A hybrid random-walk based web service recommendation enhanced by matrix factorization

Jian Lin†
School of Physics and
Electronic Information Engineering,
WenZhou University

Jun Li*
School of Physics and
Electronic Information Engineering,
WenZhou University
Email: omama@wzu.edu.cn

***Abstract*—Recently, the QoS(Quality of service) of Web Service that includes response-time, throughput and so on that needs more accuracy prediction. For many web service callers, choosing the appropriate service in right time should be more significant events. So the web service recommendation is right to be the choice. The collaborative filtering is major approach to predict the QoS of more web service through the observed data. But the sparse density of data need new technology to enhance the accuracy of prediction. And the matrix factorization is also the common approach to solve the prediction. In this paper, we propose the new hybrid approach that combined the predictions with random-walk based and matrix factorization. Comprehensive experiments on the QoS data set of real-world web service, that our approach achieve the more accuracy predictions.**

***Index Terms*—random-work, web service recommendation, matrix factorization**

I. Introduction

Overview the past few years, the collaborative filtering and matrix factorization have success in traditional fields of recommendation, such as Goods, Music, Movie and so on. The recommendation of web service was effected by the achievements. However, the scenario of web service is more complex that suffers from sparse data and incomplete related information. There are so many different web services distributing over heterogeneous network which contains several auto-systems. So the recommendation of web service should solve the problems that sparse QoS(Quality of Service) value collected from various with the untrusted information about location or network. In a word, more measures should be made to enhanced the limited information to achieve the more accuracy of web service recommendation[1][2]. Only that, the system of web service can provide the more quality service.

Web Service QoS predicted information enhanced technology is developing fast. For example, time-aware recommendation that makes prediction by history call record, location-aware recommendation[3][4][5] that make use of numbers of AS(auto system), IP or GPS(Global Position System). But the measures all achieve improvement in accuracy of prediction in small scale of the sparse data. Although the information is critical to prediction, the experiments prove the factor that the more appropriate neighborhood ranking[6] can really boost the accuracy of prediction. So the paper that Random Walk Models[7][8] can efficiently work in real-world dataset in the past few years. With the transition probability matrix which based on the principle of Markov random process[9], the no directed connected users can calculate the similarity in neighborhood selection.

In the field of web service recommendation, the random-walk model is efficient, but the accuracy of prediction needed more improvement. The matrix factorization[10][11] had ever solved the sparse efficiently in similar scenario. Naturally, we will try combining the random-walk model with the matrix factorization to get the good performance. And the matrix factorization also is the best approach to reduction of dimensions, when we calculate the similarity between user and user, the time complexity will be smaller. With more high-efficiency model[12], the hybrid algorithm improves the accuracy of prediction in final.

In summary, to solve the web service recommendation and to increase the accuracy of QoS prediction, in this paper, the contributions we made as following:

- We explore the sparse problem in known probability, and study the sparse dataset and statistic phenomenon in real-world through calculation.
- We propose the hybrid approach to combine the user-based collaborative filtering with matrix factorization.
- We conduct the experiments on real-world dataset, and achieve the best accuracy of QoS prediction.

The rest of this paper is organized as follows. Section II summarizes the related work and our thought about sparse dataset. Section III introduces our approach to combine the CF and MF algorithm. Section IV reports the experiments and analyst the result of approaches. Section V concludes the paper and discusses the future work.

II. Related Work

In this section, we will introduce the intuition of sparse density data, and explore the sparse problem in ideal environment that the sampling rate given in advance. Then the reviews of recommendation model will be displayed, including collaborative filtering, matrix factorization, and random-walk model.

A. *Intuition of sparse density data*

In the real-world dataset environment, our recommendation system samples the whole dataset with $d\%$ density. Suppose that the matrix Q have m users, n services, and the $Q \in \mathbf{R}^{m \times n}$. q_{ij} means the QoS value of $user_i$ called $service_j$.

In our ideal sampling method that we suppose the sampled data following normal distribution. So the every user's sampling number of service is $n \times d\%$. Although the sampling method is ideal, the client user implements the sampling approach in the real environment easily. For example, with parameters $m = 300, n = 5000, d\% = 5\%$, 300 users get about 300×250 service QoS values from the dataset. We approximately calculate the expectation of number on the common invoked number of service. Firstly, we suppose the $user_i$ samples the 250 values in total, the $user_j (i \neq j)$ repeats the sampling process about 299 times. The model subordinates to the binomial distribution of $X \sim b(299, 0.05)$, and we get the common invoked service number for $user_i$ by $Ex = 14.95, D = 14.20$. So the factor that samples with the sparse rate is hard to recover the data of whole. And the location-aware information improves the accuracy of collaborative filtering prediction merely, because the common invoked service scattering in different location area.

