2017 10th International Symposium on Computational Intelligence and Design

Collaborative Filtering Recommender Algorithm Based on Comments and Score

Yuanqing Zhu[1], Wei Song[1], Lizhen Liu[1,*] ,Xinlei Zhao[2], Chao Du[1]
[1]Information Engineering College, Capital Normal University, Beijing, China
[2]College of Foreign Languages, Capital Normal University, Beijing, China
Beijing, China
Email: yqzhu@cnu.edu.cn, lzliu_cnu@sina.com

***Abstract*-Accompanied by rapid development of Internet technology, people are increasingly dependent on the network. In the past, people is usually passive to accept information, but today people begin to take the initiative to create information. This case makes network data more and more. In order to ease information overload caused by inconvenience, recommender system has gradually been people's attention. Through the appropriate recommended technology, it can help people to filter out useless content, reduce the amount of information faced by individuals. In the recommender system, collaborative filtering recommendation is a more widely used method. In order to improve the recommended results, this paper based on the traditional method of rating. Using user's comments, emotional analysis tools to extract the emotional polarity of comments, combine emotional polarity and rating to enhance final recommended effect.**

Keywords-collaborative filtering; emotional analysis; k nearest neighbor recommendation;

I. INTRODUCTION

With the development of Internet technology, massive data presents an explosive growth trend. Accompanied by acceleration of the information process, the information overload problem which people are experiencing become more and more obvious. According to China Internet Network Information Center's 39th *Report on China's Internet Development in Statistics*, the report's points out that China's internet users reached 731 million people, internet penetration reached 53.2% until 2016, December [1]. Those massive data contain a large number of undiscovered information. In this context, data mining technology was born. Recommender system is a practical research direction in the field of data mining. The recommender system is a kind of information filtering systems, it used to predict a user's rating or preference for an item [2]. Now, almost all of the E-commerce platform will use recommended technology to assist their own business areas. This technology can bring business platform better earning, also help people solve the problem of information overload for individual users in a certain extent. Common recommender systems compose many types, such like based on user-based collaborative filtering recommendations, item-based collaborative filtering recommendations, content-based recommendation, mixed recommendation and so on. Business platform according to their own needs and then select the appropriate method to achieve the best recommend results.

In recommender system, a very important element is user's historical data. In the case of shopping data, the user history data contains items that purchased by this user, items that viewed or collected, and ratings or evaluations of the items. The current mainstream recommended method is to use user's history score to make user profile, thereby predicting user's behavior. This paper hopes to combine user's comments and scoring of the items in order to make more accurately prediction according to user's preferences. In order to describe the user's preferences, author use the Python TextBlob library (Steven Loria, 2016) to calculate emotional polarity of the comment. Then calculate the weight of comments and ratings to describe user's preferences.

II. RELATED WORK

Existing recommender systems are mainly classify into two categories, collaborative filtering recommendations and content-based recommendations. Collaborative filtering recommendation is mainly to use the characteristics of the user group and predict the current user's preferences, thus complete the recommended task. When the system incorporates the specific attributes of the item, content-based recommendation is generating.

Collaborative filtering recommended is the current use of a more extensive and mature recommendation algorithm. [3] Chen et.al [4] creates a general neural network-based recommendation framework. Brovman et.al [5] focus on the problem of recommendations in eBay's data and builds a recommender system.

III. EXPERIMENTAL DATA

In this paper, the author used amazon products data which provided by Julian McAuley of University of California San Diego. The dataset includes reviews (ratings, text, helpfulness votes), product metadata (descriptions, category information, price, brand, and image features), and links (also viewed/also bought graphs).

IV. TRADITIONAL METHOD

Mainstream recommendation methods include user-based recommendations, item-based recommendations, content-based recommendations and mixed recommendations [6]. In

2473-3547/17 $31.00 © 2017 IEEE
DOI 10.1109/ISCID.2017.158

business cases, such as Amazon's e-commerce system, Pandora's music recommender system, Douban book recommender system. These platforms choose the most appropriate recommended method according to the different kinds of products. Usually, the item-based recommendation is a more widely recommended method.

A. User-based collaborative filtering recommendation

User-based recommendation is an early recommendation put forward by GroupLens institute [7]. Its thoughts are giving some data of one user to system, the system use some method to find other users who seems like this user, so other's interest can be reference of recommendation. In practical application, system normally finds k similar users to be the source of data. This method also called Collaborative Filtering Recommendation Based on k Nearest Neighbor [8].

