ONLINE PERSONALIZED QOS PREDICTION APPROACH FOR CLOUD SERVICES

Jianlong Xu[1,3], Zibin Zheng[2], Zhun Fan[1] and Wenhua Liu[3]

[1]Guangdong Provincial Key Laboratory of Digital Signal and Image Processing, Department of Electronics Engineering, Shantou University, Shantou 515063, China
[2]School of Data and Computer Science,Sun Yat-sen University, Guangzhou 510275, China
[3]College of Science, Shantou University, Shantou 515063, China
{xujianlong, zfan, whliu}@stu.edu.cn, zhzibin@mail.sysu.edu.cn

Abstract: Personalized Quality-of-Service (QoS) prediction is an indispensable technique to select suitable services for service-based cloud applications. Considering the dynamic nature of services, efficiently and accurately predicting QoS value becomes an urgent and crucial research issue. In this paper, we propose an online personalized QoS prediction approach for cloud service, namely online learning based matrix factorization (OLMF). We build the objective function of online matrix factorization and use stochastic gradient descent algorithm to solve the function. Extensive experiments are conducted on real world public datasets, which verify the effectiveness and efficiency of our proposed approach.

Keywords: Cloud service; Online learning; QoS prediction; Matrix factorization; Cloud computing

1 Introduction

Nowadays, with the development of Internet technology, cloud computing has gained increasing prevalence [1], [2]. Implemented based on service-oriented architecture (SOA), many cloud computing platforms utilize cloud services as the accessible interfaces for various services, for example, data storage service and data access service [3], [4], [5]. Actually, cloud services have become the underlying components in building high quality cloud computing applications [6], [7]. However, as the number of cloud services is booming exponentially, service users (cloud applications that invoke the services) are faced with a massive set of candidate services providing equivalent or similar functionalities, which leads to great difficulty in selecting proper services for each user. Especially, since these services may vary in real-time, the service users need online select the suitable services. Therefore, exploring efficient techniques to build high quality cloud services becomes an urgent and crucial research problem [8], [9].

To select the most suitable service from a set of candidates, non-functional Quality-of-Service (QoS) performance of cloud services becomes a major concern. Many QoS-based approaches have been proposed for cloud service composition [10], [11], cloud service selection [12], [13], and so on. QoS performance properties are mainly comprised of availability, response time, reliability, throughput, etc. [14], [15]. In practice, the user-observed QoS values of most component services are unknown or vary due to the following reasons. For one thing, it is too time-consuming and expensive for users to directly invoke all of cloud services, so there are many unknown QoS property values observed by users. In order to get the unknown QoS values, it is essential to utlize effective prediction technology. For another, QoS properties (e.g., response time, invocation failure rate) observed by different user (e.g., located in different geographical location) are usually different when invoking the same cloud service. This is because the service status (e.g., workload, number of clients, etc.) and the network environment (e.g., congestion, etc.) may change over time, so the known values may be not specific. Since the user-observed QoS values may be changed, it is difficult for cloud application designers to select optimal cloud services at design time and replace low quality cloud services with better ones at runtime [8]. Building high quality cloud computing applications need the services with sufficient personalized user-observed QoS. Therefore, in order to provide users with online personalized QoS information of cloud services, it becomes an urgent task to explore effective approaches to accurately and efficiently predict the unknown QoS values of candidate services without requiring direct invocations.

To address the problems above, in this paper, we propose an online personalized QoS prediction approach based on online learning for cloud computing applications. In our approach, matrix factorization (MF) technology is used to predict the optimal unknown QoS values. Different from the conventional MF model applied in recommender systems, we extend the conventional MF model into an online QoS prediction approach, namely online learning based matrix factorization (OLMF). Under the object function, we use online stochastic gradient descent algorithm to solve the function. Noteworthily, to evaluate our approach, we conduct extensive experiments in real world public datasets in comparison with several well-known methods. The experimental results show the effectiveness and efficiency of our approach.