B. The approach to solve the sparse density

The sparse problem is always limited the recommendation, and also the hot topic study in recent years. The collaborative filtering is the simple model to give the efficient prediction. But with sparse data and large of empty value, the accuracy is hard to improve. The capital ideas including make use of the adherent information or enhance the connection between users. The front idea is limited by the discrete dataset. Another is efficient, when the connection enhanced by random walk graph.

The matrix factorization is also the efficient approach through the low-rank matrix recovery[13][14]. Although there are many MF-based approaches[15][16] proposed in recent years, the main goal is to overcome the cold start problem[17] and get the more precise predictions. In the service-recommendation, the combination of CF and MF model[18] will achieve the more accuracy.

C. User-based Collaborative Filtering

The CF(Collaborative Filtering)-based algorithm have been widely used. The CF mines a user's common invoked services, which is identified by response-time or throughput, by calculation of similarity (the Euclidean distance) between the $user_i$ and $user_j$. There is a defect that the two users have no common invoked service, the distance will be zero, we identity the smaller value means the more similarity between two users, so the condition should be excluded in algorithm.

$$sim(i,j) = \frac{1}{1 + \frac{1}{N_{ij}}\sqrt{\sum_{k \in S_{ij}} (q_{ik} - q_{jk})^2}} \tag{1}$$

where the number 1 in the denominator is a way of Laplacian smooth to avoid the denominator being 0, and the S_{ij} and N_{ij} means the common service called users and numbers respectively. It can conclude that the pair of users with the smaller distance is calculated the value more near to number 1. In the reversed condition, the value will be number 0.

With the similarity calculated by front step. We can construct the similarity matrix $Sim \in \mathbf{R}^{m \times m}$, the Sim_{ij} means the similarity between $user_i$ and $user_j$. Then, we ranking the neighbors by value of matrix, and choose the topK users to predict the QoS value with the Equation:

$$q_{ik}^{'} = \frac{\sum_{j \in TopK_i} Sim_{ij} \times (Q_{jk} - \overline{Q_j})}{\sum_{j \in TopK_i} Sim_{ij}} + \overline{Q_i} \tag{2}$$

where $\overline{Q_j}$ means the average value of $user_j$, the Equation (2) also considers the different user has different baseline of QoS prediction.

The topK neighbors selection approach is not always smart. Sometimes the number of service is large with the small value, the calculation time will be wasted. If the approach gets the similarity with low-dimension value with low noise that can save the calculation time in large scale data.

D. Matrix Factorization

The MF(Matrix Factorization) has also been chosen for its accuracy. By factorizing the matrix $Q \in \mathbf{R}^{m \times n}$ into user and service latent matrix $U \in \mathbf{R}^{m \times k}$, $S \in \mathbf{R}^{n \times k}$.

$$\begin{aligned} \arg\min_{U,S} \sum_{i=1}^{m} \sum_{j=1}^{n} (Q_{ij} - U_i \cdot S_j^T)^2 + \lambda_U \cdot \sum_{i=1}^{m} \|U_i\|_F^2 \\ + \lambda_S \cdot \sum_{i=1}^{n} \|S_i\|_F^2 \end{aligned} \tag{3}$$

The Equation (3) is used to minimize the loss of Equation, and the $\| \cdot \|_F$ denotes the Frobenius norm[19] to penalize the norms of U and S. Then we can use the gradient descent algorithm with several iterations, and find appropriate matrix U and S at last. Finally, the QoS value will be predicted by the inner product of $U_i \cdot S_j$.

However, the matrix factorization is independent process, the latent matrix $U \in \mathbf{R}^{m \times k}$ can be used as dimension reduction matrix of origin matrix $Q \in \mathbf{R}^{m \times n}$, the dimension reduces from n to k. To some degree, the condition that user is with sparse records will be alleviated, and the the data in large scale will be dealt efficiently in short time, and the calculation will be saved.

E. Random-Walk model

The random-walk model[7] is used to enhanced the similarity between users ,and to get more appropriate neighbors ranking with the transition matrix. In the random-walk model, the algorithm builds the graph $G(V_U, Sim)$ and use the Markov chain to model the state transition of random walk. Let $U_0 \in V_U$ and the $Sim_{0,k}$ means the similarity between $user_0$ and the others. The transition matrix M can calculated by user's similarity. And one step goes by following equation.

$$P_t = (1 - \alpha)P_0 + \alpha M^T P_{t-1} \tag{4}$$

where α means the probability that similarity of user transfers to others, and M means the initial transition matrix with probabilistic value. And the P_0 is always initialized by identity matrix which means the user only cares its own similarity with others with probability 1. Along with the step t being infinite, the probability will converge to be stable, which is decided by the steady state distribution of the Markov chain.

$$P^* = (1-\alpha)(I-\alpha M^T)^{-1}P_0 \tag{5}$$

When the probability is stable, then the P_t will equal to P_{t-1}, then the Equation 4 can be further transformed into Equation 5 shape by linear algebra calculation.