1) Relevancy of users

In order to find similar users in a large number of user data, system needs to think of a way to measure whether two users have similar preferences. The usually approach is to calculate the Manhattan distance or the Euclidean distance of two users. However in practical application, user's preferences have a strong subjectivity, it is difficult to reflect the user's true intent with simply calculating the distance. In actual operation, we usually use the Pearson product-moment correlation coefficient and cosine similarity to measure distance.

a) Pearson product-moment correlation coefficient

In the scoring data, it is possible to produce the following: two users made same ratings, but because of their different evaluation standard, their real opinions are not same. This is the common problem of "fractional devaluation" in the field of data mining. In order to reduce the impact of different evaluation standard, system can use Pearson correlation coefficients to measure the similarity of the two variables. The value of the Pearson correlation coefficient is [-1, 1], while -1 means completely inconsistent, 1 means completely consistent. The Pearson correlation coefficient does not take into account the difference in mean value, so the correlation between the scores is more obvious. The formula is as follows:

$$\mathrm{r} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{1}$$

where x_i and y_i means the rating of user x and user y. $\bar{x}$ and $\bar{y}$ means the average rating of user x and user y. 'n' means the quantity of items which two users had made rating.

b) Cosine similarity

In the user-based recommendation, the commonly method used in similarity measure method also includes cosine similarity. The traditional method to calculate the result of the distance may meet problems of data sparseness. When system calculates the result on data set, ratings made by two users may be null. To solve the problem, the cosine similarity ignores this zero-value match. When system use this method to calculate similarity, we can get more accurately describe the preferences of two users. The cosine similarity is calculates as follows:

$$\cos(x, y) = \frac{x \cdot y}{\|x\| \cdot \|y\|} \tag{2}$$

where '$\cdot$' means Inner product, $\|x\|$ means length of vector x. Cosine similarity range of values [-1, 1], while -1 means completely inconsistent, 1 means completely consistent.

2) Nearest neighbor recommendation

Through the above methods, you can calculate similarity between one user and all other users' similarity value. One common practice is sort the users by result in descending order of similarity, choose former k users to be similar users. Use those users' interests to predict one user's interest. The formula of predict user 'a' to item 'p' is as follows:

$$pred(a, p) = \bar{r}_a + \frac{\sum_{b \in K} sim(a, b) \times (r_{b,p} - \bar{r}_b)}{\sum_{b \in K} sim(a, b)} \tag{3}$$

where sim(a, b) is similarity, $\bar{r}_a$ is average rating of user 'a'. As for how to choose k, in the work of Herlocker et al. (2002) can find other formula about this question.

B. Item-based collaborative filtering recommendation

User-based recommendation method have been used in many areas, but when the quantity of user data and item data increase fast, problem of data sparseness and scalability of system had been a serious challenge to system. User-based recommendation is also referred to as memory-based recommendation, because the system needs to save all the scoring results to recommend. Therefore, it is possible to building a model to represent the degree of similarity between items, instead of just saving all the rating results.

1) Adjusted cosine similarity:

To solve the problem of data sparseness, in this method, the average rating result is subtract for each rating of the user, thus product an improved cosine similarity method. In order to offset the consequences of fractional devaluation, the average rating of the user will be subtract for each rating result at the time of the calculation. The adjusted cosine similarity is as follows:

$$s(i, j) = \frac{\sum_{u \in U}(R_{u,i} - \overline{R}_u)(R_{u,j} - \overline{R}_u)}{\sqrt{\sum_{u \in U}(R_{u,i} - \overline{R}_u)^2}\sqrt{\sum_{u \in U}(R_{u,j} - \overline{R}_u)^2}} \tag{4}$$

where $R_{u,i}$ means user 'u' scoring to item 'i', U means the set of users which scored both 'i' and 'j'.

2) Forecast score results:

Because the user's range of items is [1, 5], so the system needs to normalize ratings to [-1, 1]. Then we can get the formula of predict user u scores item i:

$$P(u,i) = \frac{\sum_{N \in SimiliarTo(i)} S_{i,N} \times NR_{u,N}}{\sum_{N \in SimiliarTo(i)} (|S_{i,N}|)} \quad (5)$$

Where $S_{i,N}$ is adjusted cosine similarity, $NR_{u,N}$ is normalized rating.

V. Improved: Combined with Emotional Analysis

In the real life, product would have score and evaluation. These descriptions used to describe whether the product get user's attention. However, different user has different evaluation standard in the process of user evaluation. Some users tend to give higher score, but some users are accustomed to give more strict evaluation. Merely using rating cannot fully reflect the user's preferences. If we combine with natural language evaluation, we can describe user's interest more accurate. This section will consider a method that combine emotion and score in order to measure user intention.

A. Emotion analysis tools

In this work, we use TextBlob packet to analysis emotion. TextBlob is a python library for processing textual data. It provides a simple API for diving into common natural language processing tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more. In this paper, we cut up data set for each type. About 90% of these extracted as training data, and 10% of the data used as test data.