Briefly summarized, the main contributions of this paper

978-1-5090-1256-5/16/$31.00 ©2016 IEEE

can be concluded as the following aspects: 1) We identify the problem of QoS personalized prediction for SOA based cloud computing applications. 2) We propose an online learning based personalized QoS prediction approach named OLMF, which employ techniques of online learning and matrix factorization. 3) We conduct sufficient experiments on real-world datasets, which verify the effectiveness of our models under various experimental cases.

The rest of this paper is organized as follows: Section 2 presents related work. We present the framework of QoS prediction based on OLMF in Section 3. Then we describe our OLMF approach for QoS prediction in Section 4, and demonstrate the experimental results in Section 5. Finally, we conclude the whole paper and discuss the future work.

2 Related work

In this section, we review the related work from two aspects: matrix factorization and online learning.

2.1 Matrix factorization

In most of the existing reports, collaborative filtering (CF) method is a popular approach to predict QoS values. CF can be divided into memory-based CF, model-based CF, and other hybrid CF. Matrix Factorization (MF) is a typical model-based approach in CF, which has been employed for QoS value prediction by academia and industry in recent years [16], [17]. The main idea of MF-based QoS value prediction methods is to train a model according to the available QoS values in the users-services matrix (i.e., historical QoS values contributed by different users) to predict unknown QoS values in the matrix, which is to exploit the latent factors that can determine QoS both from the user and the service aspects. Matrix Factorization models are accurate and scalable in many applications [18], [19], [20]. Zheng et al. [18] proposed an extended MF model named NIMF, which fuses the neighborhood-based and model-based collaborative filtering approaches to achieve higher prediction accuracy. Aiming to avoid the expensive and costly service invocations, Lo et al. [19] proposed an extended Matrix Factorization (named EMF) framework with relational regularization to make unknown QoS values prediction. In recent years, many researchers focus on integrating MF with additional information (e.g., geographical location, time, reputation, etc.). Zhang et al. [20] proposed a time-aware personalized QoS prediction framework called WSPred. Xu et al. [21] proposed a personalized QoS prediction method named RMF which integrates MF with reputation. Memory-based methods are easy to implement and understand, they can provide a systematic way to train a predefined compact model in the training phase that explains observed values, which is then used to make predictions. Additionally, model-based CF methods can achieve better performance [22]. Hence, we focus on model-based approaches in this paper.

2.2 Online learning

Online learning is a common technique used in areas of machine learning where it is computationally infeasible to train over the entire dataset. In this method, data becomes available in a sequential order and is used to update the best predictor for future data at each step [23], [24], as opposed to batch learning techniques which generate the best predictor by learning on the entire training data set at once. It is used in situations where it is necessary for the algorithm to dynamically adapt to new patterns in the data, or when the data itself is generated as a function of time [25]. Online learning has been gained comprehensive attention in collaborative filtering. Lemire et al [26] proposed slope one algorithm for online rating-based collaborative filtering, Lefevre et al [27] proposed an online algorithm for Non-negative Matrix Factorization. Das et al [28] proposed a scalable online collaborative filtering algorithm which is a mixture of memory-based and model-based algorithm. Mairal et al [29] proposed online learning for matrix factorization, which is applied in computer vision area. In [30] and [31], online algorithms (e.g., stochastic gradient descent method) are adopted for model-based collaborative filtering, but the properties (efficiency, convergence, etc.) of various algorithms are still not well-investigated. In this paper, we study online algorithms to solve the issues facing batch-trained CF algorithms and proposed online learning MF algorithms.

3 The prediction framework

To build hig work for online learning MF based QoS prediction, as illustrated in Fig. 1.

