ISSN: 2688-8645

An Improved Web Service Recommendation Method based on Decomposition Machine Model

Tingting Zhang, Guihua Huang and Huitong Liao*

Guangdong University of Science&Technology, Dongguan 523000, China

Abstract

With the increasing number of Web services with similar functions on the Internet, traditional collaborative filtering service recommendation methods may encounter problems such as data sparseness, cold start, and poor scalability. To solve the above problems, this project proposes a new Web service recommendation method based on the decomposition machine model. The method decomposes the user trust relationship matrix and the product rating matrix while adding the geographic location information of the service, and transforms the correlation matrix of the calculated user feature vector and item feature vector into the same latent factor space by means of a decomposition machine. Optimize training model parameters to provide users with accurate prediction scores. The ultimate purpose of QoS prediction is to recommend high-quality services to users, improve the efficiency of users' discovery and selection of high-quality services, and ultimately promote the utilization of network Web services and promote service providers to release higher-quality services. Scientific significance, but also has better application value.

Keywords

Web Service; Quality of Services Prediction; Collaborative Filtering; Factorization Machine.

1. Introduction

With the development of the Web2.0 model and related technologies, the Web has gradually become a platform for users to publish, share and consume various services. The explosive growth of Web services brings both opportunities and serious challenges. On the one hand, the network provides users with a large number of shared services, making it more and more convenient for users to obtain services and develop applications. On the other hand, in the face of so many unknown web services, it takes a lot of time and energy for service users to select the required services in a short time. It is extremely important and challenging to recommend high-quality services to users. question. The recommendation system mainly makes judgments based on the user's historical behavior data, which can be divided into two categories: explicit feedback and implicit feedback. Explicit feedback refers to explicit tendencies given by users, such as rating information, trust relationships between users, etc. Implicit feedback refers to implicit tendencies that users do not directly show, such as which products have been purchased, which movies have been rated, and so on.

At present, some scholars have carried out certain research on the above-mentioned problems, and proposed many personalized, collaborative filtering Web service QoS prediction and service recommendation methods. However, traditional collaborative filtering prediction methods may suffer from data sparseness, cold start, and poor scalability. Although some studies have combined fusion clustering, smoothing technology, data dimensionality reduction, user-item similarity scoring and other technologies in the collaborative filtering recommendation algorithm, to a certain extent, the problem of data sparseness has been

ISSN: 2688-8645

effectively improved, and the recommendation accuracy has been improved. At the same time, improved K-Means clustering, matrix factorization, implicit feedback and other methods are also widely used in predictive user rating analysis. However, these studies involve the assumption that users are independent of each other, that is, all users have the same impact on other users' rating behavior, which obviously does not conform to the characteristics of people's daily behavior. There must be differences in the behavior of users in different regions to score the same item, that is, users in different regions and countries should show different interests and hobbies, so there will be regional differences in scoring the same item.

Aiming at the above problems, this project proposes a Web service recommendation method that integrates implicit feedback information and decomposition machine model. The method comprehensively considers the explicit feedback information and implicit feedback information, and adds the user geographic location information while decomposing the user trust relationship matrix and the product rating matrix. The correlation matrix is transformed into the same latent factor space to provide users with accurate prediction scores by optimizing the parameters of the training model. The ultimate purpose of QoS prediction is to recommend high-quality services to users, improve the efficiency of users' discovery and selection of high-quality services, and ultimately promote the utilization of network Web services and promote service providers to release higher-quality services. Scientific significance, but also has better application value.

Table 1. User-service matrix (response time)

Numble	S1	S2	S3
u1	0.4	1.6	null
u2	2.8	null	3.5

2. Research Status

Ran [1] first proposed to integrate QoS into Web service discovery. QoS-based Web service discovery has recently become a research hotspot. Web service recommendation is based on Web service discovery, by using the user's basic information, historical experience or other implicit information to predict user needs and then make recommendations [2]. In the direction of Web service recommendation, it is necessary to consider the functional and non-functional attributes of the service. The most discussed is the quality of service (ie QoS), which includes response time, price, reliability, availability and other indicators [3]. The most direct way to calculate the QoS value is to calculate the average value of the QoS observed by the user on different services, or the average value of the QoS obtained by the service being invoked by different users. The advantage of these two methods is that the method is simple and the amount of calculation is small. The disadvantage is that the individual factors of the user are ignored, which will greatly affect the accuracy of the prediction.

