A New Collaborative Filtering Recommendation Approach Based on Naive Bayesian Method

Kebin Wang and Ying Tan

Key Laboratory of Machine Perception (MOE), Peking University
Department of Machine Intelligence, School of Electronics Engineering and Computer Science, Peking University, Beijing, 100871, China
caesar1017@gmail.com, ytan@pku.edu.cn

Abstract. Recommendation is a popular and hot problem in e-commerce. Recommendation systems are realized in many ways such as content-based recommendation, collaborative filtering recommendation, and hybrid approach recommendation. In this article, a new collaborative filtering recommendation algorithm based on naive Bayesian method is proposed. Unlike original naive Bayesian method, the new algorithm can be applied to instances where conditional independence assumption is not obeyed strictly. According to our experiment, the new recommendation algorithm has a better performance than many existing algorithms including the popular k-NN algorithm used by Amazon.com especially at long length recommendation.

Keywords: recommender system, collaborative filtering, naive Bayesian method, probability.

1 Introduction

Recommendation systems are widely used by e-commerce web sites. They are a kind of information retrieval. But unlike search engines or databases they provide users with things they have never heard of before. That is, recommendation systems are able to predict users' unknown interests according to their known interests[8],[10]. There are thousands of movies that are liked by millions of people. Recommendation systems are ready to tell you which movie is of your type out of all these good movies. Though recommendation systems are very useful, the current systems still require further improvement. They always provide either only most popular items or strange items which are not to users' taste at all. Good recommendation systems have a more accurate prediction and lower computation complexity. Our work is mainly on the improvement of accuracy.

Naive Bayesian method is a famous classification algorithm[6] and it could also be used in the recommendation field. When factors affecting the classification results are conditional independent, naive Bayesian method is proved to be the solution with the best performance. When it comes to the recommendation field, naive Bayesian method is able to directly calculate the probability of user's possible interests and no definition of similarity or distance is required, while in

Y. Tan et al. (Eds.): ICSI 2011, Part II, LNCS 6729, pp. 218–227, 2011.

other algorithms such as k-NN there are usually many parameters and definitions to be determined manually. It is always fairly difficult to measure whether the definition is suitable or whether the parameter is optimal. Vapnik's principle said that when trying to solve some problem, one should not solve a more difficult problem as an intermediate step. On the other side, although Bayesian network[7] have good performance on this problem, it has a great computational complexity.

In this article, we designed a new collaborative filtering algorithm based on naive Bayesian method. The new algorithm has a similar complexity to naive Bayesian method. However, it has an adjustment of the independence which makes it possible to be applied to the instance where conditional independence assumption is not obeyed strictly. The new algorithm provides us with a new simple solution to the lack of independence other than Bayesian networks. The good performance of the algorithm will provide users with more accurate recommendation.

2 Related Work

2.1 Recommendation Systems

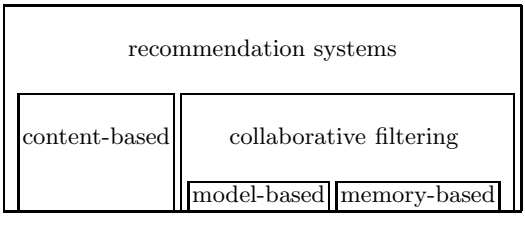
As shown in Table 1, recommendation systems are implemented in many ways. They attempt to provide items which are likely of interest to the user according to characteristics extracted from the user's profile. Some characteristics are from content of the items, and the corresponding method is called content-based approach. In the same way, some are from the user's social environment which is called collaborative filtering approach[12].

Content-based approach reads the content of each item and the similarity between items is calculated according to characteristics extracted from the content. The advantages of this approach are that the algorithm is able to handle brand new items, and the reason for each recommendation is easy to explain. However, not all kinds of items are able to read. Content-based systems mainly focus on items containing textual information[13], [14], [15]. When it comes to movies, the content-based approach does not work. Therefore in this problem, we chose collaborative filtering approach.

Compared to content-based approach, collaborative filtering approach does not care what the items are. It focuses on the relationship between users and items. That is, in this method, items in which similar users are interested are considered similar[1],[2].

Here we mainly talk about collaborative filtering approach.