Although the Equation 5 help to enhanced the similarity between users, the collaborative filtering based algorithm can not get more accuracy on the web service dataset with large number of empty value. So the hybrid approach will be the best choice to get more accuracy.

III. Hybrid approach with RW and MF

At first, the matrix Q decomposes into U and S with latent dimension k with the Equation

$$\frac{dloss}{dU_i} = \sum_{j=1}^{n}(Q_{ij} - U_i \cdot S_j^T) \cdot S_j + \lambda_U \cdot \|U_i\| \tag{6}$$

$$\frac{dloss}{dS_j} = \sum_{i=1}^{m}(Q_{ij} - U_i \cdot S_j^T) \cdot U_i + \lambda_S \cdot \|S_j\| \tag{7}$$

after maximum iteration, the matrix U and S will be achieved. And the similarity matrix Sim will be calculated by

$$Sim_{ij} = \frac{1}{k} \cdot \sum_{K} U_{ik} \cdot U_{jk} \tag{8}$$

where K is the latent dimension of matrix U. With the low dimension matrix, the calculation time of similarity matrix Sim will be saved efficiently in large number of data.

With the similarity matrix above, the probabilistic matrix P achieved by extended Equation:

$$P_{i,j} = \frac{A_{ij} \times Sim_{ij}}{\sum_{k \in Adj_i} A_{ik} \times Sim_{ik}} \tag{9}$$

where the A_{ij} parameter refers to the location dataset. With the information of $user_i$ and $user_j$ whether is in the same areas, including auto system area, country area and no direct connection area, the A_{ij} is set to 3,2,1 respectively. The initial probability calculates precisely.

Through the Equation (5) given above, and the value of identity matrix P_0 and P, the final steady transition matrix P^* will be calculated. The P^*_{ij} means the similar probability between $user_i$ and $user_j$. And the enhanced weight that the parameter φ_i can be easily calculated through Equation (10).

$$\varphi_i = \frac{1}{N(j)} \cdot \sum_{j \in S_{ij}} \frac{Sim_{ij}}{P^*_{ij}} \tag{10}$$

At the end of random-walk stage, the Equation (11) calculates the revised similarity which affects the topK nearest neighbors selected[20] eventually.

$$Sim^*_{ij} = \frac{\varphi_i \times P^*_{ij} + \varphi_j \times P^*_{ji}}{2} \tag{11}$$

With the new similarity matrix Sim^* value, the top K nearest neighbors will be selected. The random-walk based similarity enhancement is over. And with the more accuracy similarity, the hybrid approach is important to give final prediction.

$$\begin{aligned} q^*_{ij} = \lambda \cdot (&\frac{\sum_{j \in TopK_i} Sim_{ij} \times (Q_{jk} - \overline{Q_j})}{\sum_{j \in TopK_i} Sim_{ij}} + \overline{Q_i}) \\ &+ (1-\lambda) \cdot \sum_k U_{ik} \cdot S^T_{jk} \end{aligned} \tag{12}$$

The final QoS prediction will be calculated by Equation (12). With the parameter λ, the predictions can adjust to different scenarios. The combination of CF and MF is the key to get more accuracy. The algorithm should consider the $user_j$'s personal QoS value, the appropriate ranking neighbors whose QoS value is empty, and the prediction from MF. The CF algorithm in sparse data will be more low than the real QoS value, and the MF algorithm will be more low high that the real QoS value with the regularizations. The over-fitting or under-fitting and changed by hybrid algorithm. In final, the RWEMF(a hybrid random-walk based web service recommendation enhanced by matrix factorization) should be described. The details of algorithm is in Algorithm (1)(2). And the code of algorithm could been found in WebSite [1].