Considering that the QoS value of Web services is related to specific users, in recent years, many works use collaborative filtering recommendation technology to carry out personalized QoS prediction and service recommendation, and have achieved certain results. However, the traditional collaborative filtering technology is greatly affected by data sparsity in application, and has problems such as cold-start users and poor scalability. Recently, model-based collaborative filtering techniques such as matrix factorization and factorization machines have attracted increasing attention in the field of recommender systems [4, 5, 6]. The advantages of this type of method are that it can better overcome the data sparsity and cold-start user problems, and has high computational performance. Therefore, some recent works have introduced matrix factorization into recommendation for web services [7, 8]. However, the

ISSN: 2688-8645

previous work has not fully understood and utilized the QoS characteristics of Web services, so the performance preference in QoS prediction is further improved.

3. Traditional Collaborative Fitering Algorithm based on QoS Prediction

3.1. Memory-based Collaborative Filtering

Memory-Based Collaborative Filtering Algorithm: Find a set of similar neighbors through similarity calculation, and then predict the missing value based on the information of the set of similar neighbors. Typical memory-based collaborative filtering algorithms mainly include: user-based and item-based collaborative filtering algorithms.

3.1.1. User-based Collaborative Filtering

The user-based collaborative filtering algorithm calculates the similarity between the active user and all other users, obtains the neighbor set similar to the current active user's hobbies, and then predicts and recommends it based on the information of the similar user set. In the user-based collaborative filtering algorithm recommendation process, the most critical step is to calculate the similarity between different users and find a set of similar neighbors for active users. The calculation of similarity between different users mainly includes cosine similarity, Pearson correlation coefficient and modified cosine similarity.

3.1.2. Item-based Collaborative Filtering Algorithm

Item-based collaborative filtering algorithm is currently the most widely used algorithm in the field of e-commerce. User-based collaborative filtering algorithm has been applied to a certain extent, but this algorithm has a big flaw. When the number of users increases, it becomes more and more difficult to calculate the similarity between different users, and its time complexity and space complexity increase in a quadratic order. In addition, if a user has not rated the item, the user-based collaborative filtering algorithm cannot find a set of similar users and cannot make recommendations. Based on these shortcomings, Amazon proposed an item-based collaborative filtering algorithm. The item-based collaborative filtering algorithm is also divided into three steps: calculate the similarity between different items, make predictions for active users based on the similarity set information of the target items, and finally determine the recommendation list to recommend to the user.

3.2. Model-based Collaborative Filtering Algorithm

The model-based collaborative filtering algorithm obtains a model by training and learning the user-item score matrix, and then predicts the missing value. The model-based collaborative filtering algorithm uses mathematical calculations, machine learning, and data mining methods to mine the potential relationship between users and items, build a predictive model based on the user's historical rating data, and then predict missing values, and finally recommend.

Model-based collaborative filtering algorithms mainly include matrix factorization, cluster analysis, factorization machines, Bayesian networks, etc. These models are applied in different application scenarios. The algorithm generally completes the construction of the model through offline calculation, and when the model construction is completed, it can quickly complete the recommendation for the user. However, when a new user or item is added to the recommendation system, the user-item matrix needs to be retrained and learned. Therefore, the model-based collaborative filtering algorithm is not suitable for online real-time recommendation. In addition, the model-based collaborative filtering algorithm has poor interpretability. It recommends based on the hidden correlation characteristics of users and items, and it is difficult to explain these users or items. What exactly does the implicit feature mean?

