2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC 2020)

# A web service recommendation algorithm based on BaisSVD

Da Sun ,Tong Nie
Beijing Engineering Research Center for IoT Software and Systems, Information Department, Beijing University of Technology

***Abstract*—With the development of network technology and network services, the number of Web services will also be exploded, so the magnitude of Web services that can provide similar functions will gradually be increased. Quality of service (QoS) is widely used to describe and evaluate the non functional properties of Web services, and has been successfully applied to service recommendation. However, most of the current Web Service recommendations are still limited to traditional collaborative filtering, without considering the bias information of users and web services themselves. In this thesis, firstly, the author introduces the current situation of web services recommendation, then analyzes the existing problems and the characteristics of the data sets. Finally, the author applies the BiasSVD to predict QoS, and then recommends to users. By comparing the experimental results with the traditional collaborative filtering algorithm, the feasibility and superiority of the algorithm in Web services recommendation scenarios are obtained.**

*Keywords— Web service; recommendation; QoS prediction; collaborative filtering; Matrix factorization*

## I. INTRODUCTION

With the development of Internet technology, the convenience of Web services lead to the increasing demand of users for it. Nowadays, a large number of Web services are flooding the Internet, and users' choices are increasing. At the same time, they also put forward higher requirements: in the same service types and conditions, besides meeting the functional requirements, users want to get higher quality of service (QoS). At present, it has become a way to solve the information overload of Web services by predicting the service quality[1][2] and then recommending to users[3][4]. Collaborative filtering is a widely used recommendation technology which has been proved to be the most accurate prediction method[5][6][7]. Classical collaborative filtering based on historical data is mainly studied from three aspects. The first is from the perspective of users, that is user-based algorithm(UPCC) . The second one is from the perspective of items, that is item-based algorithm (IPCC) . The third combines the two aspects of user and projects,that is the algorithm based on user and item (UIPCC). UPCC is based on the premise that in the existing data, the higher the similarity between the two users, the greater the likelihood of similar results for those users who have no access to the project. IPCC is from the perspective of the project to find similar projects to predict. UIPCC is a weighted result of both UPCC and IPCC. In general, the similarity between users and projects can be obtained by cosine similarity or Pearson correlation coefficient. However, the traditional collaborative filtering web services recommendation algorithm also has some shortcomings:

If there is a correlation between items, information content does not increase linearly with the increase of vector dimension;

The score matrix elements are sparse, the calculation results are not stable, and the increase or decrease of a vector dimension will lead to a large difference in the results of near-neighbors. Based on these problems, this thesis uses BiasSVD to solve them, so as to improve the accuracy of recommendation results and the user experience .

## II. RELATED WORK

### *A. Traditional collaborative filtering*

At present, there are two kinds of collaborative filtering recommendation algorithms: user-based collaborative filtering recommendation algorithm and project-based collaborative filtering recommendation algorithm. The former is based on the assumption that if users have similar scores for some items, they have similar scores for other items, based on the nearest-neighbors (the most similar users) of the target user, the algorithm approximates the target user's score of a project. As for the latter, users' ratings for different projects are similar, to estimate the user's rating for a certain item, you can estimate the rating for several similar items of the user's item.

The key step in collaborative filtering is to calculate the similarity between users. Many methods have been used to calculate the similarity between users in collaborative filtering algorithm. Most of them are based on users' ratings of products with common preferences. The two most commonly used methods are Pearson correlation[8][9] and included angle cosine, both of which define the product set that users X and Y overplay as: $S_{xy} = S_x \cap S_y$ . The Pearson correlation between users X and Y is defined as:

$$\mathrm{sim}(x,y) = \frac{\sum_{s\in S_{xy}} (r_{x,s}-\bar{r}_x)(r_{y,s}-\bar{r}_y)}{\sqrt{\sum_{s\in S_{xy}} (r_{x,s}-\bar{r}_x)^2 \sum_{s\in S_{xy}} (r_{y,s}-\bar{r}_y)^2}} \tag{1}$$

In the included angle cosine method, users X and Y are represented by m-dimension vectors. The similarity between the two vectors can be obtained by :

$$\mathrm{sim}(x,y) = \cos(\vec{x},\vec{y}) = \frac{\sum_{s\in S_{xy}} r_{x,s} r_{y,s}}{\sqrt{\sum_{s\in S_{xy}} r_{x,s}^2 \sum_{s\in S_{xy}} r_{y,s}^2}} \tag{2}$$

The user-based method utilizes the historical QoS experiences from similar users for personalized QoS prediction, while the service-based method uses those from similar services for prediction. The hybrid method is the combination of the previous two methods, so it can achieve higher prediction accuracy.