Algorithm 1 RWEMF

Require: Q , $\overline{Q}$,max_iter,min_thr,λ_{mf},λ_{ru_mf},
Ensure: Q^*
  for $t = 0$ to max_iter **do**
    $U_i = U_i - (Q_{ij} - U_i \cdot S_j) - \lambda_{mf} U_i$
    $S_i = S_j - (Q_{ij} - U_i \cdot S_j) - \lambda_{mf} S_i$
    $loss = \sum (Q_{ij} - U_i \cdot S_j)$
    if $loss < min\_thr$ **then**
      break
    end if
  end for
  Sim=RWE_U(U)
  for $i = 0$ to m **do**
    for $j = 0$ to m **do**
      $v_{cf} = \frac{\sum_{j \in TopK_i} Sim_{ij} \times (Q_{jk} - \overline{Q_j})}{\sum_{j \in TopK_i} Sim_{ij}} + \overline{Q_i}$
      $v_{mf} = U_i \cdot S_j$
      $Q^*_{ij} = \lambda_{ru_mf} \cdot v_{cf} + (1 - \lambda_{ru_mf}) \cdot v_{mf}$
    end for
  end for
  return Q^*

The time complexity of RWEMF is inherited from CF and MF. The time complexity of CF is from O($m \times n \times n$) to O($m \times n \times K$), the K is the latent dimension of matrix U. When the n is large and the K is small, the time will be saved in large dataset. The time complexity of MF is from O($m \times n \times k$), the max_iter(maximum iteration) and $d\%$(the density of dataset) are the influenced elements. In summary, the RWEMF

[1] github.com/neoinmatrix/neosci/tree/master/rwemf

algorithm could not add the extra time complexity, although is large than the sum of RWECF and MF, according to the sparse condition, with the user-based collaborative filtering, the running time[21] is in acceptable scale even on the large web service dataset.

Algorithm 2 RWE_U

Require: U
Ensure: Sim^*
  for $i = 0$ to m **do**
    for $j = 0$ to m **do**
      $Sim_{ij} = \frac{1}{1+\frac{1}{N_{ij}}\sum(U_i-U_j)^2}$
    end for
  end for
  for $i = 0$ to m **do**
    for $j = 0$ to m **do**
      $M_{ij} = \frac{Sim_{ij}}{Sim_i}$
    end for
  end for
  calculate $P^* = (1-\alpha)(I-\alpha M^T)^{-1}$
  for $i = 0$ to m **do**
    for $j = 0$ to m **do**
      $\varphi_i = \frac{1}{N(j)} \cdot \sum \frac{Sim_{ij}}{P^*_{ij}}$
      $Sim^*_{ij} = \frac{\varphi_i \times P^*_{ij} + \varphi_j \times P^*_{ji}}{2}$
    end for
  end for
  return Sim^*

IV. Experiment and Evaluation

A. Dataset and Description

The dataset is from WS-DREAM [2]. The whole dataset includes two attribute sub-dataset: response time(RT) and throughput(TP). The statistics of dataset are shown in Table I. The dataset reflects the real-world condition that we have few clients to observe the QoS value and there are so many service on the Internet.

TABLE I
Statistics of dataset

QoS	numU	numS	min	max	mean	std
RT(sec)	339	5825	0.001	19.999	0.9086	1.9727
TP(kbps)	339	5825	0.004	1000	47.5617	110.7971

The information about the location of users and services can get in Table II. The row of "user_as" means there are 339 users in the dataset. And the 339 users are distributing in 136 areas. Every area has at least 1 user and no more than 31 users. And the average of users on one area about 2.4745 with 2.8338 standard deviation. Notice that the postfix "_as" and "_ct" means area is as(auto system) and ct(country) respectively. From the statistic information about data, the fact that users or services distribute in different area are extremely unbalanced. The location dataset provides inefficiency information, that is why the location information is hardly enhanced the accuracy of our experiments.

TABLE II
Statistics of userinfo and serviceinfo

infoattr	num	size	min	max	mean	std
user_as	339	136	1	31	2.4745	2.8338
user_ct	339	30	1	161	10.9355	28.3673
service_as	5825	992	1	1212	5.8661	40.6092
service_ct	5825	73	1	2395	78.7162	285.9846

B. Evaluation Metric and Parameter

The MAE(Mean Absolute Error) and NMAE(Normalized Mean Absolute Error) may be the common measurable metrics. MAE is defined as

$$MAE = \frac{1}{N}\sum_{i,j}|q_{ij} - \hat{q_{ij}}| \quad (13)$$

The NMAE is computed as the MAE normalized by the mean of all values, which is defined as

$$NMAE = \frac{MAE}{\frac{1}{N}\sum_{i,j}|q_{ij}|} \quad (14)$$

The MAE reflects the absolute error of the predictions. The NMAE reflects the relative error of the predictions. We can compare the ability of predictions from different dataset with NMAE relatively.

C. Result Accuracy and Comparison

There are several classical recommendation algorithms in the experiments as the comparisons. The result of response time and throughput are displayed in Table III and Table IV respectively.

