3*)*P*(*I2|I3*). Gọi *δ1*, *δ2* and *δ3* lần lượt là các điểm tiêu chuẩn của *BN1, BN2* và *BN3*. Những điểm tiêu chuẩn này biểu diễn xác suất hậu nghiệm của *g1, g2* và *g3* trên bộ dữ liệu ma trận nhị phân *D*. Áp dụng định lý Bayes, xác suất hậu nghiệm của mạng Bayesian *BNi* được xác định như sau:

Giả sử tất cả mạng có cùng xác suất tiên nghiệm *gi*(*I1, I2, I3*) và cùng xác suất trên toàn dữ liệu huấn luyện *gi*(*D*), khi đó *gi*(*I1, I2, I3*) và *gi*(*D*) đều là các hằng số. Không mất tính tổng quát, điểm tiêu chuẩn *δi* được định nghĩa là hàm khả năng *g*(*D|I1, I2, I3*).

Nếu ta gọi mỗi cột (tương ứng với từng người dùng) của *D* là một *trường hợp* (case) trong toàn dữ liệu huấn luyện thì có *4* trường hợp: *c1 =* (*I1 = 0, I2 = 1, I3 = 0*), *c2 =* (*I1 = 1, I2 = 1, I3 = 1*), *c3 =* (*I1 = 1, I2 = 0, I3 = 0*) và *c4 =* (*I1 = 1, I2 = 1, I3 = 1*). Giả sử các trường hợp độc lập lẫn nhau, điểm tiêu chuẩn là tích các xác suất hậu nghiệm trên tất cả trường hợp.

Cụ thể, đối với mạng *BN1* và phân bố hợp *g1*(*I1, I2, I3*) *= P*(*I1*)*P*(*I2|I1*)*P*(*I3|I1*), điểm tiêu chuẩn của mạng *BN1* là:

Tương tự, điểm tiêu chuẩn của mạng *BN2* và *BN3* là:

Dễ dàng nhận thấy điểm tiêu chuẩn *δ3* = 0.0039 lớn nhất, nên mạng *BN3* được chọn là mạng Bayesian tốt nhất được học từ ma trận mặt hàng.

*P*(*I3=1*) = 1/2

*P*(*I3=0*) = 1/2

*I3=1 I3=0*

*I2 =1* 1 1/2

*I2 =0* 0 1/2

*I3=1 I3=0*

*I1 =1* 1 1/2

*I1 =0* 0 1/2

**Hình 2.** Mạng Bayesian tốt nhất cùng các bảng xác suất điều kiện được học từ ma trận mặt hàng

**2.4. Thực hiện thao tác khuyến nghị**

Thao tác khuyến nghị được thực hiện trên cơ chế suy diễn dựa trên bằng chứng của mạng Bayesian. Theo đó, xác suất hậu nghiệm (posterior probabilities, ký hiệu *PoP*) của từng nút trong mạng được tính trước tiên và sau đó, nút mặt hàng nào có *PoP* cao nhất sẽ được khuyến nghị cho người dùng. Toàn mạng Bayesian là mẫu mua sắm của người dùng và những mặt hàng mà họ đánh giá trở thành những nút bằng chứng trong mạng. Phương thức khuyến nghị này có 2 ưu điểm:

* Việc tận dụng mạng Bayesian như là những mẫu mua sắm sẽ giúp phát hiện những sở thích cùng mối quan tâm của người dùng và do đó sẽ cải thiện chất lượng khuyến nghị.
* Suy diễn dựa trên bằng chứng trong mạng Bayesian là một cơ chế suy diễn mạnh mẽ và vững chắc. Điều này sẽ giảm thiểu trung bình lỗi khi ước lượng giá trị bị thiếu.