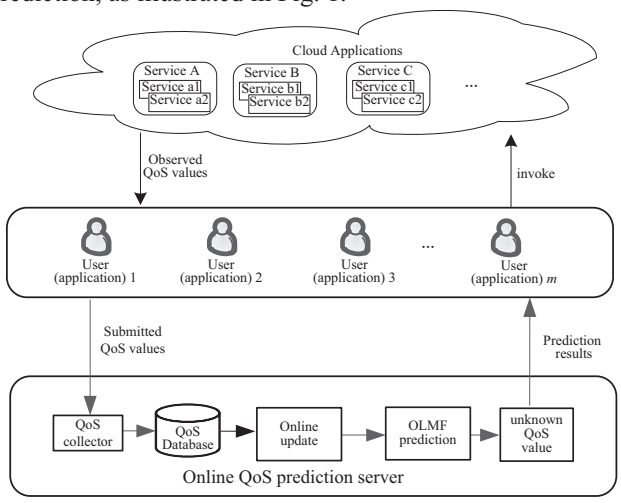

Figure 1 Framework of OLMF QoS prediction

As shown Fig.1, the main idea of our approach is: Online QoS prediction server collects users-observed QoS values and employs online learning MF model to predict unknown QoS values. The details steps are as follows: 1) Users (cloud service application) invoke remote cloud services and observe the QoS values of these services, and submit their observed QoS values to the QoS prediction server. 2) The server first collects the user-contributed QoS values and save them to database.

Then the online learning matrix factorization model performs update if new data comes, and finally makes personalized QoS prediction and returns prediction results to the target user. 3) Users employ the corresponding prediction results to select optimal Web service. Noteworthily, QoS values can be measured at the server side or the user side. QoS properties provided by server-side are not personalized, while QoS properties provided by user-side are more realistic and personalized [32]. This paper mainly focuses on user side QoS properties which will provide different property values for different users (or user applications).

4 QoS prediction based on online learning matrix factorization

In this section, we first introduce the QoS value prediction problem on cloud services in Section 4.1. Then we describe the conventional matrix factorization model in Section 4.2. At last, Section 4.3 presents our online learning matrix factorization (OLMF) models in detail.

4.1 Problem description

The problem studied in this paper is to predict the unknown QoS value at each time slice. The prediction process usually includes a user-service-time matrix, where each entry in this matrix represents the value of a certain QoS property (e.g., response-time in this example) of a cloud service observed by a service user at each time slice. The intractable issue in QoS prediction is data sparsity. High data sparsity means that most entries in the user-service-time invocation matrix are unknown. Thus, our main task is to fulfill unknown values in the matrix. However, QoS values are changed over time, which may lead to the change of QoS prediction result for each user. Therefore current approaches need modification to work more effectively.

Our goal of QoS prediction is to employ the observed QoS data to estimate other unknown values. Intuitively, suppose that there are a set of users $U=\{u_1, u_2,\cdots, u_m\}$, and a set of services $S=\{s_1, s_2,\cdots, s_n\}$ at a certain time slice, we can estabish a $m\times n$ user-service sparse matrix $R\in\mathbb{R}^{m\times n}$. In this matrix, each entry r_{ij} ($i\le m$, $j\le n$) represents the value of a certain QoS property (e.g., response time), which means user i invokes service j and submits a QoS value. The online QoS prediction problem can be modeled as a collaborative filtering problem, which can be expressed as: at each time slice, the unknown values can be predicted by approximately reconstructing from the observed values and performed online and accurately.

4.2 QoS prediction with matrix factorization

The basic matrix factorization (MF) model maps users and services to a joint latent factor space to represent user-service interactions. The key step of MF is to factorize the high dimensional matrix into two low dimensional feature matrices that are in the same feature space. Let $U\in\mathbb{R}^{l\times m}$ denotes the feature matrice of user U, and $S\in\mathbb{R}^{l\times n}$ represent service latent feature matrices, where l is the number of latent factors. The number of factors l is called dimensionality [33]. The prediction for invocation matrix R can be factorized as the product of U and S approximately as follows:

$$\tilde{R}=U^T S, \tag{1}$$

where $\tilde{R}=\{\tilde{r}_{ij}\}(1\le i\le m,\ 1\le j\le n)$, $\tilde{r}_{ij}$ is the predicted value of r_{ij}. The key step of predicting QoS using MF model is to build up an objective function, which is designed to reduce the total approximate errors. The objective of MF is to minimize the sum of squared errors with quadratic regularization terms. Let U_i and S_i denote the i^{th} and j^{th} column of U and S, respectively. The objective function can be represented as Eq.(2).