The comparisons including

- UPCC is the user-based collaborative filtering algorithm that calculate the similarity between users with Pearson correlation coefficient. In this case of small number of users, the algorithm is fast with short running time.
- IPCC is the user-based collaborative filtering algorithm that calculate the similarity between users with Pearson correlation coefficient. In this case of large number of services, the algorithm is slow with long running time.
- UIPCC is the hybrid method linearly combines the results of UPCC and IPCC, but the accuracy is more precise than that two. With the running time of total two algorithm, the algorithm is also slow.
- PMF is the matrix factorization[22] algorithm with the model of probability. In this case with sparse data, the process is fast to be convergent to stable state. So the maximum iteration and convergent threshold are significant to keep the running time within acceptable range.
- UL_RWE is user-based random walk model enhanced by the matrix factorization. The reduced-dimension matrix U with k dimensions latent elements, the algorithm is more fast and achieve more accuracy.
- RWEMF is our approach which are more efficient in the experiment. In the base of UL_RWE, the approach

[2] github.com/wsdream/wsdream-dataset

successfully combined the matrix factorization prediction. The running time is close to matrix factorization to achieve more accuracy.

TABLE III
THE MAE AND NMAE OF RESPONSE TIME PREDICTION

model	DS	5%	10%	15%	20%
UPCC	MAE	0.6159	0.5371	0.4966	0.4737
	NMAE	0.6794	0.5917	0.5471	0.5219
IPCC	MAE	0.6805	0.6572	0.5670	0.4955
	NMAE	0.7507	0.7240	0.6246	0.5459
UIPCC	MAE	0.6045	0.5336	0.4879	0.4601
	NMAE	0.6668	0.5879	0.5374	0.5068
PMF	MAE	0.5704	0.4894	0.4584	0.4390
	NMAE	0.6292	0.5391	0.5050	0.4837
UL-RWE	MAE	0.5255	0.4735	0.4462	0.4291
	NMAE	0.5797	0.5216	0.4916	0.4727
XEMF	MAE	0.5518	0.4891	0.4756	0.4877
	NMAE	0.6087	0.5388	0.5239	0.5373
RWEMF	MAE	0.5068	0.4560	0.4344	0.4251
	NMAE	0.5591	0.5023	0.4786	0.4683

TABLE IV
THE MAE AND NMAE OF THROUGHPUT PREDICTION

model	DS	5%	10%	15%	20%
UPCC	MAE	26.8039	22.2826	20.0274	18.689
	NMAE	0.5643	0.4688	0.4212	0.3931
IPCC	MAE	29.5539	29.4531	30.1322	27.5450
	NMAE	0.6222	0.6196	0.6338	0.5794
UIPCC	MAE	26.0401	21.9952	20.0911	18.6256
	NMAE	0.5483	0.4627	0.4226	0.3918
PMF	MAE	22.5499	17.9761	16.5358	15.0594
	NMAE	0.4748	0.3782	0.3478	0.3168
UL-RWE	MAE	19.4043	15.6509	14.3058	13.5797
	NMAE	0.4085	0.3293	0.3009	0.2857
XEMF	MAE	21.0512	17.2567	15.9693	15.5798
	NMAE	0.4432	0.3630	0.3359	0.3277
RWEMF	MAE	18.5121	15.1752	13.9855	13.3388
	NMAE	0.3898	0.3193	0.2942	0.2806

Form the experimental results are shown in Table III IV, we have some observations.

- The matrix factorization algorithm(PMF) achieved more accuracy than the user-based or item-based without enhanced algorithm(UPCC,IPCC,UIPCC).
- The algorithm (HL-RWE) enhanced by random-walk model is achieve more accuracy than the similarity calculated based collaborative filtering algorithm(UPCC,IPCC,UIPCC). So the precision similarity calculation and the appropriate and ranking neighbors selected are the efficient approaches to improve the accuracy.
- The RWEMF algorithm is more efficient than other algorithms and achieves the best accuracy. The sparse density of 5% is more appropriate for the algorithm to have accuracy that the dense density of 20%.
- In the different sub-dataset, the algorithms achieve different performance. The response-time dataset with value range (0.001-19.999) and standard deviation 1.9727 is with fluctuation about 9.86%. The throughput dataset with value range (0.004-1000) and standard deviation 110.797 is with fluctuation about 11.08%. The RWEMF achieves $\frac{0.6794-0.5591}{0.6794} = 0.1771$ in rt dataset and $\frac{0.5643-0.3898}{0.5643} = 0.3092$ in tp dataset. So the sparse density and the fluctuation in the dataset is the important elements to the RWEMF algorithm.