Giả sử người dùng *U* đã đánh giá mặt hàng *1* với giá trị *3*, hệ thống khuyến nghị sẽ chịu trách nhiệm giới thiệu mặt hàng nào cho người dùng *U*. Giả sử mạng Bayesian tốt nhất (hình 2) được chọn là mạng đích phục vụ cho thao tác khuyến nghị, khi đó *I1* trở thành nút bằng chứng nhị phân có giá trị 1 do mặt hàng *1* được đánh giá là *3*. Sau đây là mạng Bayesian đích với nút bằng chứng được tô bóng:

*P*(*I3=1*) = 1/2

*P*(*I3=0*) = 1/2

*I3=1 I3=0*

*I2 =1* 1 1/2

*I2 =0* 0 1/2

*I3=1 I3=0*

*I1 =1* 1 1/2

*I1 =0* 0 1/2

**Hình 3.** Mạng Bayesian đích với nút bằng chứng I1 = 1

Gọi *PoP*(*I2*), *PoP*(*I3*) tương ứng là xác suất hậu nghiệm của các nút *I2*, *I3* và gọi *g*(*I1, I2, I3*) *= P*(*I3*)*P*(*I1|I3*)*P*(*I2|I3*) là phân bố xác suất hợp của mạng Bayesian đích, ta có:

Khai triển *PoP*(*I2*) và *PoP*(*I3*) ta có:

Dễ dàng tính phân bố xác suất hợp *g*(*I1, I2, I3*) *= P*(*I3*)*P*(*I1|I3*)*P*(*I2|I3*) khi giá trị các biến *I1*, *I2* và *I3* được xác định bằng cách áp dụng các bảng xác suất điều kiện của mạng Bayesian đích. Vì thế, các xác suất hậu nghiệm *PoP*(*I2*) và *PoP*(*I3*) có giá trị cụ thể như sau:

Do xác suất hậu nghiệm *PoP*(*I3*) lớn nhất nên mặt hàng 3 được khuyến nghị cho người dùng với mức ưu tiên cao nhất. Mạng Bayesian đích trong ví dụ này rất nhỏ chỉ với 3 nút nên độ phức tạp tính toán không đáng kể và không ảnh hưởng đến khả năng đáp ứng thời gian thực của hệ thống khuyến nghị nhưng khi kích thước mạng trở nên khổng lồ với rất nhiều nút thì độ phức tạp tính toán trở thành vấn đề quan trọng cần được giải quyết. Phần tiếp theo đề cập đến một kỹ thuật cải tiến nhằm khắc phục vấn đề độ phức tạp tính toán trên.

**3. Một cải tiến giải thuật đề xuất – mạng Bayesian phân cụm**

Mặt dầu số lượng mặt hàng có giới hạn và không tăng nhiều như số lượng người dùng nhưng nó vẫn còn lớn. Hiển nhiên mạng Bayesian hợp thành từ những nút mặt hàng tương quan với nhau nhưng có thể chứa những nút không tương quan do nó được học từ dữ liệu lớn như vậy. Những nút không tương quan này khiến khả năng suy diễn thiếu chính xác. Đây là vấn đề rời rạc hay bất tương quan giữa các nút trong mạng Bayesian, cần được giải quyết.

Giả sử một cửa hàng gồm nhiều mặt hàng chia thành các loại. Áo thun, áo sơ mi, quần dài, quần bò… cùng một loại tương liên nhau là loại quần áo. Các mặt hàng quần áo, một cách tự nhiên, tạo thành một nhóm nút con liên kết nhau trong mạng Bayesian. Nhưng những mặt hàng khác, không liên quan đến quần áo, trở thành những nút rời rạc không liên kết với bất cứ mặt hàng quần áo nào. Cơ chế suy diễn trong mạng Bayesian nếu có những nút rời rạc sẽ dẫn đến sự thiếu chính xác. Vấn đề này được giải quyết khi xây dựng nhiều mạng Bayesian con, mỗi mạng con ứng với một nhóm mặt hàng tương liên. Nói cách khác, toàn bộ mạng Bayesian lớn được phân rã thành nhiều mạng Bayesian cá thể (individual Bayesian network) sao cho mối quan hệ giữa các nút trong mạng cá thể chặt chẽ hơn. Ví dụ, chúng ta xây dựng ba mạng Bayesian ứng với ba nhóm mặt hàng trong siêu thị: quần áo, đồ nội thất và điện gia dụng.

Trong trường hợp dữ liệu huấn luyện như ma trận đánh giá không đặc tả rõ ràng các nhóm mặt hàng, kỹ thuật gom cụm được áp dụng để phân nhóm mặt hàng. Theo đó, có hai bước cải tiến việc xây dựng mạng Bayesian:

* Áp dụng các phương pháp gom cụm như *K-mean, K-centroid* để phân nhóm mặt hàng. Chúng ta có thể phân nhóm một cách thủ công, khi đó một mặt hàng có thể thuộc nhiều hơn một nhóm.
* Đối với mỗi nhóm mặt hàng:
  + Dữ liệu huấn luyện được thu gọn. Cụ thể là, đối với mỗi dòng của ma trận đánh giá, những cột không tương ứng với các mặt hàng trong nhóm này sẽ bị loại bỏ.
  + Mạng Bayesian được học từ dữ liệu huấn luyện thu gọn này, được gọi là mạng Bayesian cá thể (individual Bayesian network).

Lưu ý, mỗi nút có thể thuộc về nhiều hơn một mạng cá thể, đây là một nhược điểm nhưng thường xuất hiện trong ứng dụng thương mại, một mặt hàng có thể được xếp vào nhiều nhóm.

Mỗi lần có tác vụ khuyến nghị được yêu cầu như trong bước 4, tiến trình suy diễn sẽ được thực thi trên mạng Bayesian cá thể thay vì trên toàn mạng lớn như trước đây. Tốc độ được cải thiện khi số lượng nút trong mạng cá thể nhỏ hơn rất nhiều so với toàn mạng lớn. Nhưng một vấn đề khác đặt ra là làm thế nào chọn đúng mạng cá thể để thực hiện tác vụ khuyến nghị khi mà rất tốn thời gian và tài nguyên máy tính nếu chúng ta duyệt qua tất cả các mạng cá thể cùng với nút của chúng để tìm đúng mạng thích hợp chứa nhiều nhất các mặt hàng yêu thích của người dùng. Vì thế, cần có một cách tiếp cận khác. Đây là một nghiên cứu mở nhưng chúng tôi đề xuất một giải pháp được gọi là *bảng ánh xạ* (mapping table – MT).

Ý tưởng chính của kỹ thuật bảng ánh xạ là tạo ra một bảng ánh xạ (mapping table) tại thời điểm xây dựng mạng Bayesian (học mạng). Mỗi dòng của bảng ánh xạ là một cặp khóa-giá trị. Khóa là tên nút mặt hàng, giá trị là tập bit nhị phân xác định nút nào thuộc về mạng cá thể nào. Mỗi bit của tập bit này biễu diễn sự xuất hiện của một mạng cá thể hay nói cách khác mạng cá thể này có chứa hay không chứa nút được đặc tả bởi khóa. Giả sử có 3 mạng cá thể *BN0, BN1, BN2* và 6 nút *A, B, C, D, E, F, G, H*, bảng ánh xạ được mô tả như sau:

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

**Bảng 5.** Bảng ánh xạ

Bảng ánh xạ được diễn dịch như sau:

* Nút *A* và *B* thuộc mạng *BN0*.
* Nút *C* và *D* thuộc mạng *BN1*.
* Nút *E* và *F* thuộc mạng *BN2*.
* Nút *G* và *H* thuộc mạng *BN1* và *BN2* tương ứng.

Cho trước một người dùng cùng với các mặt hàng mà người đó đã đánh giá, đối với từng mạng cá thể, số lượng nút mặt hàng xuất hiện trong mạng này được đếm. Mạng nào có số nút lớn nhất là mạng thích hợp để thực hiện khuyến nghị bằng những tác vụ suy diễn. Ví dụ, giả sử các mặt hàng mà người dùng *a* đã đánh giá là *E, F, G* và *H*, ta có:

* Tổng số nút chứa trong *BN0* là *0*, *t0* = *0* do *BN0* không chứa bất kỳ nút nào người dùng *a* đánh giá.
* Tổng số nút chứa trong *BN1* là *2*, *t1* = *2* do *BN1* chứa *2* nút *G* và *H*.
* Tổng số nút chứa trong *BN2* là *4*, *t2* = *4* do *BN2* chứa *4* nút *E, F, G* và *H*.

Do *t4* lớn nhất nên *BN2* là mạng cá thể thích hợp để thực hiện suy diễn khuyến nghị.

**4. Đánh giá**

Chúng tôi sử dụng cơ sở dữ liệu Movielen [1] gồm 100.000 đánh giá của 943 người dùng trên 1682 mặt hàng. Cơ sở dữ liệu được chia thành 5 nhóm (folder), mỗi nhóm chứa 80% tập huấn luyện (training set) và 20% tập kiểm thử (testing set). Tập huấn luyện độc lập tập kiểm thử trong cùng một bộ.

Thiết lập hệ thống phần cứng và phần mềm bao gồm: bộ xử lý Dual-Core CPU E5700 @ 3.00GHz, RAM 2GB, RAM còn trống 1GB, Microsoft Windows 7 Ultimate 2009 32-bit, Java 7 HotSpot (TM) Client VM. Giải thuật đề xuất được so sánh với bốn giải thuật khác: Green Fall là giải thuật khuyến nghị dựa trên khai thác tập phổ biến, láng giềng gần nhất dựa trên mặt hàng (item-based), láng giềng gần nhất dựa trên người dùng (user-based) và SVD.

Đánh giá sử dụng 7 độ đo: *MAE, MSE, RMSE, độ chuẩn xác, độ bao phủ, F1, ARHR* [11] and *thời gian xử lý*. Thời gian xử lý được tính bằng giây. MAE và MSE đo lường độ chính xác giữa giá trị dự đoán (ước lượng) và giá trị đánh giá thực tế. MAE và MSE càng nhỏ thì độ chính xác càng cao. Độ chuẩn xác, độ bao phủ, F1 đo lường chất lượng khuyến nghị, nghĩa là đo lường mức độ yêu thích của người dùng đối với những mặt hàng được khuyến nghị, nói cách khác, những mặt hàng mà hệ thống khuyến nghị cho người dùng phản ánh bao nhiêu phần trăm sở thích và mối quan tâm của người dùng. ARHR cũng là độ đo chất lượng khuyến nghị theo một khía cạnh khác – đo lường mức độ phù hợp giữa thứ tự các mặt hàng được khuyến nghị với thứ tự các mặt hàng mà người dùng đánh giá. Các độ đo chất lượng càng lớn, giải thuật càng tốt. Sau đây là kết quả đánh giá:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Giải thuật đề xuất** | **Green Fall** | **Item-based** | **User-based** | **SVD** |
| MAE | *0.6127* | *0.8529* | *0.5222* | *0.9319* | *0.5363* |
| MSE | *0.9023* | *1.6304* | *0.6675* | *2.1664* | *1.1734* |
| Độ chính xác | *0.1334* | *0.1058* | *0.0245* | *0.0014* | *0.0041* |
| Độ bao phủ | *0.0513* | *0.0404* | *0.0092* | *0.0005* | *0.0015* |
| F1 | *0.0731* | *0.0576* | *0.0131* | *0.0008* | *0.0021* |
| ARHR | *0.0454* | *0.0341* | *0.0040* | *0.0003* | *0.0012* |
| Thời gian | *1.8169* | *0.0050* | *8.5607* | *7.5558* | *0.0207* |

**Bảng 6**. Kết quả đánh giá

Giải thuật chúng tôi hiệu quả hơn nhiều so với các giải thuật khác qua các độ đo chất lượng như *độ chuẩn xác, độ bao phủ, F1, ARHR*. Độ chính xác xấp xỉ item-based, user-based, SVD và tốt hơn Green Fall qua các độ đo MAE, MSE và RMSE. Giải thuật tốn nhiều thời gian hơn Green Fall và SVD nhưng ít thời gian hơn item-based và user-based.

**5. Kết luận**

Điểm cơ bản của giải thuật đề xuất là xây dựng mạng Bayesian từ ma trận đánh giá (rating matrix) và áp dụng suy diễn trong mạng Bayesian cho việc khuyến nghị mặt hàng. Mạng Bayesian là đồ thị không vòng gồm một tập nút và một tập cạnh. Nút đại diện mặt hàng và cạnh biểu diễn mối quan hệ phụ thuộc giữa hai nút. Bất cứ khi nào tác vụ khuyến nghị khởi hoạt, cơ chế suy diễn trong mạng Bayesian sẽ thực thi dựa vào các bằng chứng từ ma trận đánh giá, theo đó, mặt hàng nào có xác suất hậu nghiệm cao nhất sẽ được giới thiệu đến người dùng. Vì độ chính xác của tiến trình khuyến nghị phụ thuộc vào chất lượng của cấu trúc mạng Bayesian nên có hai cách để đạt được cấu trúc mạng tốt nhất:

* Chọn thuật toán học mạng thích hợp – không đề cập trong nghiên cứu này.
* Điền những giá trị bị thiếu bằng các giá trị ước lượng nhằm biến ma trận đánh giá không đầy đủ (ma trận thưa) thành ma trận đầy đủ. Mạng Bayesian học từ dữ liệu đầy đủ sẽ đạt chất lượng cao.

Giải thuật trung bình và EM được đề xuất để bổ sung và hoàn chỉnh ma trận thưa. Phương pháp trung bình nhanh và đơn giản hơn EM nhưng kém chính xác.

Giải thuật đề xuất được đánh giá và so sánh với các giải thuật nạp ức và mô thức khác như Green Fall, item-based, user-based và SVD. Kết quả cho thấy giải thuật đề xuất hiệu quả nhất khi đạt chất lượng khuyến nghị cao qua các độ đo như độ chuẩn xác, độ bao phủ, F1 và ARHR mặc dù Green Fall là giải thuật nhanh nhất đáp ứng yếu tố khuyến nghị thời gian thực. Tuy nhiên, thời gian mô hình hóa dữ liệu trong giải thuật đề xuất dài hơn Green Fall, nói cách khác, thời gian xây dựng mạng Bayesian lâu hơn thời gian khai thác tập phổ biến. Nhìn chung giải thuật khuyến nghị dựa trên mạng Bayesian tối ưu với tiêu chí chất lượng khuyến nghị và giải thuật khuyến nghị dựa trên khai thác tập phổ biến Green Fall tối ưu với tiêu chí tốc độ đáp ứng thời gian thực.

**Lời cảm ơn**

Cảm ơn bà Đỗ Thị Minh Phụng – Trường Đại học Công nghệ Thông tin, Đại học Quốc Gia Thành phố Hồ Chí Minh, Việt Nam và ông Vũ Ngọc Đồng đã giúp tôi hoàn thiện và cải tiến nghiên cứu này bằng những góp ý và lời khuyên bổ ích.

**Tham khảo**

1. [Su, Khoshgoftaar 2009]. Xiaoyuan Su and Taghi M. Khoshgoftaar. A Survey of Collaborative Filtering Techniques. Hindawi Publishing Corporation, Advances in Artificial Intelligence, Volume 2009, Article ID 421425, 19 pages, doi:10.1155/2009/421425.
2. [Movielens dataset 2011]. Home page is http://www.movielens.org. Download dataset from http://www.grouplens.org/node/12.
3. [Richard 2003]. Richard E. Neapolitan. Learning Bayesian Networks. Northeastern Illinois University Chicago, Illinois 2003.
4. [Ungar, Foster 1998]. L. H. Ungar and D. P. Foster. Clustering Methods for Collaborative Filtering. Proceedings of the Workshop on Recommendation Systems, AAAI Press, 1998.
5. [Han, Kamber 2006]. Jiawei Han and Michelline Kamber. Data Mining: Concepts and Techniques. Second Edition. © 2006 by Elsevier Inc.
6. [Breese, Heckerman, Kadie 1998]. John S. Breese, David Heckerman and Carl Kadie. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. Technical Report MSR-TR-1998. Microsoft Research, Microsoft Corporation, One Microsoft Way, Redmon, WA 98052.
7. [Heckerman, Chickering, Meek, Rougnthwaite 2000]. David Heckerman, David Maxwell Chickering, Christopher Meek, Robert Rougnthwaite and Carl Kadie. Dependecy Networks for Inference, Collaborative Filtering and Data Visualization. Journal of Machine Learning Research 1 (2000) 49-75.
8. [Hofmann 2004]. Thomas Hofmann. Latent Semantic Models for Collaborative Filtering. ACM Transactionson Information Systems, Vol.22, No.1, January 2004, Pages 89-115.
9. [Shani, Heckerman, Brafman 2005]. G. Shani, D. Heckerman, and R. I. Brafman. An MDP-based Recommender System. Journal of Machine Learning Research, vol. 6, pp. 1265-1295, 2005.
10. [Rennie 2004]. Jason D. M. Rennie (jrennie@csail.mit.edu). A Short Tutorial on Using Expectation-Maximization with Mixture Models. MIT University, March 3, 2004
11. [Herlocker, Konstan, Terveen, Riedl 2004]. Jonathan. L. Herlocker, Joseph. A. Konstan, Loren G. Terveen, and John. T. Riedl. Evaluating Collaborative Filtering Recommender Systems. ACM Transactions on Information Systems, vol. 22, no. 1, pp. 5–53, 2004.