Based on the traditional collaborative filtering algorithm, the basic steps for both UPCC and IPCC to predict and recommend QoS are as follows:

(1).Prepare data set;

(2).Calculate the similarity of users (Web services) and get the similarity of users (Web services);

(3).Select the nearest neighbor user (Web Service) to get the set of similar users (Web Service);

(4).QoS prediction and get the prediction results;

(5).Recommend.

Different systems use different similarity calculation to make the prediction score as accurate as possible. Although collaborative filtering recommendation system has been widely used, it also faces many problems, for example, poor recommendation effect and weak extension in the system.

*B. Matrix factorization in Recommendation System*

The model-based collaborative filtering algorithm is generally based on the defined random parameter model. According to the existing QoS data set and through machine learning method, a prediction model is trained to predict the personalized QoS value.The score prediction problem of recommendation system can be regarded as a game of matrix completion, which is the task of recommendation system, and matrix decomposition is to achieve its goal. The core idea of matrix decomposition algorithm applied in personalized recommendation is to decompose the user rating matrix into a matrix of low rank, so that the product is as close to the original rating matrix as possible, and the error square between the predicted matrix and the original matrix is minimized. Singular value decomposition (SVD) is widely used in machine learning because it can be used both in feature decomposition of dimension reduction algorithm and recommendation algorithm. We regard the scores of m users and n items as a matrix M, and then use the matrix decomposition to solve the recommendation problem.

***1).FunkSVD***

PureSVD needs to fill in the matrix first, then decompose and reduce the dimension, because of inverse operation (O (n ^ 3)), has the shortcoming of high complexity. Therefore, Simon funk proposed the method of FunkSVD. Instead of decomposing the matrix into three matrices, he decomposed it into two low rank user project matrices, which reduced the computational complexity.

$$M_{mxn} = P^{T}_{mxk} Q_{kxn} \tag{3}$$

For a user score, FunkSVD is used for matrix decomposition, and the corresponding representation is $q_j^T p_i$. FunkSVD uses the idea of linear regression to find the optimal implicit vector representation of users and projects by minimizing the square of observation data. If we consider a combination of all items and samples, minimize the following formula：

$$\sum_{i,j} (m_{ij} - q_j^T p_i)^2 \tag{4}$$

If formula (3) is minimized and the corresponding extreme value $p_i$, $q_j$ is found, then matrix P and Q will be obtained. For any blank score of matrix M, scores can be calculated by $q_j^T p_i$. Meanwhile, in order to avoid over fitting observe data, FunkSVD with L2 regular term is proposed:

$$\operatorname{argmin} \sum_{i,\ j} (m_{ij} - q_j^T p_i)^2 + \beta(||p_i||_2^2 + ||q_j||_2^2) \tag{5}$$

Therefore, the steps to use FunkSVD for recommendation are:

(1). Through gradient descent method, P and Q are solved to minimize the loss function;

(2). Complete the matrix by P and Q;

(3). For a user i, find the location where the previous value is missing, and recommend according to the completion value from large to small.

***2).BiasSVD***

There are many improved versions based on FunkSVD algorithm, BiasSVD is the most popular one. The proposed algorithm is based on the following assumptions: some users or projects will bring their own characteristics. The BiasSVD recommendation system consists of three parts: some are unrelated to the user's project which are called user-biased items; some are unrelated to the user in the project, which are called project-biased items.

Assuming that the average score of the scoring system is μ, the user bias term of the i-th user is , and the project bias term of the j-th project is, the optimization objective function after adding the bias term is shown in formula (5).

$$\operatorname{argmin} \sum_{i,\ j} (m_{ij} - \mu - b_i - b_j - q_j^T p_i)^2 + \beta(||p_i||_2^2 + ||q_j||_2^2 + ||b_i||_2^2 + ||b_j||_2^2) \tag{6}$$

In Web service recommendation, there are a lot of preference information about the history of web service calls. For example, the QoS of a web service itself is generally high, so when it is necessary to predict the QoS value of a web service, the predicted QoS value should contain the offset information of the web service. Another user may prefer throughput or response time in QoS. These are bias information contained by users or web services. Adding these bias information to QoS prediction can improve the prediction accuracy and have better recommendation effect.

Compared with FunkSVD and BiasSVD, the final recommendation effect of BiasSVD is better than FunkSVD. In this thesis, the author also use the BiasSVD algorithm as the recommendation model of Web services.

