Content-Based Recommendation in E-Commerce

Bing Xu[1], Mingmin Zhang[1], Zhigeng Pan[1,2], and Hongwei Yang[1]

[1] College of Computer Science, Zhejiang University,
310027 Hangzhou, P.R.China
[2] Institute of VR and Multimedia, HZIEE,
310037 Hangzhou, P.R.China
{xubin, zmm, zgpan, yanghongwei}@cad.zju.edu.cn

Abstract. Recommendation system is one of the most important techniques in some E-commerce systems such as virtual shopping mall. With the prosperity of E-commerce, more and more people are willing to perform Internet shopping, which resulted in an overwhelming array of products. Traditional similarity measure methods make the quality of recommendation system decreased dramatically in this situation. To address this issue, we present a novel method that combines the clustering which is based on apriori-knowledge and content-based technique to calculate the customer's nearest neighbor, and then provide the most appropriate products to meet his/her needs. Experimental results show efficiency of our method.

1 Introduction

E-commerce provides customers with ways to access necessary information without any restriction. Recommendation systems using all kinds of tools to respond to customer needs, understand customer behavior, and best use the limited available customer attention.

The nearest collaborative filtering recommendation system is one of the most successful techniques in E-commerce. It produces the recommendation for the customer based on the item rating of the nearest neighbor. With the prosperity of the E-commerce, the data of customer and item increasing dramatically resulted in the extreme sparsity of rating data. The traditional similarity measure methods have their deficiency in this situation. All of them cannot find the nearest neighbor accurately leads to the quality of recommender system decreasing dramatically. In this situation, our paper presents a novel method that combines the clustering which is based on apriori-knowledge and content-based technique to calculate the customer's nearest neighbor and then provide the most appropriate products to meet his/her needs. Our experimental results are showing that the method has a bright future.

The paper is organized as follows: in Section 2 we briefly review the previous work related to our research. In Section 3 deficiency of the traditional similarity measure in sparse dataset is analyzed. Content-based and clustering recommendation algorithm is described in Section 4. We show the experiment results in Section 5, and make concluding remarks in Section 6.

O. Gervasi et al. (Eds.): ICCSA 2005, LNCS 3481, pp. 946–955, 2005.

2 Related Work

Various learning approaches have been applied to construct customer profiles and to discover customer preferences to make recommendation.

The earliest approach used nearest-neighbor collaborative filtering algorithms [2][3]. Nearest neighbor algorithm is a rather lazy algorithm. A new customer is generally associated to a target customer, the algorithm chooses the nearest customers from computing the distances between different customers and then recommends the products to the customers.

Another method, the Bayesian networks create a model based on a training set with a decision tree at each node and edges representing customer information. The model can be built off-line very quickly, from a few hours to a few days. The resulting model is very economic, fast, and essentially as accurate as the nearest neighbor methods [4]. Identifying groups of customers appearing to have similar preferences also uses clustering techniques. In some cases, clustering techniques usually less accuracy than the nearest neighbor algorithms [4]. For this reason, clustering techniques can be applied as a "first step" of nearest neighbor algorithms.

Other methods such as classifiers are general computational models for assigning a category to an input. Classifiers have been quite successful in a variety of domains ranging from the identification of fraud and credit risks in financial transactions to medical diagnosis and intrusion detection. Association rules have also been used to analyze patterns of preference across products, and to recommend products to customers based on other products they have selected [1]. Horting is a graph-based technique in which nodes are customers, and edges between nodes indicate degree of similarity between two customers, this method searches for the nearest neighbor node in the graph, and then synthesizes ratings of the neighbor node to produce the recommendation [5].

Content-based recommendation approaches have also been applied to the basic problem of making accurate and efficient products recommendation in E-commerce. The text categorization methods adopted by Mooney and Roy [6] in their LIBRA system that makes content-based book recommendations exploiting the product descriptions found in Amazon.com, use a naive Bayes text classifier as in [7]. A reinforcement learning method is applied by Personal Web Watcher [8], a content-based system that recommends web-page hyperlinks by comparing them with a history of previous pages visited by the customer.