$$\min_{U,S}\mathcal{L}(U,S)=\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}I_{ij}\left(r_{ij}-U_i^T S_j\right)^2+\frac{\lambda_U}{2}\|U\|_F^2+\frac{\lambda_S}{2}\|S\|_F^2, \tag{2}$$

where I_{ij} is a indictor function, I_{ij} is 1 or 0, indicating r_{ij} is known or not. The Frobenius norm $\|\cdot\|_F$ is employed to avoid the over-fitting issue during the learning process. λ_U, λ_S are both small positive decimals. Gradient decent algorithm can be employed to get a local minimum of the objective function, and the feature space U and S can be reconstructed as follow:

$$U_i^{'}=U_i-\alpha\frac{\partial\mathcal{L}}{\partial U_i},\quad S_j^{'}=S_j-\alpha\frac{\partial\mathcal{L}}{\partial S_j}, \tag{3}$$

where α is the learning rate. In Eq. (3), the low-rank matrices move toward the average gradient descent at each iteration. The local minimum of the objective function can be found by performing gradient descent in $\partial L/\partial U_i$ and $\partial L/\partial S_j$ controlled by α. Once the stopping criterion is met, all unknown values in the original matrix can be recovered by these feature spaces. In this process, since the entire U and S are predicted by batch-training, it may cost too much time in the online condition. Therefore, to improve the prediction efficiency, employing online algorithm is crucial and necessary.

4.3 Online learning for matrix factorization

As QoS values may vary over time, online learning algorithms are required to keep continuous and incremental updating using the sequentially user-observed QoS data. For this purpose, we employ stochastic gradient descent (SGD) [34], a classic online learning algorithm, to train our OLMF model. The main idea of OLMF algorithm can be restructured as follow: if the observed QoS values are coming sequentially, we adjust the model stochastically by taking into account that value only at each time slice. Suppose the new coming QoS value is $(U_i, S_j, r_{ij}, t_{ij})\in R_t$, where t denotes each time slice. For each QoS value observed by user U_i for invoking service S_j, we can minimize the following objective function:

$$\mathcal{L}(U,S)=\frac{1}{2}I_{ij}\left(r_{ij}-U_i^T S_j\right)^2+\frac{\lambda_u}{2}\|U_i\|_2^2+\frac{\lambda_s}{2}\|S_j\|_2^2 \quad (4)$$

The first term in the equation above is the squared error between the observation and predicted value, and the remaining two terms are the corresponding regularizations. Notice that here the trade-off constants, λ_u and λ_s, are on different scale from those in Eq. (2). We employ stochastic gradient descent algorithm to acquire a local minimum within iterative loops and obtain the following update equations:

$$U_i \leftarrow U_i-\alpha(I_{ij}(U_i^T S_j-r_{ij})(U_i^T S_j)^{'} S_j+\lambda_u U_i), \quad (5)$$

$$S_j \leftarrow S_j-\alpha(I_{ij}(U_i^T S_j-r_{ij})(U_i^T S_j)^{'} U_i+\lambda_s S_j), \quad (6)$$

where α is the learning rate which is used to control how much change to make at each step. This online algorithm is stochastic every time, it can be described that every time when a new data sample (U_i, S_j, r_{ij}, t_{ij}) comes, online updating can be performed on its corresponding factors U_i and S_j using Equation 5 and 6. The pseudo code of QoS prediction algorithm based on OLMF is provided in Algorithm 1.