B. Parameter training

In the data set, each user has a score and evaluation of some products. In order to better the user comments, which contained in the combination of information and user ratings, we assume x_1 and x_2. Where x_1 means rating and x_2 means emotional polarity calculated by TextBlob tools, y means user interest value. These three variables satisfy the following relationship:

$$y = w_i \times x_1 + w_j \times x_2 \quad (6)$$

where w_i and w_j are weights parameter of two variables in this system. The system hopes to find a method to get w_i and w_j to reflect user's real preference. For each user's data in this system, we randomly set one user evaluation record as a benchmark, set the user's remaining evaluation records as training data for calculating the parameters. Let the two weights add equal to one at the same time. For each user, the system calculates suitable weight value for making the current data nearest fit the benchmark value. After calculating all the weight values on the training sample, the system calculated the average value of weights. When the system calculating the most appropriate weight, since summation of the two weights are equal to one at the same time, one parameter incremented from 0 at a given step size, the other parameter decreased from 1 at the given step. After the system calculated predicted score according to the current user's preference value, if the score is closest to the user's scoring, that means the value of weight at this time is the most appropriate value. In other words, the goal of this part of the system is to calculate the appropriate weights make which satisfy the follows:

$$\min(pred_{u,i}(y) - rating_{u,i}) \quad (7)$$

where $pred_{u,i}(y)$ means score prediction with user interest value, $rating_{u,i}$ means real rating of the user. When the difference is minimum, system record the current weight value.

VI. Experiment

MAE(Mean Absolute Error) is a common but effective evaluation method in the recommender system. The performance of the system is measures by calculating the difference between the predicted score and the gold data. The formula of MAE is as follows:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |f_i - y_i| \quad (8)$$

where f_i is the value of formula, y_i is gold data.

For each different data set, there are not less than five evaluation records for each user, including the user identifier, the item identifier, the user rating, and the user's comments on the product. Randomly select a user's record as a benchmark, according to his remaining data according to the method of the text to predict possible scoring. The function calculate prediction and rating of the MAE to assess whether this current calculation is accurate. To measure the effect on the current dataset, the system took average of the resulting MAE values and got the following table I.

TABLE I. MAE values of Rating and polarity

Data Set	Rating	Polarity
Pet_Supplies	0.8736	0.1593
Automotive	0.8975	0.1625
Clothing_Shoes_and_Jewelry	0.8533	0.1645
Baby	0.9330	0.14722
Tools_and_Home_Improvement	0.8663	0.1470
Toys_and_Games	0.8825	0.1487
Patio_Lawn_and_Garden	0.9715	0.12462

In order to make better use of the user's intention embodied in the comment, the system respectively gives the original polarity and evaluations different weights, ensuring that both weights are add to one at the same time. Thus, a weight increases from zero, while the other weight decreases from one. Limited by the hardware performance, the step of weight's change was set for 0.1. As result, when weight of rating is set for 0.7 and weight of polarity is set for 0.3, the system can get best performance.

Acknowledgment

This work was supported in part by National Science Foundation of China under Grants No. 61402304, the Humanity & Social Science general project of Ministry of Education under Grants No.14YJAZH046, the Beijing Natural Science Foundation under Grants No. 4154065, the Beijing Educational Committee Science and Technology Development Planned under Grants No. KM201610028015, Beijing Science and technology innovation platform, Teaching teacher, and Connotation Development of Colleges and Universities.

References

[1] The39th，Report on China's Internet development in statistics [J]. htp://www. cnnic. net. cn, 2016.

[2] Ricci F, Rokach L, Shapira B. Introduction to recommender systems handbook[M]//Recommender systems handbook. springer US, 2011: 1-35.

[3] Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms[C]//Proceedings of the 10th international conference on World Wide Web. ACM, 2001: 285-295.

[4] Chen T, Sun Y, Shi Y, et al. On Sampling Strategies for Neural Network-based Collaborative Filtering[C]//Proceedings of the 23th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM. 2017.

[5] Brovman Y M, Jacob M, Srinivasan N, et al. Optimizing similar item recommendations in a semi-structured marketplace to maximize conversion[C]//Proceedings of the 10th ACM Conference on Recommender Systems. ACM, 2016: 199-202.

[6] Jafarkarimi H, Sim A T H, Saadatdoost R. A naive recommendation model for large databases[J]. International Journal of Information and Education Technology, 2012, 2(3): 216.

[7] P. Resnick, N. Iacovou, etc. "GroupLens: An Open Architecture for Collaborative Filtering of Netnews", Proceedings of ACM Conference on Computer Supported Cooperative Work, CSCW 1994. pp.175-186

[8] Bijalwan V, Kumar V, Kumari P, et al. KNN based machine learning approach for text and document mining[J]. International Journal of Database Theory and Application, 2014, 7(1): 61-70.