## III. RESULT OF EXPERIMENTS

The processes of this thesis are as follows:

Firstly, introduce the data set used in the experiment, then illustrate the real algorithm flow, evaluation index and the experimental results of traditional collaborative filtering algorithm and BiasSVD algorithm.

### A. DataSet

All the experiments in this thesis are based on the Web services data set collected by Zheng who is from Chinese University of Hong Kong[10]. A total of 1,974,675 real-world web service invocations are executed by 339 service users from 30 countries on 5,825 real-world web services in 73 countries. Each record also contains the response time and throughput capacity.

### B. Algorithm flow

Input: QoS matrix, regularization parameter β and learning rate α of user web service;

Process: (1). Sparse the QoS matrix according to the corresponding sparsity;

(2).Consider the user bias information and the bias information part of web service, and set bias;

$$b_{ij} = \mu + b_i + b_j$$

μ: Overall average of all records

$b_i$: user offset information

$b_j$: Web Service offset information

(3).In the process of iteration, the initial value of $b_i$, $b_j$ can be set to 0 , and then carry out the iteration ;

$$b_i = b_i + \alpha \ (\mathrm{m_{ij}} - \mu - \mathrm{b_i} - \mathrm{b_j} - q_j^T p_i - \beta b_i)$$

$$b_j = b_j + \alpha \ (\mathrm{m_{ij}} - \mu - \mathrm{b_i} - \mathrm{b_j} - q_j^T p_i - \beta b_j)$$

(4).Optimization objective function:

$$\operatorname{argmin} \sum_{\mathrm{i,\ j}} (\mathrm{m_{ij}} - \mu - \mathrm{b_i} - \mathrm{b_j} - q_j^T p_i)^2 + \beta(||\mathrm{p_i}||_2^2 + ||\mathrm{q_j}||_2^2 + ||\mathrm{b_i}||_2^2 + ||\mathrm{b_j}||_2^2)$$

Output: QoS prediction value p (u, i).

### C. Evaluation indices

There are many evaluation methods for collaborative filtering recommendation algorithm, the two main evaluation standards are Mae and RMSE, as well as precision, recall and other indicators. In this thesis, Mae and RMSE are used.

MAE is defined as follows:

$$\mathrm{MAE} = \frac{1}{\mathrm{n}} \sum_{\mathrm{u \in U\ and\ i \in I}} |\mathrm{p_{ui}} - \mathrm{r_{ui}}| \tag{7}$$

RMSE is defined as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{\mathrm{n}} \sum_{\mathrm{u \in U\ and\ i \in I}} (\mathrm{p_{ui}} - \mathrm{r_{ui}})^2} \tag{8}$$

It can be seen that from the above formula that MAE focuses on the absolute value of prediction error, and its weight for all errors is the same, regardless of the magnitude of the error; RMSE performs a square operation on the difference between the predicted value and the real value before summing up the difference, and the result with large difference is given a greater weight, and the result will be changed accordingly. The value of Mae and RMSE is inversely proportional to the accuracy of prediction, that is, the smaller the value, the more accurate the prediction.

### D. Experimental results and analysis

In this thesis, the author compared UPCC, IPCC, UPICC and FunkSVD as the final comparison algorithm to verify the effect of this algorithm. UPCC, IPCC and UIPCC are user-based, project-based and hybrid collaborative filtering algorithms. FunkSVD is a matrix decomposition algorithm with no bias.

In the experience, we mainly verify whether BiasSVD algorithm can improve the prediction accuracy compared with traditional collaborative filtering algorithms under the same sparse data.

Fig.1 shows the results of MAE and RMSE for different response time and throughput prediction methods by using 5%,15%, and 25% training matrix densities.

Fig.1.Results Of Experiments

| Attributes | Density | Index | Method | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | UMEAN | IMEAN | UPCC | IPCC | UIPCC | FunkSVD | BiasSVD |
| RT | 10% | MAE | 0.9125 | 0.9932 | 0.8028 | 0.7531 | 0.7410 | 0.6149 | 0.5943 |
| | | RMSE | 1.9270 | 1.8948 | 1.6320 | 1.6145 | 1.6318 | 1.4527 | 1.4237 |
| | 20% | MAE | 0.8861 | 0.8528 | 0.7658 | 0.7026 | 0.6929 | 0.5038 | 0.4874 |
| | | RMSE | 1.8947 | 1.8647 | 1.6007 | 1.5739 | 1.5579 | 1.4294 | 1.3838 |
| TP | 10% | MAE | 63.4026 | 61.4932 | 58.3018 | 57.2941 | 57.3011 | 49.3872 | 46.1112 |
| | | RMSE | 88.1048 | 86.3920 | 82.2810 | 82.2230 | 82.5049 | 78.2930 | 75.4039 |
| | 20% | MAE | 59.9610 | 59.2381 | 55.0930 | 54.9961 | 54.1039 | 50.1163 | 48.2937 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | RMSE | 86.1932 | 83.2083 | 81.3940 | 80.3853 | 80.3940 | 73.4091 | 70.3396 |