Algorithm1 OLMF-based QoS prediction construction

Input: original training matrix R，sequentially QoS value (U_i, S_j, r_{ij}, t_{ij}), all the model parameters

Output: prediction result: $\tilde{r}_{ij} \leftarrow (U_i, S_j)$

```
repeat
    initialize U_i and S_j.with small random numbers;
    if a new data (U_i, S_j, r_ij, t_ij) comes then
        update r_ij, t_ij;
        update U_i, S_j by Equation 5 and 6;
        if convergence then
            wait until new QoS data comes;
        end if
    end if
until forever
```

In Algorithm 1, the newly QoS data is collected to update the model at each iteration, or else initialized with randomly numbers U_i and S_j when conduct matrix factorization.

5 Experiment

In this section, we conduct experiments to validate our OLMF approach, our experiments are intended to: 1) verify the rationality of our approach; 2) discuss how the model parameters affect the prediction accuracy; and 3) compare our OLMF approaches with other state-of-the-art methods. Our experiments were conducted on a machine with a 2.4 GHz Intel CPU and 8GB RAM, running Win7 operating system.

5.1 Dataset description

In this paper, the datasets is from a real world Web service QoS dataset [20], which includes both response time and throughput values. These QoS values are collected by 142 users (PlanetLab nodes) invoking 4,500 Web services for 64 consecutive time slices, at an interval of 15 minutes, which can be built to a three-dimensional user-service-time matrix. In our experiments, we use the response time dataset to verify our approach.

5.2 Evaluation metrics

We use MRE（Median Relative Error）and 90% NPRE (ninety percentile relative error) as the evaluation metrics of prediction accuracy. MRE and NPRE are defined as follows:

$$MRE=\underset{I_{ij}=0}{median}(\left|\tilde{r}_{ij}-r_{ij}\right|/r_{ij}) \quad (7)$$

$$NPRE=90\%\times\left|\tilde{r}_{ij}-r_{ij}\right|/r_{ij} \quad (8)$$

Due to the large variance of QoS values, MRE and NPRE are more appropriate to evaluate the QoS prediction optimization efforts than other metrics (e.g. mean absolute error).

5.3 Performance comparison

To prove the effectiveness of our model, we ran extensive experiments and compare our method with probabilistic matrix factorization PMF [35], which is a probabilistic linear model with Gaussian observation noise.

Table I Parameter settings

Parameters	Methods	
	PMF	OLMF
density	10%-50%	10%-50%
time slice	64	64
rounds	20	20
dimension	10	10
α	0.01	0.8
λ_u=λ_s	30	0.003
maxIter	100	100

In this experiment, matrix density is defined as the density of the training dataset. In this experiment, each QoS prediction method is run on 10 different matrices, whose densities are 10% to 50% at a step increase of 10%. For example, a matrix density of 10% means that 10% entries in the matrix are used for predicting unknown QoS values, while the remaining 90% are ones waiting to be predicted. Additionally, the maximum iterations are both set to 100. At each time slice, each approach is performed 20 times and taken the average.

Table II Accuracy comparison on response time

Density	Metrics	PMF	OLMF	Impro. vs PMF
MD=10%	MRE	0.593	0.309	47.9%
	NPRE	3.017	0.970	67.8%
MD=20%	MRE	0.596	0.281	52.8%
	NPRE	3.414	0.899	73.6%
MD=30%	MRE	0.581	0.266	54.2%
	NPRE	3.390	0.865	74.5%
MD=40%	MRE	0.564	0.258	54.3%
	NPRE	3.294	0.849	74.2%
MD=50%	MRE	0.546	0.255	53.2%
	NPRE	3.198	0.844	73.6%

Table II shows the MRE and NPRE results of different methods with different density. The experimental results

show that our OLMF approach can achieve smaller MRE and NPRE values than PMF for response-time with different matrix densities, which indicates further higher accuracy than existing approaches and verifies the effectiveness of our approach. Concretely, in average sense, OLMF can acquire 52.5% improvement in MRE and 72.7% improvement in NPRE than PMF model.