ISSN: 2688-8645

4. Improved Factorization Machine Model

4.1. Construction of QoS Matrix

In the traditional Web service deployment method, this method is difficult to implement, because Web services are often deployed on the provider's server, and the QoS of the service is difficult to monitor. However, with the widespread application of cloud computing platforms, more and more Web services are deployed on public cloud computing platforms, so cloud computing platform providers have the ability to use service QoS monitoring systems to monitor the quality of services deployed on them. In reality, such a service QoS monitoring system may even become an infrastructure of a cloud computing platform. The QoS records generated by the service invocation of all users can be represented by a matrix, referred to herein as a QoS matrix, wherein each QoS record may include multiple QoS parameter values.

For ease of explanation later, we use the following notation to define users, services, matrices and QoS records:

(1) Let U represent the set of all service users, and S represent the set of all Web services.

(2) Represents the QoS records generated by all users calling services. Each row represents the QoS vector of a user; each column represents the QoS vector of a service; each QoS record represents the QoS parameter value generated by user ui calling service sj. Null if ui has not called service sj.

(3) In practice, since a user only uses very few Web services, the above QoS matrix should be very sparse.

4.1.1. Traditional Factorization Machine Model

Collaborative filtering technology has been widely welcomed since it was proposed, and it has also been widely used in the industry. Various supplements and improvements to its model emerge in an endless stream. Especially since the Netflix Challenge started in 2006, the factorization model, also known as the latent factor model, has become the hottest research topic in the recommendation field in recent years. The advantage of FM is that it can simulate the factorization model through eigenvectors, which not only combines the generality and applicability of the feature engineering method, but also can use the factorization model to analyze the interaction between different categories of variables. Modeling estimation, with the help of the open-source implementation tool libFM, can quickly complete the learning task and achieve good accuracy. In addition, FM can handle high-dimensional data in big data environment, can estimate parameters even in sparse data, and has the advantage of linear complexity, which is a general factor model for user interest recommendation or rating prediction.

In statistics, the polynomial regression analysis method that studies a dependent variable and one or more independent variables is called polynomial regression. Polynomial regression is a form of linear regression in which the relationship between the independent variable and the dependent variable can be expressed as an nth-order polynomial. Steffen removed the autocorrelation term in the polynomial regression, and decomposed the interaction between the categorical variables to obtain the factorization machine model. The second-order factorization machine model expression is defined as:

$$\hat{y}(x) = w_0 + \sum_{i=1}^{p} w_i x_i + \sum_{i=1}^{p} \sum_{j>i}^{p} < v_i, v_j > x_i x_j \quad (1)$$

The model parameters are: $w_0 \in R, W \in R^p, V \in R^{p \times k}$

$k << p$ dimension representing factorization, v_i represents the i vector with k features in the V matrix. k k is an assumed value of a positive integer representing the dimension of the eigenvector matrix. w_0 stands for global bias, w_i represents the weight of the i feature variable.

ISSN: 2688-8645

$< v_i, v_j >$ represents the inner product of two low-rank matrices, $< v_i, v_j >$ It is expressed as follows:

$$< v_i, v_j >= \sum_{f=1}^{k} v_{i,f} v_{j,f} \tag{2}$$

After defining the FM model, we need to learn the parameter w_0, W, V from the training set. For the convenience of expression, the parameters in the FM model are uniformly expressed as: $\Theta = \{w_0, w_1, \cdots, w_p, v_{1,1}, \cdots, v_{p,k}\}$。As with any kind of supervised learning, in order to optimize the model parameters, a loss function L needs to be defined to minimize the error between the observed data S and the model:

$$OPT(S) = \arg\min_{\Theta} \sum_{(x,y)\in S} L(\hat{y}(x \mid \Theta), y) \tag{3}$$

For each pair (x, y) in the observed data set, find the sum of the errors between the observed value y and the predicted value $\hat{y}(x \mid \Theta)$, and minimize it to obtain the best parameter set Θ. The loss function L is defined as follows:

$$L(\hat{y}(x), y) = (\hat{y}(x) - y)^2 \tag{4}$$

The FM model contains a large number of parameters, especially when the factorization dimension k is large, then a regularization term needs to be added to prevent the model from overfitting.