Table 1. Various recommendation systems



2.2 Collaborative Filtering

Collaborative filtering systems try to predict the interest of items for a particular user based on the items of other users' interest. There have been many collaborative systems developed in both academia and industry[1]. Algorithms for collaborative filtering can be grouped into two-general classes, memory-based and model-based[4], [11].

Memory-based algorithms essentially are heuristics that make predictions based on the entire database. Values deciding whether to recommend the item is calculated as an aggregate of the other users' records for the same item.[1]

In contrast to memory-based methods, model-based algorithms first built a model according to the database and then made predictions based on the model[5]. The main difference between model-based algorithms and memory-based methods is that model-based algorithms do not use heuristic rules. Instead, models learned from the database provide the recommendations.

The improved naive Bayesian method belongs to the model-based algorithms while the k-NN algorithm which appears as a comparison later belongs to the memory-based algorithms.

2.3 k-NN Recommendation

k-NN recommendation is a very successful recommendation algorithm used by many e-commerce web sites including Amazon.com[2], [9].

The k-NN recommendation separates into item-based k-NN and user-based k-NN. Here we mainly talk about item-based k-NN popularized by Amazon.com.

First an item-to-item similarity matrix using cosine measure is built. For each pair of items in the matrix, the similarity is defined as the cosine value of two item-vectors. The item-vectors' M dimensions corresponding to the M users is one, which means the user is interested in the item, or zero otherwise.

The next step is to infer each user's unknown interests using the matrix and his known interests. The items most similar to his known interests will be recommended according to the matrix.

3 Improved Naive Bayesian Method

3.1 Original Naive Bayesian Method

For each user, we are supposed to predict his unknown interests according to his known interests. User's unknown interest is expressed in such a way.

$$p(m_x|m_{u_1}, m_{u_2}, \cdots) \tag{1}$$

When considering the user's interest on item m_x, we have $m_{u_1}, m_{u_2} \cdots$ as known interests. Of course, m_x is not included by the user's known interests. The

conditional probability means the possibility of the item m_x being an interest of the user whose known interests are m_{u_1}, m_{u_2}, etc. In our algorithm, the items of higher conditional probability have higher priority to be recommended and our job is to compute the conditional probability of each item for each user.

$$p(m_x|m_{u_1}, m_{u_2}, \cdots) = \frac{p(m_x) \cdot p(m_{u_1}, m_{u_2}, \cdots|m_x)}{p(m_{u_1}, m_{u_2}, \cdots)} \tag{2}$$

We have the conditional independence assumption that

$$p(m_{u_1}, m_{u_2}, \cdots|m_x) = p(m_{u_1}|m_x) \cdot p(m_{u_2}|m_x) \cdot \cdots \tag{3}$$

In practice, comparison only occurred among the conditional probabilities of the same user where the denominators of equation (2) $p(m_{u_1}, m_{u_2}, \cdots)$ are all the same and have no influence on the final result. Therefore its calculation is simplified as (4).

$$p(m_{u_1}, m_{u_2}, \cdots) = p(m_{u_1}) \cdot p(m_{u_2}) \cdot \cdots \tag{4}$$

So the conditional probability can be calculated in this way.

$$p(m_x|m_{u_1}, m_{u_2}, \cdots) = p(m_x) \cdot q, \tag{5}$$

where

$$q = \frac{p(m_{u_1}, m_{u_2}, \cdots|m_x)}{p(m_{u_1}, m_{u_2}, \cdots)} = \frac{p(m_{u_1}|m_x)}{p(m_{u_1})} \cdot \frac{p(m_{u_2}|m_x)}{p(m_{u_2})} \cdot \cdots \tag{6}$$

3.2 Improved Naive Bayesian Method

In fact, the conditional independence assumption is not suitable in this problem. Because the relevance between items is the theory foundation of our algorithm.

$p(m_x)$ in (5) shows whether the item itself is attractive, and q shows whether the item is suitable for the very user. In our experiment, it is revealed that the latter has more influence than it deserved because of the lack of independence. To adjust the bias we have

$$p(m_x|m_{u_1}, m_{u_2}, \cdots) = p(m_x) \cdot q^{\frac{c_n}{n}} \tag{7}$$

n is the number of the user's known interests and c_n is a constant between 1 and n. The transformation makes the influence of the entire n known interests equivalent to the influence of c_n interests, which will greatly decrease the influence of the user's known interests. Actually, c_n represents how independent the items are. The value of c_n is calculated by experiments and for most of the n's the value is around 3.