The new generation of web personalization recommender tools is attempting to incorporate techniques for pattern discovery in Web usage data. Web usage systems run a number of data mining algorithms on usage or click stream data gathered from web sites in order to discover customer profiles. A paper by Schafer [9] presents a detailed taxonomy and examples of recommender systems in E-commerce applications and how they can provide one-to-one personalization and capture customer loyalty at the same time.

3 The Deficiency of Traditional Similarity Measure

3.1 The Traditional Similarity Measure

Collaborative filtering recommender produces the target customer's recommended list according to other customer's rating. It accepts such assumption, if some customers have similar ratings on certain items; they have the same rating for other items. Collaborative filtering recommender system uses statistics method to search for the target customer's some nearest neighbors and then, according to the nearest neighbors' item rating, predicts the item rating of the target customer and produces the related recommender list.

To find the nearest neighbor of the target customer, we firstly use similarity measure method to calculate the similarity of the customers, and then select some nearest neighbors who have the closest similarity to the target customer. The accuracy of finding the nearest neighbor of the target customer affects the quality of the recommendation system directly and also plays the important role in the overall collaborative filtering algorithm.

The similarity measure methods include cosine-based similarity, correlation-based similarity and adjusted cosine similarity.

***Cosine-Based Similarity*:** In this case, two customers are thought of as two vectors in the dimensional item-space. If the customer does not rate an item, the item will be looked as zero. The similarity between them is measured by computing the cosine of the angle between these two vectors. Formally, in the $m \times n$ rating matrix, similarity between customer i and j, denoted by $sim(i, j)$ is given by $sim(i, j) = \cos(i, j) = \frac{\boldsymbol{i} \cdot \boldsymbol{j}}{\|\boldsymbol{i}\| \|\boldsymbol{j}\|}$ Where $\cdot$ denotes the dot-product of the two customers.

***Correlation-Based Similarity*:** In this case, similarity between two customers i and j is measured by computing the person r correlation $corr(i, j)$ lets the set of items which are both rated by customers i and j be denoted by I_{ij} , then the correlation similarity is given by

$$sim(i, j) = corr_{i,j} = \frac{\sum_{u \in U} \{R_{u,i} - \bar{R}_i\}\{R_{u,j} - \bar{R}_j\}}{\sqrt{\sum_{u \in U} \{R_{u,i} - \bar{R}_i\}^2} \sqrt{\sum_{u \in U} \{R_{u,j} - \bar{R}_j\}^2}} \tag{1}$$

Here $R_{u,i}$ denotes the rating of customer i on item u, $\bar{R}_i$ is the average of the $i - th$ customer's rating.

***Adjusted Cosine Similarity*:** The cosine-based similarity does not consider the rating scale between different customers. Adjusted cosine similarity adopts a method by subtracting the average rating of the customer to mend its deficiency. Lets the set of items that are both rated by customers i and j be denoted by I_{ij}; I_i and I_j are the item set rated by customer i and customer j separately. Then the correlation similarity is given by

$$sim(i, j) = \frac{\sum_{c \in I_{ij}} \{R_{i,c} - \bar{R}_i\}\{R_{j,c} - \bar{R}_j\}}{\sqrt{\sum_{c \in I_i} \{R_{i,c} - \bar{R}_i\}^2} \sqrt{\sum_{c \in I_j} \{R_{j,c} - \bar{R}_j\}^2}} \tag{2}$$

Here $R_{i,c}$ denotes the rating of customer i on item c, R_i is the average of the $i-th$ customer's rating.

3.2 Analysis of the Traditional Similarity Measure

In commercial recommendation system, many approaches based on nearest neighbor algorithm have been very successful. But the widespread use revealed some potential challenges, such as:

Sparsity: in practice, many commercial recommendation systems are used to evaluate large item sets in these systems, even active customers may have purchased well under 1% of the items. Accordingly, a recommendation system based nearest neighbor selected from the customers may be unable to make any item recommendations for a particular customer. As a result, the accuracy of recommendations may be poor.