The regularization formula is as follows:

$$\mathrm{Reg}(w_0, w_i, v_{i,f}) = \lambda_1 w_0^2 + \lambda_2 \| w_i \|_F^2 + \lambda_3 \| v_{i,f} \|_F^2 \tag{5}$$

$\lambda_1, \lambda_2, \lambda_3$ represents the coefficient of regularization. After adding the regularization term, the optimization function becomes:

$$OPT(S) = \mathrm{argmin}_{\Theta} \sum_{(x,y)\in S} L(\hat{y}(x), y) + \mathrm{Reg}(w_0, w_i, v_{i,f}) \tag{6}$$

In the formula, $\hat{y}(x)$ is the final predicted value of FM, and y is the real value. In order to minimize the loss function, the stochastic gradient descent method (SGD) is introduced to optimize the learning, and for each pair of samples in the observation data, the direction of the gradient descent of the objective function is carried out, iterate as follows:

$$\frac{\partial}{\partial w_0} L(\hat{y}(x), y) = \hat{y}(x) - y \qquad \frac{\partial}{\partial w_i} L(\hat{y}(x), y) = (\hat{y}(x) - y) x_i$$

$$\frac{\partial}{\partial v_{i,f}} L(\hat{y}(x), y) = (\hat{y}(x) - y)(x_i \sum_{j=1}^{n} v_{j,f} x_j - v_{j,f} x_i^2) \tag{7}$$

Finally, the stochastic gradient descent method finally obtains the required parameters through the following formula:

$$w_0 \leftarrow w_0 + \eta \frac{\partial}{\partial w_0} L(\hat{y}(x), y)$$

$$w_i \leftarrow w_i + \eta \frac{\partial}{\partial w_i} L(\hat{y}(x), y) \quad v_{i,f} \leftarrow v_{i,f} + \eta \frac{\partial}{\partial v_{i,f}} L(\hat{y}(x), y) \tag{8}$$

$\eta > 0$ is the learning rate during calculation, or it can be understood as the descending speed. If the value is too large, the model may not converge, and if the value is too small, the convergence speed will be too slow.

ISSN: 2688-8645

4.2. An Improved Factorization Machine Model Recommendation Method

Since the FM model has the advantages of its high accuracy and low complexity, we combine the location information and the FM model to improve the prediction performance of Web service QoS. The formula for calculating FM can be written as follows:

$$\hat{y}(x) = w_0 + w_u + w_i < v_u, v_i > \tag{9}$$

Among them, x represents a service invocation of the user, and y is the QoS value generated by the service invocation. Our proposed location-aware factorization machine takes into account the characteristics of similar users and similar services, and the calculation formula is defined as follows:

$$\begin{aligned} \hat{y} = w_0 + w_u + w_i + \frac{1}{|N_u|}\sum_{m \in N_u}(w_m + < v_u, v_m > + < v_i, v_m >) + \\ \frac{1}{|N_i|}\sum_{n \in N_i}(w_n + < v_u, v_n > + < v_i, v_n >) + \\ \frac{1}{|N_u||N_i|}\sum_{m \in N_u}\sum_{n \in N_i} < v_m, v_n > \end{aligned} \tag{10}$$

N_u is the set of similar users for user u, N_i is the set of similar services for service i, $|N_u|$ and $|N_i|$ is the size of the user set and service set, respectively.

4.2.1. Experimental Datasets

To evaluate the validity of the experiments, we adopt two real datasets QoSDataset 1 and QoSDataset 2 obtained from the wsdream.com website, which have been commonly used to evaluate the performance of web service QoS prediction algorithms, described as follows:

(1) QoSDataset 1 contains about 1.5 million service call records for 100 web services distributed in 25 countries. The data format of the Web service call record is shown in Table 1. Each record contains fields such as user IP, service ID, response time (RTT), data size, and HTTP return code. By processing the records, we get 15,000 users and a RTT matrix. Finally, we extract 5,550 records from 1.5 million records of Web service calls to carry out the experiment.

(2) QoSDataset 2 obtained 1,974,675 QoS records by invoking 5,825 Web services in 73 countries by 339 users in 30 countries. At the same time, the dataset also records the IP addresses of these users, the URLs of the services and the countries they are located in, as well as the QoS records generated by each user calling each web service. The RTT information is extracted from this data set, and the RTT matrix of 339 users × 5825 services is obtained. More descriptions of this dataset are shown in Table 2. Through the analysis, 339 users are distributed in 136 autonomous systems and 30 countries, and 5825 services are distributed in 1021 autonomous systems and 73 countries.