In a word, relying on the average to predict the effect is the worst, while based on traditional collaborative filtering methods, including UPCC, IPCC and UPICC, the prediction effect is better than the mean prediction, but still not ideal. The key point is that in the case of data sparsity, the method based on matrix decomposition is significantly superior to the traditional collaborative filtering. With the density increases, the performance of all methods is improved which shows that higher quality information can improve prediction accuracy, what′s more, BiasSVD has more superiority in predicting the accuracy.

## IV. CONCLUSION AND FUTURE WORK

The traditional collaborative filtering recommendation algorithm based on users often faces the disadvantage of data sparsity and poor scalability when calculating user similarity. However, using BiasSVD to calculate user similarity in matrix decomposition can improve the disadvantage of data sparsity, and the scalability of itself is also very good. Therefore, the author proposes a recommendation algorithm based on BiasSVD. The algorithm uses FunkSVD to decompose the similarity matrix while considering the user preference.

Through the data sets collected by Zheng, it shows that the algorithm effectively overcomes the shortcomings of data sparsity and improves the accuracy of QoS prediction compared with the traditional algorithm. However, in the process of calculating similarity, the time complexity of iteration needs further optimization. Moreover, the performance of BiasSVD is still weak when faces the cold start. Therefore, the next step is to find solutions to optimize the time complexity and improve the cold start problem. At the same time, as for the hot technology in recent years, for example, deep learning can also be used in recommendation algorithm, and later research will also consider the application of deep learning in recommendation algorithm.

## REFERENCES


[1] Hua Ma, Zhigang Hu, Keqin Li, Haibin Zhu. Variation-aware Cloud Service Selection via Collaborative QoS Prediction[J]. IEEE Transactions on Services Computing, 2019, PP(99): 1-14.

[2] You Ma, M.Aslam, Shangguang Wang, Patrick C.K. Hung, Ching-Hsien Hsu, Fangchun Yang. A Highly Accurate Prediction Algorithm for Unknown Web Service QoS Values [J]. IEEE Transactions on Services Computing, 2016, 9(4): 511-523.

[3] Khan, Arif & Chakkarwar, V.. (2018). A Privacy Preserving Improvised Approach for QOS Aware Web Service Recommendation. 1-7. 10.1109/ICCCNT.2018.8493866.

[4] Y. Hu, Q. Peng and X. Hu, "A Time-Aware and Data Sparsity Tolerant Approach for Web Service Recommendation," 2014 IEEE International Conference on Web Services, Anchorage, AK, 2014, pp. 33-40.

[5] Xin Luo, MengChu Zhou, YunNi Xia, QingSheng Zhu. Predicting Web Service QoS via Matrix-factorization -based Collaborative Filtering under Non-negativity Constraint[C]// 2014 23rd Wireless and Optical Communication Conference. Newark: NJIT, 2014: 1-6.

[6] Chen Wu, Weiwei Qiu, Xinyu Wang, Zibin Zheng, Xiaohu Yang. Time-Aware and Sparsity-Tolerant QoS Prediction based on Collaborative Filtering[C] // 2016 IEEE International Conference on Web Services. CA: IEEE,2016: 637-640.

[7] Zibin Zheng, Hao Ma, Michael R. Lyu, Irwin King. QoS-Aware Web Service Recommendation by Collaborative Filtering[J]. IEEE Transactions on Services Computing, 2011, 4(2): 140-152.

[8] Z. Zheng, H. Ma, M. R. Lyu and I. King, "WSRec: A Collaborative Filtering Based Web Service Recommender System," 2009 IEEE International Conference on Web Services, Los Angeles, CA, 2009, pp. 437-444.

[9] Jiang, Yechun & Liu, Jianxun & Tang, Mingdong & Liu, Xiaoqing. (2011). An Effective Web Service Recommendation Method Based on Personalized Collaborative Filtering. Proceedings - 2011 IEEE 9th International Conference on Web Services, ICWS 2011. 211-218. 10.1109/ICWS.2011.38.

[10] Zheng Z , Zhang Y , Lyu M R . Investigating QoS of Real-World Web Services[M]. IEEE Computer Society, 2014.