5.3 Impact of Matrix Density

To study the impact of matrix density, we vary the density of matrix from 5% to 45% with a step value of 10%. We set dimensionality = 10, and $\lambda_u = \lambda_s$=0.003, respectively. Fig. 2 illustrates the experimental results.

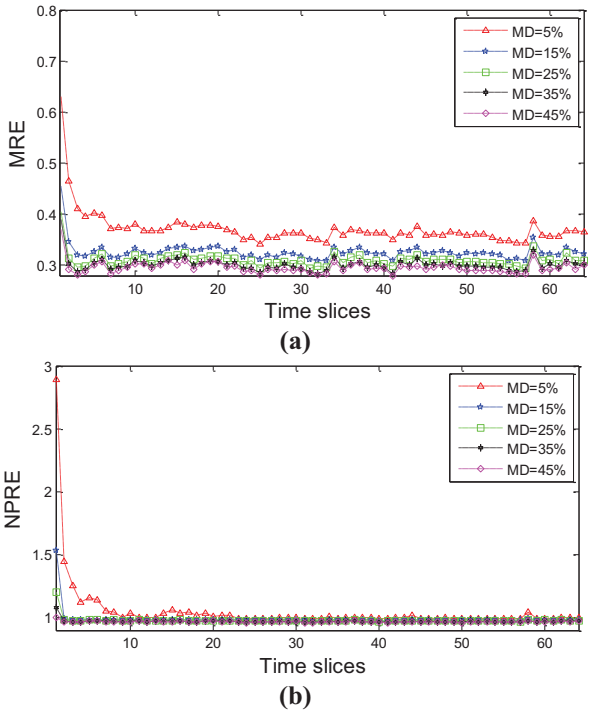

Figure 2 (a) MRE at different time slice. **(b)** NPRE at different time slice.

Fig.2 shows the average MRE and NPRE under different density matrix for 64 time slice. We can see that with the increase of density matrix, prediction result becomes more accurate. In particular, the smaller the matrix density (e.g., 5%), the bigger the error will be shown. This is because the matrix is sparse and the prediction model is easy to fall into overfitting problem. This problem weakens with the increase of matrix density, which will achieve more accurate prediction result.

5.4 Convergence time analysis

In order to evaluate the effectiveness of online MF method, we compared convergence time with PMF. As shown in figure 3, for OLMF, the convergence time is long at the beginning, while it becomes fast after the fifth time slice. This is because OLMF incrementally updates the model by online learning algorithm, while PMF updates the model by batch learning algorithm. PMF need to re-train the whole model at each time slice, leading to high computational cost. Therefore, our OLMF model is more effective than PMF.

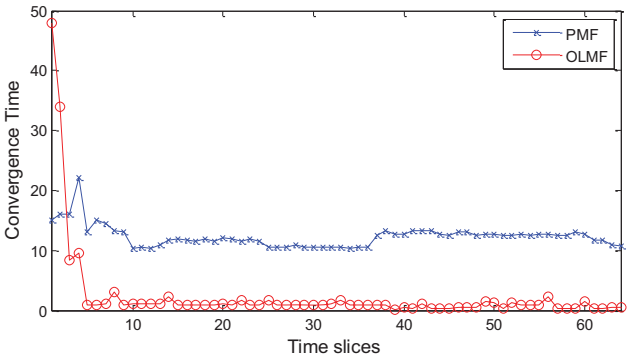

Figure 3 Convergence time comparison

6 Conclusion and future work

Cloud computing is popular nowadays. Many cloud applications are built by cloud services. To build high quality cloud computing applications, exploring efficient techniques become an urgent task. Due to the time-variant user-observed QoS, it is a challenge to select proper services for each cloud application. In this paper, we propose an online learning based personalized cloud service QoS prediction approach named OLMF for cloud computing applications. Firstly, we build the online MF objective function. Then we use online stochastic gradient descent algorithm.to solve the function. Finally, Extensive experiments based on a real-world dataset are conducted to verify its good performance in achieving accuracy and efficiency.