4.2.2. Evaluation Metrics

The metrics for evaluating the recommendation quality of recommender systems are measured by Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). The accuracy of the prediction is measured by calculating the deviation between the predicted user rating and the actual user rating, defined as:

$$MAE = \frac{\sum_{u \in T} |R_u(i_a) - \tilde{R}_u(i_a)|}{|T|} \tag{11}$$

$$RMSE = \sqrt{\frac{\sum_{u \in T} (R_u(i_a) - \widetilde{R}_u(i_a))^2}{|T|}} \quad (12)$$

It can be observed from the formula that RMSE is more sensitive to larger errors, so it can detect larger errors well. Smaller values of MAE and RMSE indicate better prediction performance of the prediction method, and vice versa.

4.2.3. Performance Comparison

Results of running various Web service QoS prediction methods on QoS_Dataset#1 and QoS_Dataset#2. Regardless of the matrix density, our method has smaller MAE and RMSE than other methods, indicating that our method has higher prediction accuracy. Relatively speaking, the performance of FM is higher than other traditional collaborative filtering models and matrix factorization models. However, compared to FM, our method further improves the accuracy of QoS prediction due to the explicit consideration of service and user location factors. We can also observe that when the matrix density is small, regardless of the method, the MAE and RMSE become larger, which means that the data sparsity does have a greater impact on the QoS prediction method based on collaborative filtering. Exactly how matrix density affects QoS predictions will be evaluated later.

5. Conclusion

Aiming at the QoS-aware Web service recommendation problem, this paper proposes an improved factorization machine model recommendation method. The method first considers the network-dependent characteristics of Web service QoS, and combines the network location information of users and Web services with the classical factorization machine model to predict the user's QoS experience on unknown services, thereby providing support for Web service recommendation. Experimental results on real datasets show that by using the factorization machine model, the QoS prediction accuracy is significantly improved compared to previous methods. Moreover, the factorization machine model has linear computational complexity, which can not only solve the problems of data sparseness and cold start, but also solve the problem of poor scalability of traditional collaborative filtering algorithms.

Acknowledgments

Project source: Guangdong University of Science and Technology school-level scientific research project, project number: GKY-2019KYQN-15, GKY-2019KYYB-37.

References

[1] Rich E.User modeling via stereotypes.Cognitive Service,Vol.3,No.4,1979.

[2] Z.H.Liao, J.X.Liu, Y.Z.Liu et al. Review of Web Service Discovery Technology Research [J]. Journal of Information Science, 2008.27(2):186-192.

[3] Z.Zheng, Hao Ma, Michael R.Lyu et. al. WSRec: A Collaborative Filtering Based Web Service Recommender System[C]. Proc. 7th International Conference on Web Sercices (ICWS)2009,pp. 437-444,2009.

[4] Tsai C F, Hung C. Cluster ensembles in collaborative filtering recommendation[J]. Applied Soft Computing, 2012, 12(4): 1417-1425.

[5] Wu J, Chen L, Feng Y, et al. Predicting quality of service for selection by neighborhood-based collaborative filtering[J]. Systems, Man, and Cybernetics: Systems, IEEE Transactions on, 2013, 43 (2): 428-439.

ISSN: 2688-8645

[6] Chen X, Liu X D, Huang Z C, et al. Region KNN: A Scalable Hybrid Collaborative Filtering Algorithm for Personalized Web Service Recommendation[C]. Proc. 8th International Conference on Web Services(ICWS 2010),pp.9-16,2010.

[7] Zheng Z, Ma H, Lyu M R, et al. Collaborative web service qos prediction via neighborhood integrated matrix factorization[J]. Services Computing, IEEE Transactions on, 2013, 6(3): 289-299.

[8] Lo W, Yin J, Li Y, et al. Efficient web service QoS prediction using local neighborhood matrix factorization[J]. Engineering Applications of Artificial Intelligence, 2015, 38: 14-23.