Scalability: finding the nearest neighbor requires computation that shows with both the number of customers and the number of items. With millions of customers and items, a typical item-based recommender system will suffer serious scalability problems.

For example, in the cosine-based similarity method, the items that the customers do not evaluate are zero. Lets the rating item of the customer is denoted by R_{ij}. Then

$$R_{ij} = \begin{cases} r_{ij} & r_{ij} \neq \Phi \\ 0 & otherwise \end{cases} \tag{3}$$

Where r_{ij} is the rating of customer i on item j. If the customer i rates the item j, then R_{ij} equals to r_{ij} , else R_{ij} equals to zero.

This settlement can enhance computational performance, however, in the case of the extreme sparsity of the items and the greatness of the quantity, the reliability of the assumption is poor. Because, in practice, the preference of the customers is different for un-rating items and they cannot be the same rating, i.e., zero. Adjusted cosine similarity also has this problem.

In the method of correlation-based similarity, let u_i denote the item set rated by the customer i, u_j is the item set rated by the customer j. The intersection of items rated both by customer i and customer j is $u_i \Lambda u_j$.

In common sense, they can only get the higher similarity when there exists many items whose ratings are very adjacent for the two customers. When the rating items are very sparse, the item set that both rated by the two customers are very small, only one or two items. In this situation, even the two customers have very high similarity; we cannot say they are similar actually. This method also has some deficiency.

From the above, we can say that the traditional similarity measure cannot measure the similarity between the customers effectively when the rating data are extremely sparse. This resulted in the inaccurate neighbors and the decrease of the recommender accuracy.

4 Content-Based and Clustering Recommendation Algorithm

The weakness of the similarity measure for large and sparse database leads us to explore alternative recommending algorithm. One simple method is to set up the unevaluated items as a constant, and the middle rating is often used in general. The experiment shows that this modified method can improve the quality of the recommendation system. However, it is impossible that the unrated items have the same value, so the modified method cannot solve the traditional similarity measure for sparse database radically. Therefore, we put forward a novel method. Firstly, split the unvalued items into several parts using the apriori-knowledge clustering, and then predict the unvalued items, finally, find out the nearest neighbors using these rating items and accomplish the recommendation. This method can let unvalued items get a reasonable value and accordingly it can provide more rating items for the nearest neighbor algorithm.

In the following, we introduce content-based collaborative filtering recommendation algorithm in detail. The algorithm is divided into two steps: find the nearest neighbor and process recommendation.

4.1 Obtaining Nearest Neighbor

We must find the union of the rating items before calculating the similarity between the customers. In the union, the unvalued items are predicted through the apriori-knowledge clustering method and then the neighbors can be found by calculating the rating items belong to the union. This method can not only deal with the problem that the unvalued items are equal to a constant in cosine-based similarity effectively but also treat with the shortcoming that the workable rating items are very few in correlation-based similarity, which resulted in obtaining the exactness of the nearest neighbor and improving the quality of recommendation.

In the traditional clustering method, there is no supervise nor some apriori-knowledge. The clustering result is laid on the given data and the certain selected clustering algorithm, which to a degree resulted in the poor result. The reason is that unclassified data do not offer any information in the process of the clustering. In practice, there are some known knowledge in the related domain, and we can obtain some classified examples known as apriori-knowledge. The knowledge can be used in the process of the clustering to guide clustering which will increase the accuracy of the clustering.

The algorithm of Clustering based on apriori-knowledge is show as below:

Step 1: each example whose classification has been known can be as the single element for each s_i and construct the initial set $\{s_i\}$, $1 \cdots n$.
Step 2: calculate the similarity of every two sets.

$$\frac{1}{|C_l| \bullet |C_k|} \sum_{x_l \in c_l}^{x_k \in c_k} s(x_l, x_k) \tag{4}$$

Where c_l, c_k are two different classification set.

Step 3: suppose s_m and s_n are the two sets that have the greatest similarity. If the examples in c_m and c_n belong to the same class, we will unit the two sets and go to step 2. If not, we turn to step 4.