In the future, to continuously improve our prediction performance, we will take more different scenarios into consideration, such as adapting the prediction model when the scale of users or services changes. We also will try to combine other related information (e.g., location, reputation) to improve the prediction outcome in terms of matrix factorization itself.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Project No. 61472338) and the Guangdong High-Level University Project “Green Technologies for Marine Industries”.

References

[1] Venters, Will, and Edgar A. Whitley. A critical review of cloud computing: researching desires and realities. Journal of Information Technology, vol.27, no.3, 2012, pp. 179-197.

[2] Moura, Jose, and David Hutchison. Review and analysis of networking challenges in cloud computing, Journal of Network and Computer Applications, vol.60, 2016, pp. 113-129.

[3] Google AdWords. AdWords API. In https://developers. bgoogle.com/adwords/api/docs/reference/, 2015

[4] Google Apps Script, Soap Services. In https://developers. google. com/apps-script/reference/ soap/, 2015

[5] Amazon Relational Database Service. SOAP API. In http://docs.aws.amazon.com/AmazonRDS/latest/UserGui

de/ using-soap-api. html, 2015
[6] Shangguang Wang, Lei Sun, Qibo Sun, Jie Wei, Fangchun Yang. Reputation Measurement of Cloud Services based on Unstable Feedback Ratings. International Journal of Web and Grid Services, vol. 11, no. 4, 2015, pp. 362-376.
[7] Phaphoom, Nattakarn, et al. A survey study on major technical barriers affecting the decision to adopt cloud services. Journal of Systems and Software, vol.103, 2015, pp. 167-181.
[8] Jieming Zhu, Pinjia He, Zibin Zheng, and M.R. Lyu. Towards Online, Accurate, and Scalable QoS Prediction for Runtime Service Adaptation. in Proceedings of the 34th International Conference on Distributed Computing Systems (ICDCS), 2014, pp. 318-327.
[9] Shangguang Wang, Qibo Sun, Hua Zou, Fangchun Yang. Towards an Accurate Evaluation of Quality of Cloud Service in Service-oriented Cloud Computing. Springer Journal of Intelligent Manufacturing, vol.25, no.2, 2014, pp. 283-291
[10] Zia ur Rehman, Omar Khadeer Hussain, Farookh Khadeer Hussain. Parallel Cloud Service Selection and Ranking Based on QoS History. International Journal of Parallel Programming, vol.42, 2014, pp. 820-852.
[11] BQ Huang, CH Li, F Tao. A chaos control optimal algorithm for QoS-based service composition selection in cloud manufacturing system. Enterprise Information Systems, vol.8, 2014, pp.445-463.
[12] Quan Z. Sheng, Xiaoqiang Qiao, Athanasios V. Vasilakos, Claudia Szabo, Scott Bourne, Xiaofei Xu. Web services composition: A decade's overview. Information Sciences, vol.280, 2014, pp.218-238
[13] Jian Wu, Liang Chen, Tingting Liang. Selecting Dynamic Skyline Services for QoS-based Service Composition. Applied Mathematics & Information Sciences, vol.8, 2014, pp. 2579-2588
[14] B. A. Malloy, N. A. Kraft, J. O. Hallstrom, and J. M. Voas. Improving the predictable assembly of service-oriented architectures. IEEE Software, vol. 23, 2006, pp. 12–15
[15] D. A. Menasce. QoS issues in web services. IEEE Internet Computing, vol. 6, 2002, pp. 72–75.
[16] Y. Xu, J. Yin, W. Lo, Z. Wu. Personalized Location-Aware QoS Prediction for Web Services using Probabilistic Matrix Factorization. In Proc. of International Conference on Web Information System Engineering (WISE), 2013, pp. 229-242.