3.3 Implementation of Improved Naive Bayesian Method

Calculation of prior probability. First we calculate the prior probability $p(m_i)$. The prior probability is the possibility that the item m_i is interesting to all the users. The algorithm 1 shows how we do the calculation.

```
foreach item i in database do
    foreach user that interested in the item do
        t_i = t_i + 1;
    end
    p(m_i) = t_i/ TheNumberOfAllUsers;
end
```

Algorithm 1. Calculation of prior probability

Calculation of conditional probability matrix. In order to calculate the conditional probability, first the joint probability is calculated and then the joint probability is turned into conditional probability. The algorithm 2 shows how we do the calculation.

```
foreach user in database do
    foreach item a in the user's known interests do
        foreach item b in the user's known interests do
            if a is not equal to b then
                t_{a,b} = t_{a,b} + 1;
            end
        end
    end
end
foreach item pair (a,b) do
    p(m_a, m_b) = t_{a,b}/ TheNumberOfAllUsers;
    p(m_a|m_b) = p(m_a, m_b)/p(m_b);
end
```

Algorithm 2. Calculation of conditional probability matrix

Making recommendation. Now we have the prior probability for each item and the conditional probability for each pair of items. The algorithm 3 will show how we make the recommendations.

How to compute c_n. As mentioned before, c_n is calculated by experiments. That is, the database is divided into different groups according to the size of user's known interest. For each group we use many c_n's to do the steps above and choose the one with the best result.

3.4 Computational Complexity

The offline computation, in which prior probability and conditional probability matrices are calculated, has a complexity of O(LM), where L is the length of log

foreach *user that needs recommendation* **do**
  foreach *item* x **do**
    $r(m_x) = p(m_x)$;
    foreach *item* u_i *in user's known interests* **do**
      $r(m_x) = r(m_x) \times (\frac{p(m_x|m_{u_i})}{p(m_x)})^{\frac{c_n}{n}}$;
    end
    $p(m_x|m_{u_1}, m_{u_2}, \cdots) = r(m_x)$;
  end
end

Algorithm 3. Making recommendation

in which each line represent an interest record of a user and M is the number of items. The online computation which gives the recommendation of all users, also has a complexity of O(LM). Therefore the total complexity is O(LM) only.

4 Experiment

Many recommendation algorithms are in use nowadays. We have non-personalized recommendation and k-NN recommendation mentioned before to be compared with our improved naive Bayesian.

4.1 Non-Personalized Recommendation

Non-Personalized recommendation is also called top-recommendation. It presents the most popular items to all users. If no relevancy is there between user's interests and the user, the Non-Personalized will be the best solution.

4.2 Data Set

The movie log from Douban.com is used in the experiment. It has been a non-public dataset up to now. The log includes 7,163,548 records of 714 items from 375,195 users. It is divided into matrix-training part and testing part. Each user's known interest of testing part is divided into two groups. One of them is considered known and is used to infer the other which is considered unknown. The Bayesian method ran for 264 seconds and the k-NN for 278 seconds. Both of the experiments are implemented in Python.

4.3 Evaluation

We have F-measure as our evaluation methodology. F-measure is the harmonic mean of precision and recall[3]. Precision is the number of correct recommendations divided by the number of all returned recommendations and recall is the number of correct recommendations divided by the number of all the known interests supposed to be discovered. A recommendation is considered correct if it is included in the group of interests which is set unknown. It is to be noted that the value of our experiment result shown later is doubled F-measure.

4.4 Comparison with Original Naive Bayesian Method

As it is shown in Figure 1, the improvement on naive Bayesian method has a fantastic effect. Before the improvement it is even worse than the non-personalized recommendation. After the improvement, naive Bayesian method's performance is obviously better than the non-personalized recommendation at any length of recommendation.