D. Analysis and Deduction

The significant parameters in RWEMF are top K, latent dimensions, the rate of MF union.

From the Figure 1 2, the number of nearby neighbors selected obviously effects the accuracy. Although the accuracy tendency is different in different dataset, the appropriate number of nearby neighbors selected decided the best accuracy in different sparse density. When topK=3, the RWEMF achieves the best accuracy in both response-time and throughput dataset.

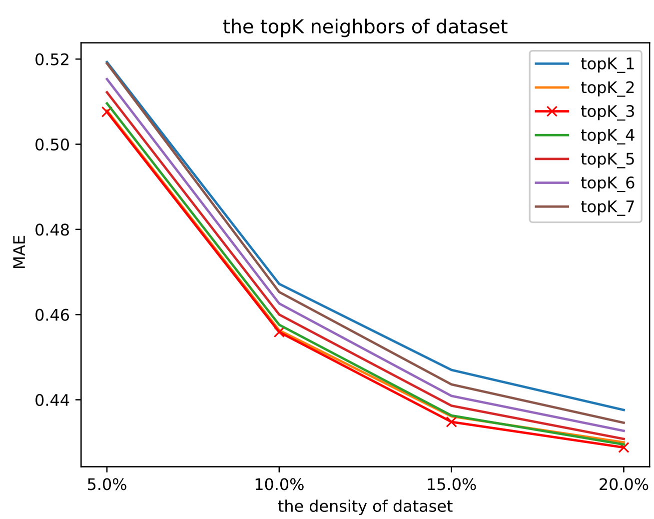

Fig. 1. the MAE of different topK of RWEMF on the response-time dataset

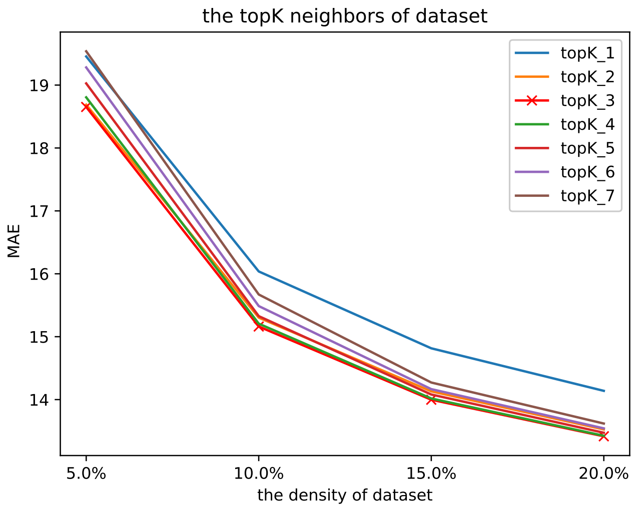

Fig. 2. the MAE of different topK of RWEMF on the throughput dataset

From the Figure 3 4, the rumf parameter is the rate united the MF. In the experiments, we choose the 5% and 20% density. And every rate of density has three lines including the rwe line , MF line and rwemf line. It is clearly to know, the MAE of MF is largest in three, the MAE of rwe is smaller than MF's, with the 0.7 of rumf, the rwemf reaches the best accuracy of MAE. The phenomenon in the throughput dataset is similar. But the response-time dataset with small value is more sensitive to the rate, and it reaches the best accuracy in short range.