[17] Y. Xu, J. Yin, W. Lo. A Unified Framework of QoS-based WebService Recommendation with Neighborhood Extended Matrix Factorization. In Proc. of IEEE International Conference on Service Oriented Computing and Applications (SOCA), 2013, pp.198-205
[18] Z. Zheng, H. Ma, M. Lyu, and I. King. Collaborative Web service QoS prediction via neighborhood integrated matrix factorization. IEEE Transactions Services Computing, vol. 6, 2013, pp. 289-299
[19] W. Lo, J. Yin, S. Deng, Y. Li, and Z. Wu. An Extended Matrix Factorization Approach for QoS Prediction in Service Selection. In Proc. of International Conference on Services Computing (SCC), 2012, pp.162-169.
[20] Yilei Zhang, Zibin Zheng, Michael R. Lyu. WSPred: A Time-Aware Personalized QoS Prediction Framework for Web Services. in Proceedings of the 22th IEEE Symposium on Software Reliability Engineering (ISSRE), 2011, pp. 210-219.
[21] Jianlong Xu, Zibin Zheng, and Michael R. Lyu. Web Service Personalized QoS Prediction via Reputation-based Matrix Factorization. IEEE Transactions on Reliability. vol.65, no.1, 2015, pp.1-10
[22] R. Salakhutdinov and A. Mnih. Probabilistic Matrix Factorization. In Proc. of Advances in Neural Information Processing Systems, 2007, pp. 1257-1264
[23] Shalev-Shwartz S. Online Learning and Online Convex Optimization. Foundations and Trends® in Machine Learning, vol.4, no.2, 2011, pp.107-194
[24] Bertsekas D P. Incremental gradient, subgradient, and proximal methods for convex optimization: A survey. Optimization for Machine Learning, 2010, pp.1-38.
[25] Kushner H, Yin G G. Stochastic approximation and recursive algorithms and applications. Springer Science & Business Media, 2003.
[26] Lemire D, Maclachlan A. Slope One Predictors for Online Rating-Based Collaborative Filtering. in SIAM Data Mining, 2005, pp.1-5.
[27] A. Lefevre, F. Bach, and C. F´evotte. Online algorithms for nonnegative matrix factorization with the itakura-saito divergence. in WASPAA, 2011, pp. 313–316.
[28] A. Das, M. Datar, A. Garg, and S. Rajaram. Google news personalization: scalable online collaborative filtering. in proc. of International Conference on World Wide Web (WWW), 2007, pp. 271–280.
[29] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. Journal of Machine Learning Research, vol. 11, 2010, pp. 19–60
[30] Y. Koren. Collaborative filtering with temporal dynamics. Communications of the Acm, vol.53, no.4, 2010, pp. 89-97.
[31] N. D. Lawrence and R. Urtasun. Non-linear matrix factorization with gaussian processes. in Proc. of the 26th Annual International Conference on Machine Learning (ICML), 2009, pp. 601-608
[32] Zheng Z, Wu X, Zhang Y, et al. QoS ranking prediction for cloud services. IEEE Transactions on Parallel and Distributed Systems , vol.24, no.6, 2013, pp. 1213-1222
[33] Zibin Zheng and Michael R. Lyu. Personalized Reliability Prediction of Web Services. ACM Transactions on Software Engineering and Methodology, vol. 22, no. 2, 2013, pp. 1-28.
[34] Bottou, Léon. Large-scale machine learning with stochastic gradient descent. In Proceedings of COMPSTAT'2010. Physica-Verlag HD, 2010, pp. 177-186
[35] R. Salakhutdinov and A. Mnih. Probabilistic Matrix Factorization. in Proc. of Advances in Neural Information Processing Systems, pp. 1257-1264, 2007