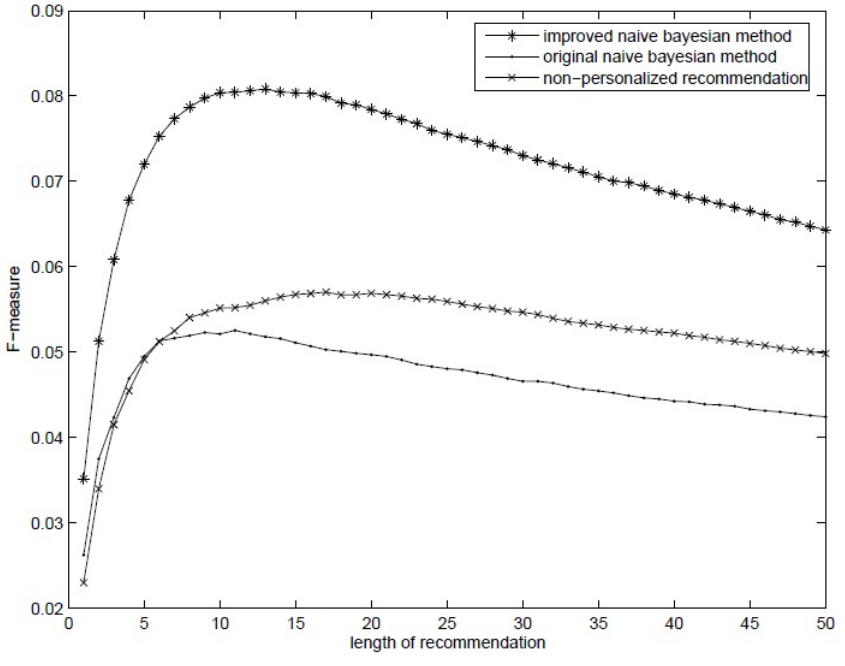


Fig. 1. comparison with original naive Bayesian method

4.5 Comparison with k-NN

As it is shown in Figure 2, before the peak k-NN and improved naive Bayesian method have almost the same performance. But when more recommendations are made, k-NN's performance declines rapidly. At the length larger than 45, k-NN is even worse than the non-personalized recommendation while improved naive Bayesian method still has a reasonable performance.

4.6 Analysis and Discussion

It is noticed that though there are great difference between different algorithms, the performances of all these algorithms turn out to have a peak. Moreover, the value of F-measure increases rapidly before the peak and decreases slowly after the peak. The reason for the rapid increase is that the recall rises and the precision is almost stable, while the reason for the slow decrease is that the precision reduces but the recall hardly increases.

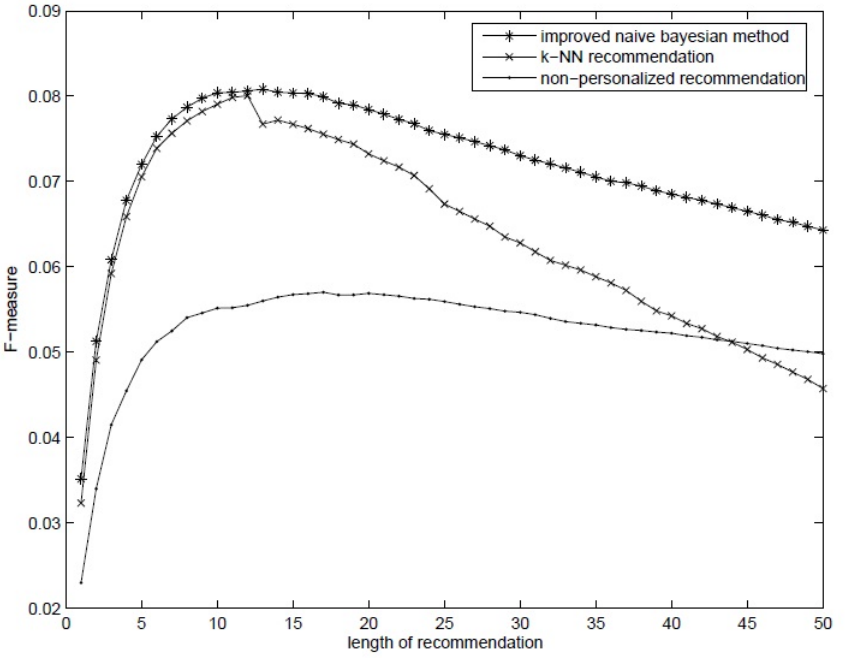


Fig. 2. Comparison with k-NN

According to our comparison between ordinary and improved naive Bayesian method, the improvement on naive Bayesian method has an excellent effect. The result of ordinary naive Bayesian method is even worse than that of non-personalized recommendation. However, after the improvement the performance is obviously better than the non-personalized recommendation. It is concluded that there is a strong relevance between user's known and unknown interests. The performance of non-personalized recommendation tells that the popular items are also very important to our recommendation. When a proper combination between two aspects is made, as it is in the improved naive Bayesian method, performance of the algorithm should be satisfactory. When the combination is not proper, it may lead to a terrible performance as it is shown in the ordinary naive Bayesian method.