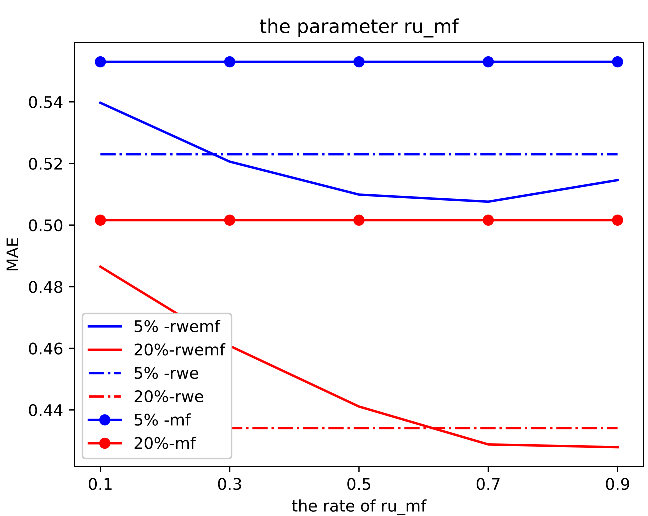

Fig. 3. the MAE of different rate union MF on the response-time dataset

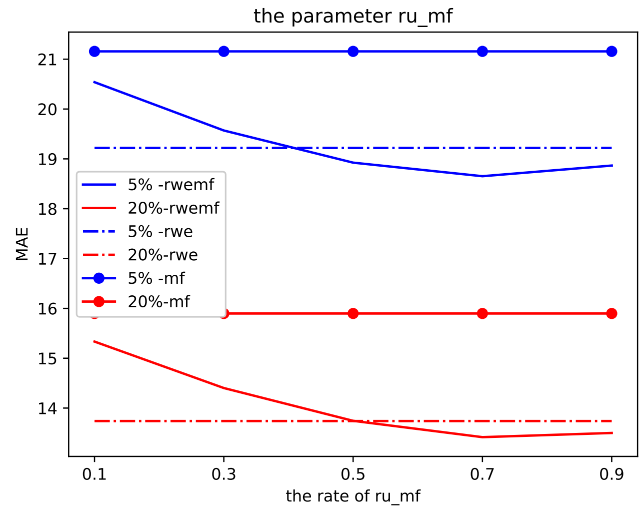

Fig. 4. the MAE of different rate union MF on the throughput dataset

Every point in Figure 5 6 means the prediction of RWEMF,RWE,MF three algorithm minus the real QoS value of dataset, and the points in view are sampling randomly that on behalf of the whole predictions. It is easy to see the AE(Absolute Error) of RWEMF is locating in the middle between RWE's and MF's. Sometimes, the RWE can get the accuracy, but it also processes with the big variance. And the ME can not get the accuracy, but it also runs steadily with the small variance. And the predictions of algorithms are sensitive to value of dataset. The absolute error in throughput dataset fluctuated in large range compared to response-time dataset's.

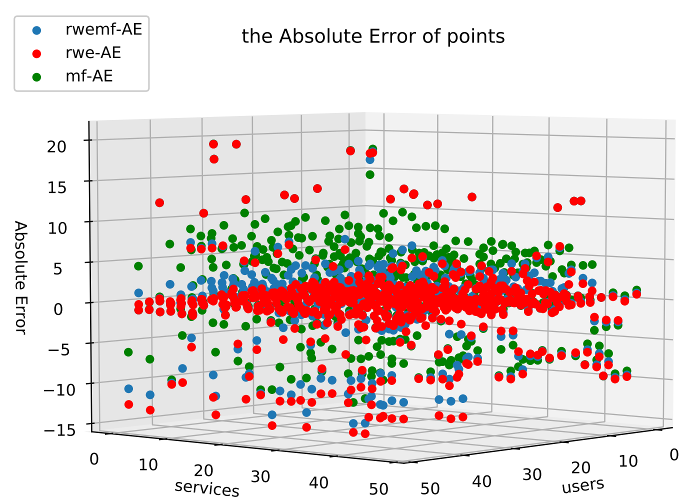

Fig. 5. the Absolute Error on the response-time dataset

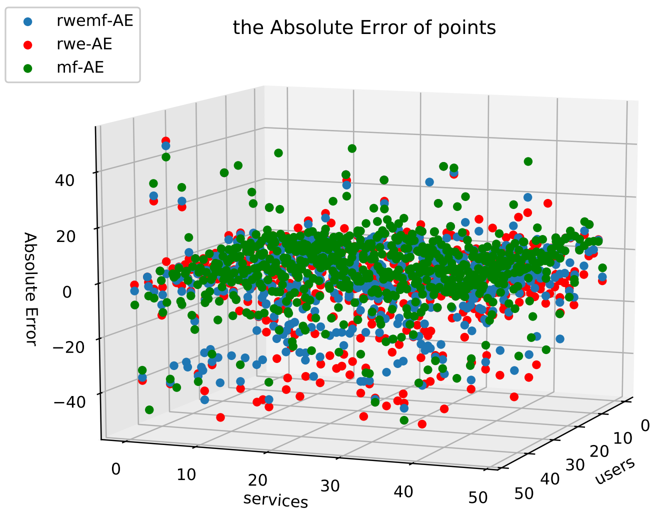

Fig. 6. the Absolute Error on the throughput dataset

V. Conclusion

We propose the RWEMF a hybrid approach to best accuracy in QoS real-world web service dataset. Firstly, We explore the sparse density dataset through statistic calculation. Clearly, the similar calculation and the nearby neighbors selection are significant. And the combination of random-walk user-based collaborative filtering and matrix factorization algorithm is described in the papers. The experiments of RWEMF prove our algorithm is most efficient and the best parameters chosen made RWEMF achieved the best accuracy in this QoS dataset.