Step 4: output the k value.

Step 5: use the k value and select $c_1, c_2 \cdots c_k$ randomly from the dataset as the initial training factor, then use SOM algorithm [13] to obtain the clustering result.

For example, suppose $E = \{e_1, e_2 \cdots e_{15}\}$ are composed of two classification. $c_1 = \{e_1, e_2, e_3, e_4, e_5, e_{12}, e_{13}, e_{14}, e_{15}\}$, $c_2 = \{e_6, e_7, e_8, e_9, e_{10}, e_{11}\}$. Here, $\{c_1, c_2\}$ represent two classifications individually. Table 1 shows the dataset.

Table 1. The Dataset

Attribute Example	Attr1	Attr2	Attr3	Attr4	Attr5	Attr6	Attr7	Attr8	Attr9	Attr10
1	1	1	1	0	1	0	0	0	1	0
2	1	1	1	0	0	1	0	0	1	0
3	1	1	1	1	1	0	0	0	1	0
4	1	1	1	0	0	0	0	0	1	0
5	1	1	1	1	0	0	0	0	1	0
6	0	1	0	1	1	1	0	1	0	1
7	0	0	1	1	1	1	0	1	1	0
8	0	1	0	1	1	1	0	1	0	1
9	0	1	0	1	1	1	0	1	0	1
10	0	0	1	1	1	1	0	1	0	1
11	1	0	0	1	1	1	0	1	0	1
12	0	0	0	1	0	0	1	0	1	0
13	0	0	0	1	0	1	1	0	1	0
14	0	0	0	1	1	1	1	0	1	0
15	0	0	0	1	1	0	1	0	1	0

Assume the examples $\{e_1, e_2, e_9, e_{15}\}$ have known their attributes and classifications, others only know: their attributes. The description of the algorithm is as follows:

(1) In the $\{e_1, e_2, e_9, e_{15}\}$, each example can be looked as an independent subset, $s_1 = \{e_1\}, s_2 = \{e_2\}, s_3 = \{e_9\}, s_4 = \{e_{15}\}$;

(2) Calculate each subset's average similarity according to Equation (1), we can conclude that the similarity between and are the most.

Then, estimate whether they belong to the same class or not.

If yes, we will unit s_1 and s_2 and produce new subset $s_1 = \{e_1, e_2\}, s_2 = \{e_9\}, s_3 = \{e_{15}\}$. Repeat this step until the subset cannot unit.

If not, go to step (3).

(3) Output the result $s_1 = \{e_1, e_2\}, s_2 = \{e_9\}, s_3 = \{e_{15}\}$.

(4) Select the original centers randomly: e_2, e_4, e_{10}.

Use the self-organizing map to calculate the clustering results:
$s_1 = \{e_1, e_2, e_3, e_4, e_5\}, s_2 = \{e_6, e_7, e_8, e_9, e_{10}, e_{11}\}, s_3 = \{e_{12}, e_{13}, e_{14}, e_{15}\}$
The clustering centers are: $s_1 = \{1, 1, 1, 0, 0, 0, 0, 0, 1, 0\}$,
$s_2 = \{0, x, 0, 1, 1, 1, 0, 1, 0, 1\}, s_3 = \{0, 0, 0, 1, x, x, 1, 0, 1, 0\}$; Where x represent any value belongs to $\{0, 1\}$.

Then estimate the class for each subset. Examples e_1, e_2 belong to subset s_1 and the ex-ample e_9 belongs to s_2. From step (3) and step (4), we conclude that $\{e_3, e_4, e_5, e_{12}, e_{13}, e_{14}\}$ belong to class c_1, the example e_{15} belongs to s_3, so $\{e_6, e_7, e_8, e_{10}, e_{11}\}$ belong to class c_2.