The comparison of improved naive Bayesian method and k-NN shows that the improved naive Bayesian method has a better performance than the popular k-NN recommendation especially when it comes to long length recommendation. It is worth notice that the performance of two different algorithms are fairly close at short length recommendation, which leads to the conjecture that the best possible performance may have been approached though it calls for more proofs. Unlike short length recommendation, the performance of k-NN recommendation declines rapidly after the peak. It is even worse than the non-personalized recommendation at the length larger than 45. It is concluded that Bayesian method's good performance is because of its solid theory foundation and better

obedience of Vapnik's principle while k-NN's similarity definition may not be suitable for all the situations, which leads to the bad performance at long length recommendation.

5 Conclusion

In this article, we provide a new simple solution to the recommendation topic. According to our experiment, the improved naive Bayesian method has been proved able to be applied to instances where conditional independence assumption is not obeyed strictly. Our improvement on naive Bayesian method greatly improved the performance of the algorithm. The improved naive Bayesian method has shown its excellent performance especially at long length recommendation.

On the other hand, we are still wondering what the best possible performance of a recommendation system is and whether it has been approached in our experiment. The calculation of c_n is still not satisfactory. There may be a more acceptable way to get c_n, which is not by experiments. All of these call for our future work.

Acknowledgments. This work was supported by National Natural Science Foundation of China (NSFC), under Grant No. 60875080 and 60673020, and partially supported by the National High Technology Research and Development Program of China (863 Program), with Grant No. 2007AA01Z453. The authors would like to thank Douban.com for providing the experimental data, and Shoukun Wang for his stimulating discussions and helpful comments.

References

1. Adomavicius, G., Tuzhilin, A.: The next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering (2005)
2. Linden, G., Smith, B., York, J.: Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing (2003)
3. Makhoul, J., Kubala, F., Schwartz, R., Weischedel, R.: Performance measures for information extraction. In: Proceedings of Broadcast News Workshop 1999 (1999)
4. Breese, J.S., Heckerman, D., Kadie, C.: Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In: Proc. 14th Conf. Uncertainty in Artificial Intelligence (July 1998)
5. Hofmann, T.: Collaborative Filtering via Gaussian Probabilistic Latent Semantic Analysis. In: Proc. 26th Ann. Int'l ACM SIGIR Conf. (2003)
6. Kotsiantis, S.B., Zaharakis, I.D., Pintelas, P.E.: Machine learning: a review of classification and combining techniques. Artificial Intelligence Review (2006)
7. Yuxia, H., Ling, B.: A Bayesian network and analytic hierarchy process based personalized recommendations for tourist attractions over the Internet. Expert System With Applications (2009)
8. Resnick, P., Varian, H.R.: Recommender systems. Communications of the ACM (March 1997)

9. Koren, Y.: Factorization Meets the Neighborhood: a MultifacetedCollaborative Filtering Model. ACM, New York (2008)
10. Schafer, J.B., Konstan, J.A., Reidl, J.: E-Commerce Recommendation Applications. In: Data Mining and Knowledge Discovery. Kluwer Academic, Dordrecht (2001)
11. Pernkopf, F.: Bayesian network classifiers versus selective k-NN classifier. Pattern Recognition (January 2005)
12. Balabanovic, M., Shoham, Y.: Fab: Content-Based, Collaborative Recommendation. Comm. ACM (1997)
13. Rocchio, J.J.: Relevance Feedback in Information Retrieval. In: Salton, G. (ed.) SMART Retrieval System-Experiments in Automatic Document Processing, ch. 14. Prentice Hall, Englewood Cliffs (1979)
14. Pazzani, M., Billsus, D.: Learning and Revising User Profiles: The Identification of Interesting Web Sites. Machine Learning 27, 313–331 (1997)
15. Littlestone, N., Warmuth, M.: The Weighted Majority Algorithm. Information and Computation 108(2), 212–261 (1994)