In the future, with the best accuracy in this dataset, the RWEMF can be extended by more efficient model. The short running time and exquisite mind can help the algorithm using in real-world web service recommendation easily. The parameters for the hybrid model need more exploration and more study to keep the algorithm more efficient.Although the adherent users's and service's information improve the accuracy finitely, there are more latent information[23] value should be mined in the dataset. The MAE in 5% on response-time dataset is 0.5068 now, Although the value is relative, sometime it could be metrics to measure the ability of algorithm in sparse dataset. Further, the MAE could be lower that 0.5000 by the new hybrid model.

Acknowledgment

The authors would like to thank...

References

[1] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "WSRec: A collaborative filtering based web service recommender system," in *2009 IEEE International Conference on Web Services*, pp. 437–444.

[2] ——, "QoS-aware web service recommendation by collaborative filtering," vol. 4, no. 2, pp. 140–152.

[3] J. Liu, M. Tang, Z. Zheng, X. . Liu, and S. Lyu, "Location-aware and personalized collaborative filtering for web service recommendation," vol. 9, no. 5, pp. 686–699.

[4] B. Xia, Y. Fan, C. Wu, K. Huang, W. Tan, J. Zhang, and B. Bai, "Domain-aware service recommendation for service composition," pp. 439–446.

[5] M. Tang, Y. Jiang, J. Liu, and X. Liu, "Location-aware collaborative filtering for QoS-based service recommendation," pp. 202–209.

[6] W. Lo, J. Yin, Y. Li, and Z. Wu, "Efficient web service QoS prediction using local neighborhood matrix factorization," vol. 38, pp. 14–23.

[7] Y. Yin, F. Yu, Y. Xu, L. Yu, and J. Mu, "Network location-aware service recommendation with random walk in cyber-physical systems," vol. 17, no. 9, p. 2059, random-walk.

[8] H. Park, J. Jung, and U. Kang, "A comparative study of matrix factorization and random walk with restart in recommender systems."

[9] A.-W. Mohammed, Y. Xu, H. Hu, and B. Agyemang, "Markov task network: A framework for service composition under uncertainty in cyber-physical systems," vol. 16, no. 9, p. 1542.

[10] D. Yu, Y. Liu, Y. Xu, and Y. Yin, "Personalized QoS prediction for web services using latent factor models," in *2014 IEEE International Conference on Services Computing*, pp. 107–114, biase LM-LFM.

[11] Y. Ma, S. Wang, F. Yang, and R. N. Chang, "Predicting QoS values via multi-dimensional QoS data for web service recommendations," in *2015 IEEE International Conference on Web Services*, pp. 249–256.

[12] P. Rodrguez-Mier, M. Mucientes, and M. Lama, "Hybrid optimization algorithm for large-scale QoS-aware service composition," vol. 10, no. 4, pp. 547–559.

[13] J. W. Choi and B. Shim, "Statistical recovery of simultaneously sparse time-varying signals from multiple measurement vectors," vol. 63, no. 22, pp. 6136–6148.

[14] A. Agarwal, S. N. Negahban, and M. J. Wainwright, "Stochastic optimization and sparse statistical recovery: An optimal algorithm for high dimensions," in *2014 48th Annual Conference on Information Sciences and Systems (CISS)*, pp. 1–2.

[15] Z. Zhou, B. Wang, J. Guo, and J. Pan, "QoS-aware web service recommendation using collaborative filtering with PGraph," pp. 392–399.

[16] D. N. Tran, S. Huang, S. P. Chin, and T. D. Tran, "Low-rank matrices recovery via entropy function," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4064–4068.

[17] G. Ongie and M. Jacob, "A fast algorithm for convolutional structured low-rank matrix recovery," vol. 3, no. 4, pp. 535–550.

[18] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," vol. 42, no. 8, pp. 30–37.

[19] Z. Chen, L. Shen, D. You, and F. Li, "A user dependent web service QoS collaborative prediction approach using neighborhood regularized matrix factorization," pp. 316–321.

[20] J. E. Hadad, M. Manouvrier, and M. Rukoz, "TQoS: Transactional and QoS-aware selection algorithm for automatic web service composition," vol. 3, no. 1, pp. 73–85.

[21] P. Wang, A. K. Kalia, and M. P. Singh, "A collaborative approach to predicting service price for QoS-aware service selection," pp. 33–40.

[22] R Salakhutdinov A Mnih, "Probability matrix factorization," pp. 1257–1264.

[23] X. Liu and I. Fulia, "Incorporating user, topic, and service related latent factors into web service recommendation," pp. 185–192.