Here, we use UCI Machine Learning database to verify the algorithm. Table 2 shows the result. Note that, in the column of selected examples, 2*5 express that

Table 2. The Experiment Result

Database name	Attribute character	Example numbers	Class number	Selected examples	Accuracy1	Accuracy2
Voting-record	symbol	435	2	2*5	89.4%	86.2%
Zoo	symbol	101	7	7*2	94.8%	90%
Soybean-large	symbol	683	19	19*2	75.1%	62.5%
Thyroid-disease	symbol, numeric	3772	3	3*10	90.0%	74.8%
Iris	numeric	150	3	3*3	91.1%	87.7%

voting-record database has two classes. In each class, we select 5 examples as the known classification examples. Column 6 showed the accuracy obtained by using our method and column 7 showed the accuracy obtained by ordinary clustering methods. From Table 1, we can see that amendatory clustering techniques can acquire better results.

The further research work about the relationship between classified data, the precision of the clustering algorithm and the number of the clustering center are showed in [10].

The unvalued items can get a rating after the above algorithm. The cosine-based similarity or correlation-based similarity is then used in the union to calculate the similarity of the customers.

4.2 Producing Recommendation

After the neighbors acquired, $P_{u,i}$ is the final rating of the customer u to product i, which can be obtained from the product i by the neighbors. The equation is as follows [4]:

$$P_{u,i} = \bar{R}_u + \frac{\Sigma_{n \in neighbor} sim(u, n) \times (R_{n,i} - \bar{R}_n)}{\Sigma_{n \in neighbor}(|sim(u, n)|)} \tag{5}$$

Where $sim(u, n)$ is the similarity measure between customer u and n, and where $R_{n,i}$ is the rating of customer n to product i. $\bar{R}_u$ is the average rating of customer n, so $\bar{R}_u$ denotes the average rating of customer u.

5 Experiment Results and Analysis

Here, the experiment dataset comes from [12]. There is a brief description of the data. This data set consists of:

* 100,000 ratings (1-5) from 943 customers on 1682 movies.

* Each customer has rated at least 20 movies.

* Simple demographic info for the customers (age, gender, occupation, zip) the most important files in the data set are:

u.data – The full u data set, 100000 ratings by 943 customers on 1682 items.

u.genre – A list of the genres.

u.item – Information about the items (movies); The last 19 fields are the genres, a 1 indicates the movie is of that genre, a 0 indicates it is not; movies can be in several genres at once.

In the experiment, items whose rating are 1 can be seen as a class, items whose rating are 2 or 3 can be seen as another class, items whose rating are 4 or 5 can be a class too. Using the u.item files to predict the items for the unevaluated items based on the apriori-knowledge clustering algorithm and then use u.data to implement the similarity of the customers as well as the products recommendation.

5.1 Experiment Results

We adopt MAE (Mean Absolute Error) as the evaluated standard.

The definition is:

$$MAE = \frac{\Sigma_{i=1}^{N} |p_i - q_i|}{N} \tag{6}$$

Where p_i represents the degree of satisfaction that the customer assess the product, q_i represents the degree of the satisfaction that recommendation algorithm assess the product, and N represents the total customers. MAE represents the mean absolute error between the real ratings items and the predicable rating items. The more decreased the MAE is, the more the quality of recommendation is increased.

According to [11], cosine-based similarity method can obtain better result than correlation-based similarity method. So, the cosine-based similarity method is chosen as the measure for our experiment.

The experiment result is shows in Fig.1.

5.2 Analysis of the Experiments

The most difference between the traditional collaborative filtering and the combination of content-based and the apriori-knowledge is how to get the validate neighbors. In the traditional collaborative filtering recommender algorithms, the correlation-based similarity method only employs the rating items that the customers are both rated. In the sparse dataset, since the intersection contains few items, so the nearest neighbor calculated may not be the actual neighbor. The experiment results also show that the quality of the recommendation based on the

