

Time-Aware Customer Preference Sensing and Satisfaction Prediction in a Dynamic Service Market

Haifang Wang, Zhongjie Wang, and Xiaofei Xu (✉)

Harbin Institute of Technology,
Harbin, Heilongjiang, People's Republic of China
{wanghaifang, rainy, xiaofei}@hit.edu.cn

Abstract. In the dynamic service market, massive services and variations of their Quality of Services (QoS) and service contract make it difficult for customers to acquire the information of all the services comprehensively and timely. As a result, customers cannot raise accurate expectations. A customer has to choose services in terms of the incomplete information of the dynamic service market to achieve higher Satisfaction Degree (SD) as much as possible. Besides, because a customer's preferences vary over time, his SD is also time-aware. Therefore, for service providers, to accurately recommend services to customers, it is necessary to sense the customer preferences varying against time and predict personalized customers' satisfaction. To address this challenge, we propose a time-aware customer preference sensing and satisfaction prediction method based on customer's service usage history and change history of services. Firstly, the customer satisfaction model on contract-based services is proposed to measure customers' satisfaction for services. Then, we adopt the box-plot method and the frequency histogram to sense time-aware customer preferences. In addition, a time-aware personalized SD prediction algorithm called SDPred is presented to predict the missing values due to information asymmetry. Meanwhile, several experiments have been conducted based on a released data set, which verify the effectiveness of our methods. Besides, the impact of parameter settings in the SDPred algorithm is further studied, which provides more evidences to illustrate the superiority of our method.

Keywords: Customer satisfaction · Customer preference · Time-aware · Satisfaction degree prediction

1 Introduction

The increasingly diverse services have been deployed on the Internet, however, facing massive services with the same or similar functionalities, customers usually feel vacant while selecting one. Since these services are variable over time in the dynamic service market, it is impossible for customers to know all the information of services comprehensively and propose their expectation clearly. However, customers usually choose services that approximate to their preferences based on the known information of services, which makes it difficult to identify the customer requirement for the Quality of

Services (QoS) and to satisfy their preferences only by reducing overhead blindly. Therefore, it is becoming vital to confirm the personalized selection rules based on their preferences. In addition, in real life, our experience tells us that customer preferences to goods vary against time, for instance, customer *A* preferred to *Air China* last year, nevertheless, he enjoy *China Southern Airlines* today, which is called customer preference drift [8, 16]. Besides, the time-aware customer preferences play an important role in selecting services from a large pool of services with same or similar functionalities. Therefore, a novel issue can be summarized as **RQ1** shown in Fig. 1, which takes a customer as an example: **How can we acquire the time-aware customer preference sensing based on customers' service usage history?** For instance, for customer *A*, his time-aware preferences to airline companies can be sensed based on his service usage history, which will affect his future selection.

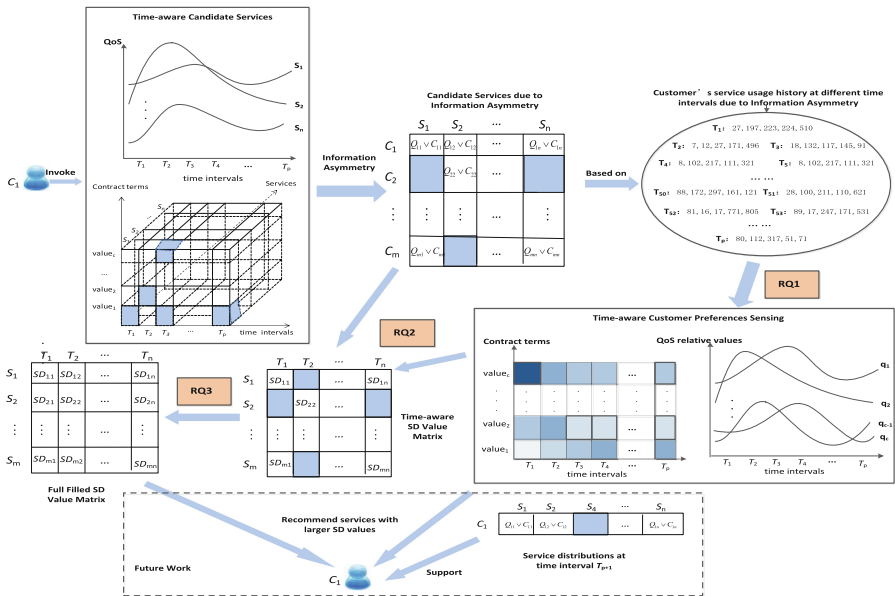


Fig. 1. Issues to be solved in this paper

While depicting customer preferences, it is easy to discover that conventional functional properties and QoS are not enough to describe the individual customer requirements as well as the characteristics of the services offered by the broker, e.g. payment method, etc. Therefore, service contract [1] is introduced into services in this paper to describe the individual customer requirements and the characteristics of services. Service contract is the agreement between providers and customers while invoking a specific service in given conditions [15], which may meet the actual demands. However, the traditional customer satisfaction measurement methods based on QoS cannot satisfy the demand. Thus, **RQ2** presented in Fig. 1 is identified as following: **How do we measure the customer satisfaction degree for contract-based services?** For customer *A*, he focused on QoS preferences before, e.g. the response time

of the app or website to reserve plane tickets should be no more than 20 s. However, today he raises another preference that the payment method should be *Alipay*, which cannot be described by QoS and measured by the conventional methods.

In addition, customer preferences affect their Satisfaction Degree (SD) for goods, which means that the SD values change over time. Similarly, the SD values of customers for services vary against with time as well. Given a set of services with same or similar functionalities, Fig. 2 illustrates a toy example of the SD changes we study in this paper. In this figure, customer C_1 has invoked several services S_1, S_2 , etc. in the past. C_1 has different preferences to them with specific invocation time, which means these services correspond to the distinct SD values during different time intervals. Likewise, other customers keep diverse preferences to various services with corresponding SD values called time-aware personalized SD values. The premise of the issues above is that customers can make their requirements clearly and they are provided all the information of services. However, in reality, due to information asymmetry, they cannot raise their requirements distinctly for services in the dynamic service market, which also leads to the loss of parts of the time-aware personalized SD values. As a result, a critical challenge is formed, which is defined as **RQ3** shown in Fig. 1: **How can we predict the missing SD values?** For the example above, customer A keeps different SD values for the *China Southern Airlines* at diverse time intervals, however, due to information asymmetry that the QoS and contracts of services are fuzzy at some a time interval, his satisfaction for the *China Southern Airlines* cannot be measured at the time interval, e.g. $T_1: 0.71, T_2: 0.69, T_3: \text{null}, T_4: 0.75$, etc. Then, **RQ3** focuses on the SD prediction at T_3 based on the known SD values and time-aware personalized SD values of other customers.

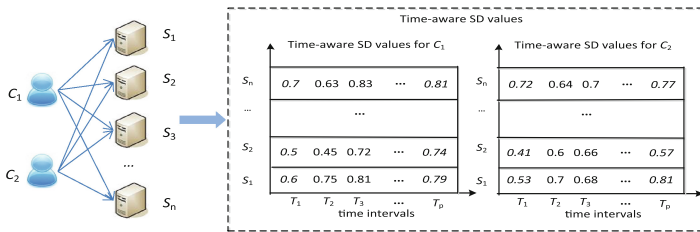


Fig. 2. SD values changes for customers with various services at different time intervals

Based on the customer preferences analyzed and the full filled SD value matrix, we can recommend the more valuable services to customers precisely, which will be focused on in our future work.

To answer **RQ1**, we have sensed time-aware customer preferences by the box-plot method and colorful frequency histogram according to their service usage history to show the customer preference distributions. Then, the Customer Satisfaction Model on Contract-based Services called CMS and the SDCalculation algorithm are proposed based on the paper [11, 17] to answer **RQ2**. The SD values are calculated based on the QoS and contract terms. To predict the missing time-aware personalized SD values, the

SDPred algorithm has been proposed, which has answered **RQ3**. This method collects SD values of different customers for various services at individual time intervals to build a customer-service-time tensor. By performing tensor factorization technique, the unknown SD values will be predicted by the feature matrices.

To verify our methods, several experiments have been conducted based on the data set WR-Dream released by [19]. Firstly, we extract 1000 services with similar functionalities from the data set. Then, we randomly assign two contract term values to each customer-specific and time-specific service. Afterwards, according to the generation customers' service usage history, time-aware customer preference distributions can be sensed by our methods. Besides, a customer-service-time tensor can be built by the SDCalculation algorithm. Then, the missing SD values can be predicted by the SDPred algorithm. Eventually, several comparison experiments have been conducted, which adequately illustrate the effectiveness of our algorithm. In addition, the impact of three parameters in the SDPred algorithm is studied, which provides more evidences to prove the superiority of our methods.

In summary, this paper makes the following contributions:

- We formally identify the time-aware customer preference sensing issue in the dynamic service market and adopt statistical method to answer the question.
- We propose a novel customer satisfaction model, which considers service contract and presents the SDCalculation algorithm to measure the customer satisfaction.
- We define the time-aware personalized SD prediction issue formally, and propose a novel algorithm SDPred to predict the missing values, which outweighs another three methods demonstrated by our experiments.

The remainder of the paper is organized as follows. Section 2 presents the customer satisfaction model on contract-based services. Section 3 introduces the time-aware customer preference sensing methods. Section 4 presents the time-aware satisfaction degree prediction. Section 5 discusses the experimental setting and results. Section 6 gives a review on the related work. Finally, Sect. 7 offers some concluding remarks and overviews the future work.

2 Customer Satisfaction Model on Contract-Based Services

Based on the introduction above, it is clear that service contract focuses on the non-functional properties and consists of several contract terms. The paper [9] divided contract terms into five types, namely Provider Obligations, Usage of Information, Warranties and Liabilities, Delivery Time, Price and Payment terms. However, we distinguish QoS from contract and describe them respectively. QoS are quantitative with numeric values associated with specific measurement units, e.g., price = 21\$. Nevertheless, contract terms are qualitative where the values are assumed to be defined by concepts, e.g., payment method = credit card. According to the research, QoS mainly includes price, response time, etc. Meanwhile, several terms have been summarized, including payment method, right of privacy, service coverage, etc.

The paper [11] tells us that each service has its own contract. By introducing contracts to services, we build a novel service model $S = (FP, QoS, Contract)$, where

FP represents the functional properties, and $Contract$ stands for a group of non-numerical contract terms defined as $Contract = \langle term_1, term_2, \dots, term_a \rangle$. In addition, each term can be described as $term_i = \langle name, operator, value \rangle$, where $name$ says its identifier, e.g. payment method, and $operator$ mainly includes set operators, e.g. “all”, “exactlyOne”, etc., and $value$ represents specific values of $term_i$. Meanwhile, QoS may contain several indicators, which can be expressed as $QoS = \langle q_1, q_2, \dots, q_b \rangle$. And $q_i = \langle name, operator, value \text{ unit} \rangle$, where $name$ also stands for identifier, e.g. price, however, $operator$ represents the numeric operators, e.g. $<$, $>$, $=$, etc., and $value$ means the numeric values, e.g. 200, $unit$ says the measurement unit, e.g. dollar.

$$SD = \begin{cases} 1, & \text{if } QoS_1 \wedge Contract_1 \\ \dots & \dots \\ SD_Value_i, & \text{if } QoS_i \wedge Contract_i \\ \dots & \dots \\ 0, & \text{if } QoS_w \wedge Contract_w \end{cases} \quad (1)$$

Provided that a stack of services has the same or similar functionalities, in the novel service model above, how to measure the level that a service may satisfy the customer preferences is becoming a critical challenge. Therefore, a new customer satisfaction model called CSM is proposed in our paper, which can be defined as Formula (1).

In the formula, SD stands for the satisfaction degree of a service satisfying customer requirements. Each set of $QoS_i \wedge Contract_i$ represents the customer expectation on QoS and contracts. If $QoS_i \wedge Contract_i$ is satisfied, then $SD = SD_Value_i$. In addition, $\forall i > j$, $SD_Value_i < SD_Value_j$, and $SD_Value_1 = 1$, $SD_Value_w = 0$. $\forall 1 < i < w$, $SD_Value_i \in (0, 1)$. Typically, $QoS_i = \langle q_{1,i} \wedge q_{2,i} \wedge \dots \wedge q_{a,i} \rangle$, $Contract_i = \langle terms_{1,i} \wedge terms_{2,i} \wedge \dots \wedge terms_{b,i} \rangle$.

Usually, the SD_value_i in the Formula (1) is provided by customers, however, due to the information asymmetry, customers cannot raise their requirements preferences clearly. Therefore, a novel approach called SDCalculation is proposed to calculate the SD value based on the corresponding QoS and contract values. For a service, this method calculates the satisfaction degree defined as SD_Q that its QoS satisfies the QoS constraints offered by the customer and the satisfaction degree called SD_C that its contract terms satisfy the contract preferences provided by the customer. Then, it is clear that $SD = SD_Q \times SD_C$. The method is presented in the algorithm 1, where $Num(t_i)$ is the function that count all the values of the term t_i , and Q and C represent the QoS values and the contract term values of the service, and W stands for the different preference weights to the indicators of QoS, besides, Q_i^{\max} is the maximum value of i th indicator of QoS in the whole candidate service set. QC and CP represent the QoS Constraints and the Contract preferences offered by the customer, respectively.

Algorithm 1. The SDCalculation Algorithm

```

Input:  $QC, CP, Q, C, W$ 
Output:  $SD$ 
1 Initialize  $SD_Q=0, SD_C=0;$ 
2 if each  $q_i$  in  $QC$  is satisfied by  $Q$  then  $SD_Q=1;$ 
3 else
4   for ( $i=0; i < QC.size; i++$ )
5      $SD_Q += W[i] \cdot \frac{|Q_i - QC_i|}{Q_i^{max}};$ 
6 while  $t_i$  in  $CP$  do
7   if  $t_i = C.t_i$  then  $SD_C = 1;$ 
8   else if the value of  $t_i$  is enumeration type
9     if  $t_i \cap C.t_i \neq \emptyset$ , then  $SD_C = \frac{Num(t_i \cap C.t_i)}{Num(t_i)}$ 
10  else  $SD_C = 0;$ 
11 else if  $t_i$  is contained by  $C.t_i$  then  $SD_C = 1;$ 
12  else  $SD_C = 0;$ 
13  $SD = SD_Q \times SD_C;$ 
14 output  $SD;$ 
    
```

3 Time-Aware Customer Preference Sensing

In this section, we try to sense the time-aware customer preferences based on customers’ service usage history at different time intervals. For a customer whose requirement is unclear, we analyze his selection standards or preferences in turn at each time interval, which reveals his requirements partly, and the process is shown in Fig. 3.

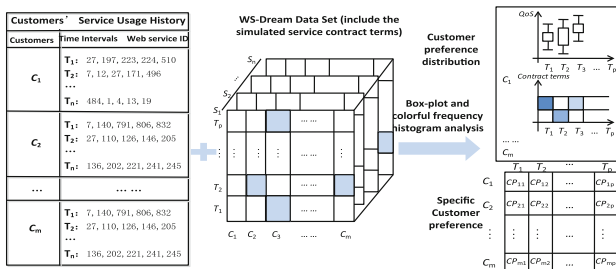


Fig. 3. Time-aware customer preference sensing process

To observe customer preferences for QoS or numerical contract terms, the box-plot method from mathematical statistics is adopted. Thus, the values of Q_1, Q_2, Q_3 , the Whisker upper limit and the Whisker lower limit need to be calculated respectively. Before this, the QoS values of services in the service usage history should be unified by Formula (2), which may reflect the actual preferences at individual time intervals.

$$P = \frac{s_{QoS_i} - QoS_{imin}}{QoS_{imax} - QoS_{imin}} \tag{2}$$

Here, QoS_{imax} , QoS_{imin} represent the maximum and minimum values of the i th indicator of QoS respectively. And s_{QoS_i} stands for the i th indicator value of the current service.

In terms of non-numerical contract terms, we employ the colorful frequency histogram to show the customer preference distributions. For a term value, we count the times of it appearing in the service usage history at each time interval, which will contribute to draw the histogram, where the deeper the color, the greater its frequency in the service usage history is, which will be illustrated by the Figures in Sect. 5.2.

Based on the analysis methods above, we can know the general distribution of the customer preferences against with time as well as the specific customer preferences at individual time interval. Here, to acquire the specific customer preferences, we adopt the Q_2 value as the QoS preferences at different time intervals, besides, we utilize the term with highest frequencies as the contract term preferences at each time interval.

4 Time-Aware Satisfaction Degree Prediction

4.1 Problem Formulation

Based on the introduction above, through integrating all the SD information from other customers, a three-dimensional customer-service-time tensor can be formed motivated by [19], which is shown in Fig. 4. The tensor can be split into massive slices with each one representing a time interval. Within a slice, each entry denotes a SD value of a service from a customer during the specific time interval. In this paper, we pay more attention to the study how to efficiently and precisely predict the missing entries in the tensor based on the existing entries. Thus, the problem of SD prediction for various users and services can be defined as follows. Typically, all the SD values are unified to the interval $[0, 1]$.

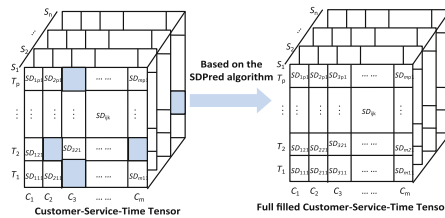


Fig. 4. Time-aware satisfaction degree prediction process

Definition: Suppose that the group of m customers, the set of n services and the set of p time intervals can be defined as C , S and T respectively, then a SD element is a quartet (i, j, k, SD_{ijk}) representing the SD value of service S_j for customer C_i at time interval t_k , where $i \in \{1, \dots, m\}, j \in \{1, \dots, n\}, k \in \{1, \dots, p\}$, and $SD_{ijk} \in \mathbb{R}$.

Besides, we regard the set of all triads (i, j, k) as Σ and define the set of all known triads (i, j, k) in Σ as Δ . If we consider a tensor $Z \in \mathbb{R}^{m \times n \times p}$ with each entry Z_{ijk} representing the SD value of S_j for C_i at time interval t_k , then the missing entries $\{Z_{ijk} | (i, j, k) \in \Sigma - \Delta\}$ should be predicted based on the existing entries $\{Z_{ijk} | (i, j, k) \in \Delta\}$.

4.2 Feature Matrix Learning

To acquire the feature matrices of customers, services and time, tensor factorization technique is adopted to fit a factor model to the customer-service-time tensor. The factorized customer-specific, service-specific and time-specific matrices are utilized to further predict missing entries, which is shown in Fig. 4. Therefore, a high-quality low-dimensional feature representation of customers, services and time can be derived by analyzing the customer-service-time tensor. The premise is that only a small number of factors influence the various SD values. In this paper, we consider a $m \times n \times p$ SD tensor consisting of m customers, n services and p time intervals. A low-rank tensor factorization approach is adopted to seek to approximate the SD tensor Z by a multiplication of r -rank factors [14], which is expressed as Formula (3).

$$Z \approx M \times_u C \times_s S \times_t T \quad (3)$$

Where $M \in \mathbb{R}^{r \times r \times r}$, $C \in \mathbb{R}^{m \times r}$, $S \in \mathbb{R}^{n \times r}$ and $T \in \mathbb{R}^{p \times r}$ are feature matrices. r is the number of features. Each column in C , S and T represents a customer, a service and a time interval, respectively. \times_u , \times_s and \times_t are tensor-matrix multiplication operators, e.g. $M \times_u C = \sum_{i=1}^r M_{ijk} C_{ij}$. M is set to the diagonal tensor. If $i = j = k$ then $M = 1$, otherwise $M = 0$.

Typically, in reality, each customer has invoked only a small portion of services and massive customers had not evaluated the services they had invoked, thus, $r \ll m, n, p$. Besides, customers cannot clearly raise expectations due to information asymmetry. Therefore, Z is usually very sparse. From the definition above, it is clear that the low-dimensional matrices C , S and T are unknown and need to be estimated.

$$\min_{C, S, T} F(Z, C, S, T) = \sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^p D_{ijk} (Z_{ijk} - \hat{Z}_{ijk})^2 + \alpha \sum_{i=1}^m \sum_{j=1}^r |C_{ij}|^2 + \beta \sum_{i=1}^n \sum_{j=1}^r |S_{ij}|^2 + \gamma \sum_{i=1}^p \sum_{j=1}^r |T_{ij}|^2 \quad (4)$$

To access the quality of tensor approximation, a loss function for evaluating the residue between the estimated tensor and the original tensor needs to be constructed. We define the optimization problem as Formula (4), which avoids the over fitting issue. Here, D_{ijk} is the indicator function that equals to 1 if customer C_i invoked and evaluated service S_j at the time interval t_k and equals to 0 otherwise.

In Formula (4), where $\alpha, \beta, \gamma > 0$ minimizes the objective function defined as the sum-of-squared-residue with quadratic regularization terms. Therefore, a local minimum of the objective function given by Formula (4) can be found by performing incremental gradient descent in feature vectors C_i , S_j , and T_k . And the corresponding formulas during performing incremental gradient descent are presented in Formula (5).

$$\begin{aligned}
\frac{\partial F_A}{\partial C_{iy}} &= 2 \sum_{j=1}^n \sum_{k=1}^p D_{ijk} (\hat{Z}_{ijk} - Z_{ijk}) S_j^T T_k + 2\alpha C_{iy}, \\
\frac{\partial F_A}{\partial S_{jy}} &= 2 \sum_{i=1}^m \sum_{k=1}^p D_{ijk} (\hat{Z}_{ijk} - Z_{ijk}) C_i^T T_k + 2\beta S_{jy}, \\
\frac{\partial F_A}{\partial T_{ky}} &= 2 \sum_{i=1}^m \sum_{j=1}^n D_{ijk} (\hat{Z}_{ijk} - Z_{ijk}) C_i^T S_j + 2\gamma T_{ky}.
\end{aligned} \tag{5}$$

$$\hat{Z}_{ijk} = D_{ijk} \sum_{y=1}^r C_{iy} S_{jy} T_{ky} \tag{6}$$

Based on the formulas above, the customer-specific, service-specific and time-specific feature matrices C , S and T can be learned. Hence, for the missing entry Z_{ijk} in the tensor, the SD value can be predicted by Formula (6).

Based on the analysis above, the algorithm name SDPred is summarized, which is presented in Algorithm 2. It also shows the iterative process for feature metrics learning. Firstly, matrices C , S and T are initialized with small random non-negative values. According to the update rules derived from Formula (5), the updating process is iterated until the algorithm converges to a local minimum of the objective function given in Formula (4). After the iteration, the matrices C , S and T can be acquired eventually. Then, the missing SD values can be predicted by Formula (6).

Algorithm 2. the SDPred algorithm

Input: $Z, r, \alpha, \beta, \gamma$

Output: \hat{Z}

1 Initialize $C \in \mathbb{R}^{m \times r}, S \in \mathbb{R}^{n \times r}, T \in \mathbb{R}^{p \times r}$ with small random numbers;

2 repeat

3 while $(i, j, k) \in \Delta$ do

4 $\hat{Z}_{ijk} = D_{ijk} \sum_{y=1}^r C_{iy} S_{jy} T_{ky}$;

5 while $(i, j, k) \in \Delta$ do

6 for ($q=1; q \leq r; q++$)

7 $C_{iq} \leftarrow C_{iq} - [(\hat{Z}_{ijk} - Z_{ijk}) S_j^T T_k + \alpha C_{iq}]$;

8 $S_{jq} \leftarrow S_{jq} - [(\hat{Z}_{ijk} - Z_{ijk}) C_i^T T_k + \beta S_{jq}]$;

9 $T_{kq} \leftarrow T_{kq} - [(\hat{Z}_{ijk} - Z_{ijk}) C_i^T S_j + \gamma T_{kq}]$;

10 until the converge;

11 while $(i, j, k) \in \Sigma - \Delta$ do

12 $\hat{Z}_{ijk} = D_{ijk} \sum_{y=1}^r C_{iy} S_{jy} T_{ky}$;

13 output \hat{Z} ;

5 Experiments and Comparisons

5.1 Experimental Setting

(1) Candidate Services. We adopt a released QoS data set <http://www.wsdream.net/> called WS-Dream. Firstly, 1000 Web services with same or similar functionalities have been extracted. Then, we collect the QoS information of these services invoked by 142 customers at 64 time intervals, including response time and throughput. Based on the data extracted, we simulate the contract term values of each Web service for each customer at each time interval, including payment method and right of privacy with the corresponding values {*Credit card*, *Applypay*, *Alipay*}, {true, false} respectively.

(2) Customer’s Service Usage History. Based on the reconstructed WS-Dream above, we simulate service usage histories of 142 customers during different time intervals with the format < customerID, timeIntervalID, wsID list, amount >, where the four dimensions stand for the customer ID, time interval ID, all the service IDs selected by the customer at the corresponding time interval and the invocation times, respectively.

5.2 Time-Aware Customer Preference Sensing

Based on the customers’ service usage history generated above, we adopt the methods presented in Sect. 3 to analyze the time-aware preference distributions of each customer. Here, we take it an example where customerID = 0 and customerID = 1. The analysis results are shown in Figs. 5 and 6.

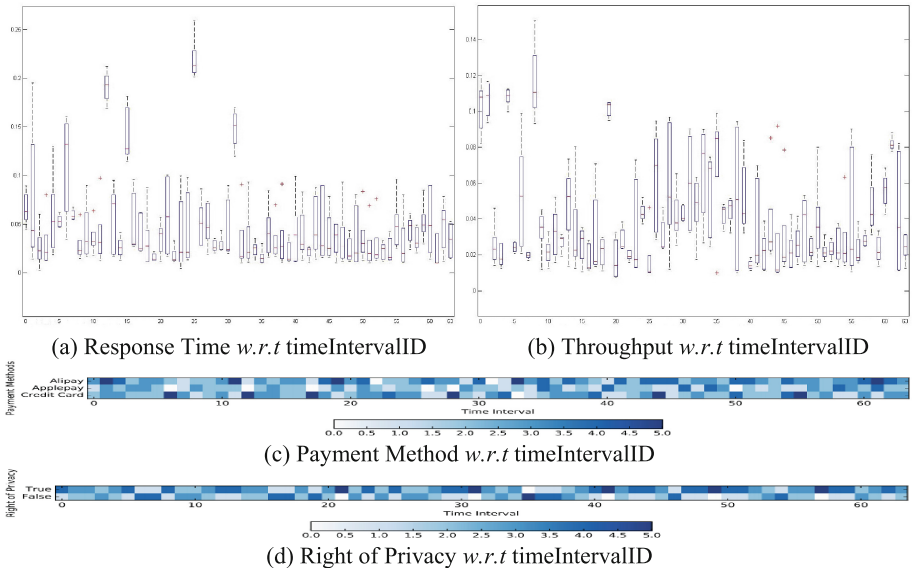


Fig. 5. Customer preference distributions based on customerID = 0

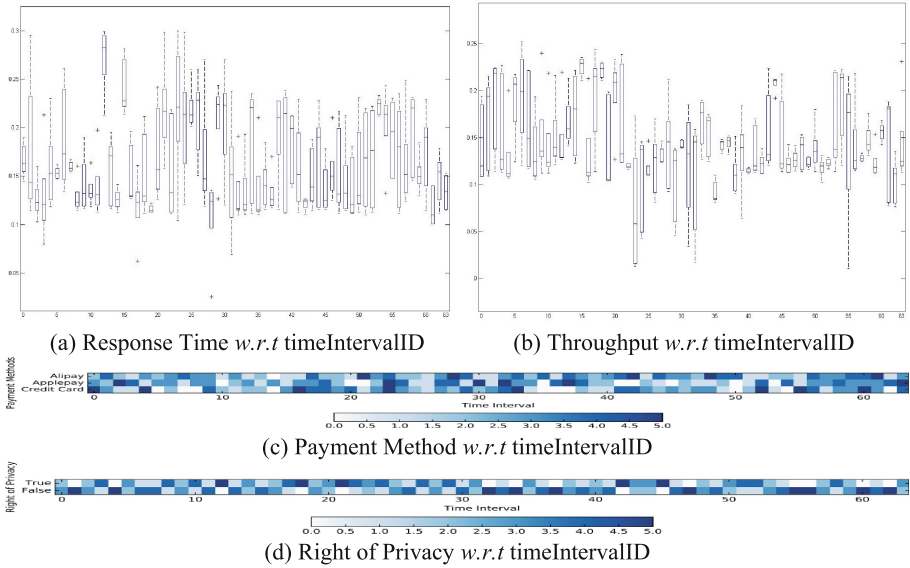


Fig. 6. Customer preference distributions based on customerID = 1

According to Fig. 5 (a), (b), it is clear that the preferences to QoS of the customer (ID = 0) have not changed significantly during the 64 time intervals, except for several cases, which may reflect the actual customer preference, e.g. the customer’s location changed, as a result, his preferences to response time and throughput varied as well. Figure 5 (c) shows the customer preferences to three payment methods at different time intervals. Obviously, the customer more prefers to *Alipay*. Figure 5 (d) tells us that the customer only takes notice of his privacy sometimes with the small time interval IDs. And he pays more attention to his privacy with time intervals changing. However, the time-aware preference sensing of another customer (ID = 1) is different from the former, which is shown in Fig. 6, where he had less rigorous constraints for response time and stricter constraints for throughput in general. His preferences to contract terms had some differences from the former. Likewise, the time-aware preferences of another 140 customers can be sensed, which illustrates the validity of our methods.

5.3 Verification of the SDPred Algorithm

From the analysis above, we can know the specific customer preferences at different time intervals. Then, the time-aware personalized SD values can be calculated and a customer-service-time tensor can be built.

To verify the SDPred method proposed in this paper, we have summarized several collaborative filtering methods as follows:

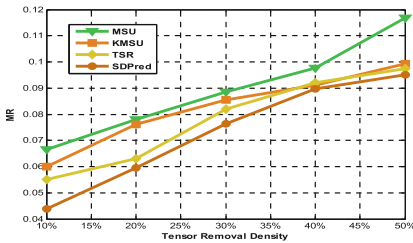
- (1) **MSU**(Most similar user): For a specific time interval t , if user U_i does not evaluate the service S_j , then MSU takes the corresponding SD value the most similar user of U_i keeps for S_j as the prediction of the missing SD value.
- (2) **KMSU**(Top-k most similar user): Similar to MSU, it identifies top-k most similar users and regards the average SD value of them at a time interval as the prediction.
- (3) **TSR**(Time-aware similar requirement): For a single user U_i , if he has not accessed the service S_j at a specific time interval t_x , then TSR tries to distinguish the most similar time interval t_y when the user requirement is similar to that of t_x . Afterwards, TSR takes the SD value corresponding to t_y as the prediction of the missing value at t_x .

In addition, in order to compare these methods with SDPred, two metrics have been proposed, including the Mean Residue (MR) defined as Formula (7) and the Root Mean Squared Residue ($RMSR$) expressed in Formula (8).

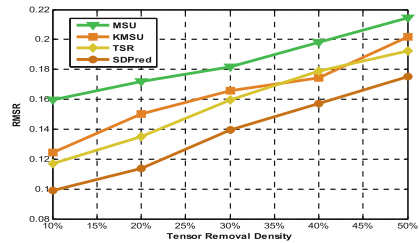
$$MR = \frac{1}{N} \sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^p |\hat{Z}_{ijk} - Z_{ijk}| \tag{7}$$

$$RMSR = \sqrt{\frac{1}{N} \sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^p (\hat{Z}_{ijk} - Z_{ijk})^2} \tag{8}$$

Besides, to evaluate the performance of different approaches above, some entries have been removed from the tensors, which will be compared with the prediction values. The tensors with missing SD values are in different densities, which are defined as the removal density. For instance, 20 % means that 20 % entries have been removed from the original tensor and will apply the remaining 80 % entries to predict the removed ones. The values of α, β, γ are tuned by performing cross-validation on the existing SD values. By massive attempts, we confirm that $r = 30, \alpha = \beta = \gamma = 0.002$. And in the KMSU method, $k = 5$. With different removal densities, these methods are compared with each other by the values of MR and $RMSR$, which are presented in Fig. 7.



(a) MR w.r.t. Tensor Removal Density



(b) $RMSR$ w.r.t. Tensor Removal Density

Fig. 7. Performance comparisons based on Tensor Removal Density

From Fig. 7, it is clear that the SDPred algorithm outperforms another three methods (lower MR and $RMSR$ values) under different tensor removal densities. With the *Tensor Removal Density* growing up, both MR and $RMSR$ of these methods increase gradually, which means that the more missing SD values exist, the less accurate the predictions of the four methods are. In general, the SDPred method has the best performance, then the followings are TSR, KMSU, MSU respectively, which adequately illustrate the effectiveness of our method. However, sometimes the KMSU method performs better than the TSR method. Therefore, if the tensor is much sparse, the SDPred will be recommended to acquire more precise predictions.

5.4 Impact of Three Parameters in the SDPred Algorithm

Here, we further focus on the impact of three parameters in the SDPred algorithm, including m , n and p representing the amount of customers, the number of services and the amount of time intervals individually. Suppose that *Tensor Removal Density* = 50 %, three experiments are set $n = 1000$, $p = 64$; $m = 142$, $p = 64$; $m = 142$, $n = 1000$, respectively. Then, the comparison results are shown in Figs. 8, 9 and 10.

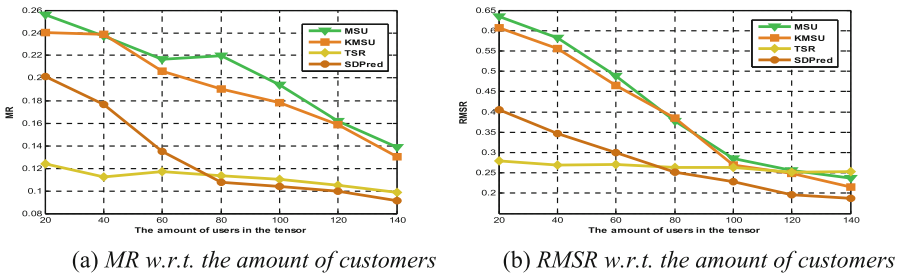


Fig. 8. Performance comparisons based on the amount of customers in the tensor

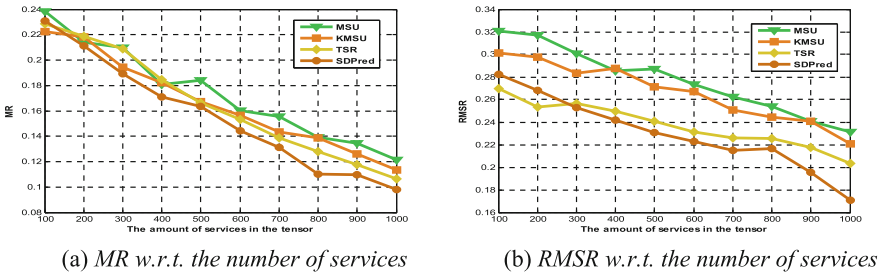
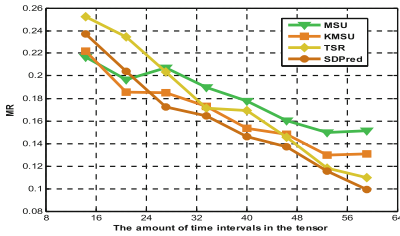
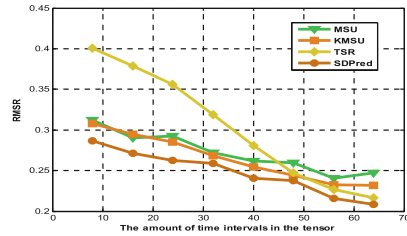


Fig. 9. Performance comparisons based on the number of services in the tensor

From Fig. 8, we can know that with the increase of m , MR and $RMSR$ of the three methods (including MSU, KMSU, SDPred) both decline significantly. However, MR and $RMSR$ of TSR go down slowly. Although TSR performs best with smaller values of m , SDPred can achieve higher prediction accuracy while m increasing. Figure 8 tells

(a) *MR w.r.t. the amount of time intervals*(b) *RMSR w.r.t. the amount of time intervals***Fig. 10.** Performance comparisons based on the amount of time intervals in the tensor

us that the amount of customers has greater impact on MSU, KMSU and SDPred along with smaller influence on TSR, which are consistent with their mechanisms.

Figure 9 clearly presents that the *MR* and *RMSR* corresponding to these methods also fall down while n increases. In addition, the former declines significantly shown in Fig. 9 (a), while the latter goes down gradually presented in Fig. 9 (b). It is convinced that SDPred has best performance basically followed by TSR, which is followed closely after by KMSU and MSU. However, they have fewer differences, which mean that the amount of services has little impact on these methods.

Based on Fig. 10, it is easy to know that in terms of these methods, both the *MR* and *RMSR* decline while p grows up. Besides, TSR has a remarkable decline and another three methods drop down non-significantly. Therefore, it clearly illustrates that the amount of time intervals has larger impact on TSR. However, SDPred can achieve higher prediction accuracy on the whole, which can prove its superiority.

Combined with Figs. 8, 9 and 10, it is obvious that regardless of various assignments of the three parameters (m , n and p), overall, the SDPred method performs best, which provides more evidences to illustrate the effectiveness of our method.

6 Related Work

Service contract has been widely studied from different perspective to describe a service comprehensively. A policy centered meta-model was presented in [4] to describe service contracts, which unified the policy [7], service level agreements [10], etc. In this model, each term corresponded to one non-functional property.

In addition, several methods to calculate the customer satisfaction degree had been proposed, including the method based on QoS [18] and the method based on QoS and contract [3]. However, [13] only considered the QoS, which cannot reflect the reality adequately. And the method in [17] may not be applied actually, since the SD values were provided by customers, which were difficult to be collected. In [3], a novel model to calculate customer satisfaction in the cloud was presented, which took the service level agreements into account. However, these methods have not considered the actual factor that SD values have varied with various time intervals changing.

Customer preference analysis has aroused wide attentions, including modeling user preference by mutual information in a statistical framework [6], the context-aware user

preference study [2]. Nevertheless, the important factor time has been weakened, which needs to be enhanced in the user preference analysis.

Besides, several researchers recognized information asymmetry in the service field [12]. Then, trust was proposed as the important source of value in service relationships characterized by high degrees of information asymmetry in [5]. And two types of trust-based approach for reducing information asymmetry were presented in [13].

7 Conclusions and Future Work

This paper proposes the customer satisfaction model on contract-based services. In the model, a novel SD calculation method is presented. Besides, we adopt the box-plot method and colorful frequency histogram to sense the time-aware customer preferences. Then, time-aware personalized SD values can be calculated, including several missing values due to information asymmetry. To predict the missing values, the SDPred algorithm is proposed, which is the first to research time-aware personalized satisfaction. Based on WS-Dream and the customers' service usage history, the verification and comparison experiments are conducted, which verify the effectiveness of our methods. And we further research the impact of parameter settings in the SDPred algorithm, which provides more evidence to illustrate the superiority of our method.

Some future work is summarized shown in Fig. 1, including (1) to research the changing trends of services over the time; (2) to further focus on the time-aware customer preferences; (3) to recommend services for customers based on the work (1), (2).

Acknowledgments. This work is supported by the Natural Science Foundation of China (No. 61272187, 61472106).

References

1. Alrifai, M., Risse, T., Nejd, W.: A hybrid approach for efficient web service composition with end-to-end qos constraints. *ACM Trans. Web* **6**(2), 1–29 (2012)
2. Campos, P.G., Díez, F., Cantador, I.: Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. *User Model. User-Adap. Inter.* **24**(1–2), 67–119 (2014)
3. Chen, J., Wang, C., Zhou, B.B., Sun, L., Lee, Y.C., Zomaya, A.Y.: Tradeoffs between profit and customer satisfaction for service provisioning in the cloud. In: *Proceedings of the 20th International Symposium on High Performance Distributed Computing*, pp. 229–238. ACM (2011)
4. De Paoli, F., Palmonari, M., Comerio, M., Maurino, A.: A meta-model for non-functional property descriptions of web services. In: *Proceedings of IEEE International Conference on Web Services*, pp. 393–400 (2008)
5. Gallouj, C.: Asymmetry of information and the service relationship: selection and evaluation of the service provider. *Int. J. Serv. Ind. Manage.* **8**(1), 42–64 (1997)
6. Jung, S.Y., Hong, J.H., Kim, T.S.: A statistical model for user preference. *IEEE Trans. Knowl. Data Eng.* **17**(6), 834–843 (2005)

7. Kamoda, H., Yamaoka, M., Matsuda, S., Broda, K., Sloman, M.: Policy conflict analysis using free variable tableaux for access control in web services environments. In: Proceedings of the Policy Management for the Web Workshop at the 14th International World Wide Web Conference, pp. 121–126 (2005)
8. Koychev, I., Schwab, I.: Adaptation to drifting user's interests. In: Proceedings of ECML Workshop: Machine Learning in New Information Age, pp. 39–45 (2000)
9. Lamparter, S., Luckner, S., Mutschler, S.: Semi-automated management of web service contracts. *Int. J. Ser. Sci.* **1**(3–4), 288–314 (2008)
10. Lewis, L., Ray, P.: Service level management definition, architecture and research challenges. In: Proceedings of the Global Telecommunications Conference, pp. 1974–1978 (1999)
11. