

Time-aware Recommender System Using Naïve Bayes Classifier Weighting Technique

Sutheera Puntheeranurak

Faculty of Engineering
King Mongkut's Institute of Technology Ladkrabang
Bangkok, 10520 Thailand
kpsuthee@kmitl.ac.th

Pongpan Pitakpaisarnsin

Faculty of Engineering
King Mongkut's Institute of Technology Ladkrabang
Bangkok, 10520 Thailand
pongpan43@gmail.com

Abstract— Collaborative filtering method have been widely used in the recommender system which has problems of scalability and highly time consuming between the process of recommendation and traditional method didn't concern over the difference of time that a user has rating for an item. In this paper, we proposed a Naïve Bayes Classifier technique that uses the user rating for the item and item information to construct a model with time awareness. It will provide better scalability and accuracy recommendation result even the system has more sparsity. In addition, it decreases time consuming by constructing a model in the offline and calculate the recommendation results in online phase. We show the result from our proposed can make the system get more efficient.

Keywords-component; Recommender system, Naïve Bayes Classifier, Time-awareness

I. INTRODUCTION

In the present year where the Internet has been globally used, everyone can access to the Internet from their home, workplace or anywhere with mobile connection that makes the numbers of users is growth respectively. The information on the Internet is also increased, which is the problem of information overload that makes users hard to find what they interested in. For example, an E-Commerce shopping website has a great number of items, which cause users to waste time to find what they want. In other hands, a website owner can lose their customers because of that. Recommender systems have been used to solve this problem and have been widely used in many E-commerce, the content providers such as movie, music, multimedia, food recipe, etc., which uses user preference to produce a recommendation result.

There are many kinds of techniques that have been used in recommender systems. First, a content-based filtering [1], that finds the items to recommend by using user profile. It can show the items that the user is interested in. The second, a collaborative filtering [2] [3] [4], that uses to find other users that have similar preference or same interest, which is called neighbors. And It makes recommended result based on those neighbors to recommend the item that the user has no rating before. And the other kind is a hybrid technique that combines content-based filtering and collaborative filtering to solve the weakness of each other.

The collaborative filtering is widely used in the recommender system, and it can be divided into two types. First, a memory-based filtering [2] [3] uses all data to make recommendation results, which have more accuracy in the cost of high computation and low scalability. The second type is a model-based filtering [5], which construct models from all data offline and make recommendation results based on that model in online mode. It can improve scalability but have lower accuracy than memory-based filtering.

Hybrid user model [10] is proposed by constructing a user model using user rating, detailed item description and user demographic information and genetic algorithm are used to improve accuracy before generating recommendation based on both user model and user rating matrix.

Traditional method using only user-item rating to compute the recommendation result, it does not concern about when users have rated for items. Time when users have rated for an item can be useful information, for example, one product at that time it has high rating but when the time pass. It has lower ratings than the previous time. It can be that item has lost interest.

Recently, there have been many researchers who come up with the ideas to make a recommender system with time consideration when a user has rated items. E.g. a user model which includes user rating, detailed item information, user demographic information and time when users have rated item information into the user model [8] to generate recommendations based on the user model in each time range and the original user rating matrix are being used with each other. The other researchers used collaborative filtering with time weighting [11] [12] which combine user rating and time weight together to sense the change of user preference and can adjust the recommendation to match users' preferences at a specific time properly and other than just user rating time. The item is being weighted along with the user rating and computing before the process of recommendation [13].

In this paper, we propose a method which has four steps. First is to construct a user model in offline phase by using Naïve Bayes classifier weighting technique [6] [7] and using a linear combination to calculate user-item rating and item features information to have rated items. Second step is to

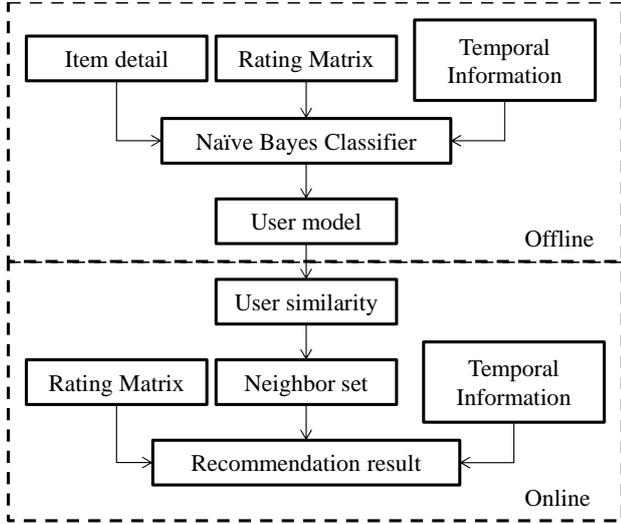


Figure 1. Workflow of model-based Naïve Bayes Classifier Weighting

find user similarity by computing the relevant similarity of feature vector between target user and other users. A we get Top-N most similar users who are called neighbors. Final step is to predict the rating for the user by using the Top-N item that has highest rating will be the recommendation result.

II. OUR APPROACH MODEL

The workflow of Naïve Bayes Classifier Weighting user model is shown in Fig. 1. Firstly, a model offline is constructed by using Naïve Bayes Classifier Weighting Technique based on rating with a time range for each rating data in the model. The model size is much smaller than the original user-item rating matrix and can improve scalability of the system. After the process of constructing a model, we compute similarity between the user base on the model and generate the neighbor set which uses less time in the computation when compared to the whole rating matrix. In the final step, we generate the recommendation results by using memory-based collaborative filtering on the entire rating matrix, and neighbor set that we compute from the model.

A. Naïve Bayes Classifier

Naïve Bayes Classifier is one of the probabilistic classifiers based on Bayes Theorem with independence assumption is defined as follows.

$$p(c|f_i) = \frac{p(c)p(f_i|c)}{p(f_i)} \quad (1)$$

Where $p(c|f_i)$ is the probability of class c given the feature f_i . $p(c)$ is the prior probability of class c . $p(f_i|c)$ is the probability of the feature f_i given the class c .

The Naïve Bayes is that all features are conditionally independent [9] given the class label as follows.

$$p(c|f_i) = p(c) \prod_{i=1}^n p(f_i|c) \quad (2)$$

Naïve Bayes classifier computes the maximum posterior probability of class c given feature f_i . The class with the highest posterior probability for this model is as follows.

$$\text{classify}(f_1, \dots, f_n) = \max p(C = c) \prod_{i=1}^n p(F_i = f_i | C = c) \quad (3)$$

Where p is the posterior probability of an event, c is the class of a model, f_i is featured, n is the number event.

B. Naïve Bayes Classifier Weighing Technique

Naïve Bayes Classifier Weighing Technique is an algorithm that uses a Naïve Bayes theorem for weighing preference that use a linear combination to combine all weighed preferences to construct a model as follows.

$$V_{i,k,t} = \frac{\sum_{r=1}^5 r * p(c_r | User = i, Feature = k, Time = t)}{\sum_{r=1}^5 p(c_r)} \quad (4)$$

Where $v_{i,k,t}$ is a rating in a model that is given by user i for item feature k at time t , $p(c_k | user=i, item feature=k, time =t)$ is the posterior probability of user i give rating for item feature k in time t with r is the number of classes.

C. Example User Model

We are going to make an example to explain the concept of model constructing. The user-item rating matrix in Table.I contains of two users, rating range from 1 to 5, 3 items, 2 item feature and 2 times range. Each item has only one feature. The user can rate same item in different time range. We compute Naïve Bayes Classifier Weighing Technique from data in Table. I and result show in Table. II.

In Table. II, we get the result by computing Naïve Bayes Classifier Weighing. The user model showed in specific time range. The user model is constructed in offline process, and we use a model to find nearest neighbor and generate the recommendation results in the online process, which can reduce time that users have to wait for recommendations and user model size is smaller than the original user-item rating matrix. It is reduced from user*item to user*item-feature size that time for similarity to take less time when compared to the original user-item rating matrix as we will describe about collaborative filtering.

III. COLLABORATIVE FILTERING RECOMMENDATION

The main concept of collaborative filtering is to find a similarity between users based on users' preference. A user

TABLE I. EXAMPLE OF USER PREFERENCE

User	Item Feature			Time
	Item	Feature	Rating	
1	1	1	1	1
1	2	1	3	1
1	3	2	4	1
2	1	1	2	1
2	1	1	1	2
2	2	1	4	1
2	2	1	3	2
2	3	2	5	1

TABLE II. NAÏVE BAYES CLASSIFIER WEIGHTING USER MODEL

User	Time 1		Time 2	
	Item Feature 1	Item Feature 2	Item Feature 1	Item Feature 2
1	2.76041	3.30208	2.46875	3.01041
2	2.84375	3.38541	2.55208	3.09375

who has similarly interested is called a neighbor. Collaborative filtering can generate recommendation based on these neighbors. The process takes two steps: Its first is to find nearest neighbor's data and final step is to compute the recommendation result based on these neighbors set.

A. Find user neighbors

Neighbors are users who have high similarity to target user. There are many kinds of similarity algorithms, for example, Pearson's correlation, Spearman correlation, cosine vector, adjusted cosine vector, mean square difference and relevant similarity. We applied relevant similarity for computing similarity between users. The formula is as follows.

$$sim(a, b, t) = \frac{\sum_{k=1}^{p+q} |H_{akt} - \overline{H_{at}}| \times |H_{bkt} - \overline{H_{bt}}|}{\sum_{k=1}^{p+q} (H_{bkt} - \overline{H_{bt}})^2 \times \sum_{k=1}^{p+q} (H_{akt} - \overline{H_{at}})^2} \quad (5)$$

Where H_{akt} and H_{bkt} are a Naïve Bayes classifier weighting value for feature k at time t of user a and b respectively. $\overline{H_a}$ and $\overline{H_b}$ are the average rating of user a and b in time t respectively.

B. Predicted rating

Neighbors set is used in the prediction process, to calculate a rating formula as follows.

$$p(a, j, t) = \overline{R_{at}} + \frac{\sum_{b \in U} Sim(a, b, t) \times (R_{bjt} - \overline{R_{bt}})}{\sum_{b \in U} Sim(a, b, t)} \quad (6)$$

Where $P(a, j, t)$ is predicted rating that a user a has rated an item j at time t . U is the nearest neighbors set of the user a .

R_{at} is the average rating of the user a in time t . R_{bjt} is the rating of neighbor b for the item j at time t . $\overline{R_{bt}}$ is the average rating of neighbor b at time t .

Set of an item with highest predict ratings will be the recommendation result for that user.

IV. PERFORMANCE MEASUREMENT

We use Epinion.com dataset for an experiment that each rating has timely information. The data set contains 922,267 rating with 22,164 users and 296,277 items with 27 features and have 11 times range each time range is one year. This dataset rating value is 1 to 5. The high rating means the user likes this item. The experimental data are divided into training data and testing data. The percentage of training data is 90, 80, 70, 60 and 50 respectively.

Mean Absolute Error (MAE) is being used to evaluate the recommendation accuracy. MAE has computed between real rating and predicted rating. MAE formula is as follows.

We are going to make an example to explain the concept of model constructing. The user-item rating matrix in Table.I contains of two users, rating range from 1 to 5, 3 items, 2 item feature and 2 times range. Each item has only one feature. The user can rate same item in different time range. We compute Naïve Bayes Classifier Weighting Technique from data in Table. I and result show in Table. II.

In Table. II, we get the result by computing Naïve Bayes Classifier Weighting. The user model showed in specific time range. The user model is constructed in offline process, and we use a model to find nearest neighbor and generate the recommendation results in the online process, which can reduce time that users have to wait for recommendations and user model size is smaller than the original user-item rating matrix. It is reduced from user*item to user*item-feature size that time for similarity to take less time when compared to the original user-item rating matrix as we will describe about collaborative filtering.

$$MAE = \frac{\sum_{i=1}^n |x_i - y_i|}{n} \quad (7)$$

Where x_i is predicted rating and y_i is an actual rating and n is the number of items with a rating in testing data. The smaller MAE means higher accuracy of recommendation.

V. CONCLUSION

The user model combines with collaborative filtering method can solve the problems of system scalability, sparse data. We compute a user model in the offline mode and predict recommendation in online mode that can reduce computing time. Naïve Bayes classifier is used to construct a user model by combining user-item rating, item feature's information and time information, which improve recommendation accuracy and can provide an item to user preference at that time which can satisfy users more.

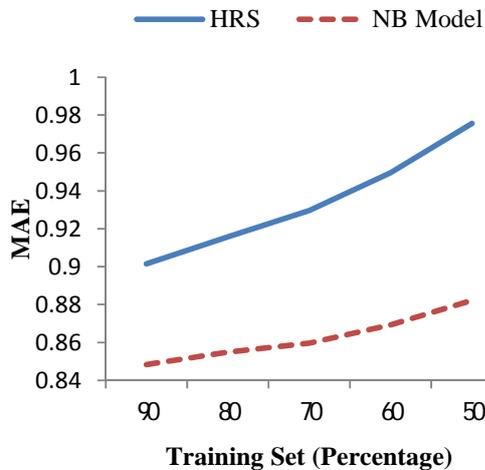


Figure 2. Comparison of MAE (Training set 90 to 50)

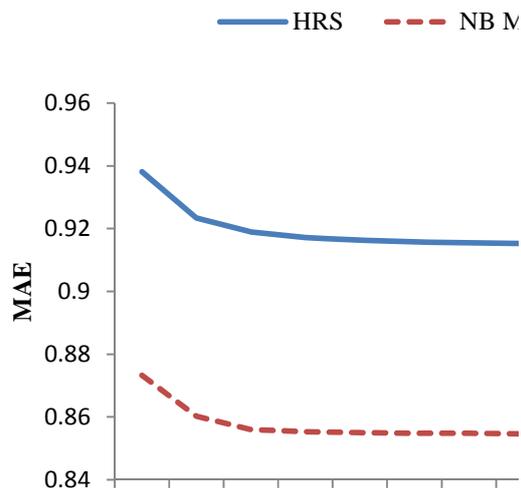


Figure 3. Comparison of MAE (neighbor set 5 to 50)

REFERENCES

- [1] Lang, K. (1995) NewsWeeder: Learning to filter Netnews. In Proceeding of the 12th international conference on machine learning, Tahoe City, CA.
- [2] Resnick, P., Iakovou, N., Sushak, M., Bergstrom, P., & Riedl, J. (1994) Grouplens: An open architecture for collaborative filtering of Netnews. In Proceedings of ACM conference on computer supported cooperative work (pp. 175-186) NC: Chapel Hill
- [3] Shardanand, U., & Macs, P. (1995) Social information filtering: Algorithms for automating 'Word of Mouth'. In Proceedings of the conference on human factors in computing systems (CHI'95), Denver (pp. 210-217)
- [4] Boucher-Ryan, P. D., & Bridge, D. (2006) Collaborative recommending using formal concept analysis. Knowledge-based Systems, 19, (pp. 309-315)
- [5] Shahabi, C., Banaei-Kashani, F., Chen, Y., & McLeod, D. (2001) Yoda: An accurate and scalable web-based recommendation systems. In proceedings of the sixth international conference on cooperative information systems (CoopIS 2001), Trento, Italy.
- [6] L. Korranat and B. Veera, "Collaborative filtering with naïve bayes weighing," Proceedings of the Second Conference on Knowledge and Smart Technologies, pp.53-58, 2010.
- [7] Puntheeranurak S., Sanprasert S. "Hybrid Naive Bayes Classifier Weighting and Singular Value Decomposition Technique for Recommender System", In proceedings of the second IEEE international conference on software engineering and service science (ICSESS2011), Beijing, China.
- [8] Farman U., Ghulam S., Sung C. L., Yun K. P., Kyeong D. M., Jin T. K. (2012) ICOIN; "Hybrid Recommender System with Temporal information" (pp.421-425).
- [9] K. P. Murphy, "Naïve Bayes classifiers", 2006.
- [10] Qian W., Xianhu Y., Min S. "Collaborative Filtering Recommendation Algorithm based on Hybrid User Model" International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2010)
- [11] Chen D., Xu D. "A Collaborative Filtering Recommendation Based on User profile weight and Time weight" Computational Intelligence and Software Engineering (CiSE) 2009
- [12] Yang H.Z., Li L. "An Enhanced Collaborative Filtering Algorithm Based on Time Weight" International Symposium on Information Engineering and Electronic Commerce 2009
- [13] Yuchuan Z., Yuzhao L. "A Collaborative Filtering Algorithm based on Time period Partition" 3rd International Symposium on Intelligent Information Technology and Security Informatics.