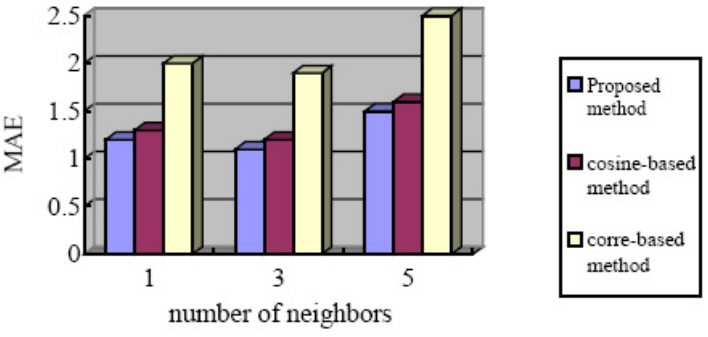

Fig. 1. The comparison of accuracy of recommendation algorithms

correlation-based similarity method is poor. Another method, cosine-based similarity method employs the union of the rating items. In the union, the un-valued items are endued with the same value. Though the quality of the recommendation are improved, the unvalued items have the same value are unreasonable. A content-based clustering method uses the apriori-knowledge clustering to rate the unevaluated items and then find out the nearest neighbor of the target customer. The experiment results show this method has the highest accuracy of the recommendation.

6 Conclusion

This paper begins with the analysis of the deficiency in the traditional similarity measure method for the greatly sparse rating data. To deal with the problem, content-based recommendation method is proposed. This method uses the characters of the products as well as the rated products that can be regarded as the classified data. These apriori-knowledge can be used to supervise the clustering and accordingly the rating of the unevaluated products can be predicted. This method can solve the problem that exists in traditional similarity measure method effectively and get the more accurate neighbors of the target customers. The experiment shows that our method can greatly increase the quality of the recommendation.

Acknowledgements

This research work is supported by 973 project (grant no: 2002CB312100), and TRAPOYT Program in Higher Education Institution of MOE, PRC.

References

1. Lee, C.-H., Kim, Y.-H., Rhee, P.-K. Web Personalization expert with combining collaborative filtering and association rule mining technique. Expert Systems with Applications (2001) 21(3) 131-137

2. Resnick, P.,Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. Grouplens: An open architecture for collaborative filtering of netnews. In Proceedings of ACM CSCW'94 Conference on Computer-Supported Cooperative Work, (1994) 175-186
3. Shardanand, U. and Maes, P. Social information filtering: Algorithms for automating "word of mouth". In Proceedings of ACM CHI'95 Conference on Human Fac-tors in Computing Systems, (1995) 210-217
4. Breese, J., Heckerman, D., and Kadie, C. Empirical analysis of predictive algorithms for collaborative filtering. In Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI-98), (1998) 43-52
5. Wolf, J., Aggarwal, C., Wu, K-L., and Yu, P. Horting Hatches an Egg: A New Graph-Theoretic Approach to Collaborative Filtering. In Proceedings of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, San Diego, CA. (1999) 201-212
6. Mooney, R. J. and Roy, L. Content-based book recommending using learning for text categorization, Proceedings of the VACM Conference on Digital Libraries, San Antonio, USA, (2000) 195-204
7. Abbattista, F., Degemmis, M., Fanizzi, N., Licchelli, O., Lopes, P., Semeraro, G., Zambetta F. Learning customer profiles for content-based filtering in e-commerce. Italian Artificial Intelligence Conference. (2002)
8. Thorsten Joachims, Dayne Freitag, Tom Mitchell. WebWatcher: A tour guide for the world wide web, Proceedings of the XV International Joint Conference on Ar-tificial Intelligence, Nagoya, Japan (1997) 770-775
9. [9] Schafer J. B., Konstan J. Electronic commerce recommender applications, Journal of Data Mining and Knowledge Discovery, (2001) vol. 5 num. 1-2. 115-152
10. WANG XINGQI the doctor thesis. Research on algorithm of machine learning and its application, Zhejiang University.(2002)
11. Deng Ai-lin, Zhu Yang-yong, Shi Bai-le. A collaborative filtering recommendation algorithm based on item rating prediction. Journal of Software. (2003) 14(9): 1621-1628
12. http://www.research.compaq.com/src/eachmovie
13. Kohonen, T. The self-organizing map. Proceedings of the IEEE, (2001) 78(9):1464-1480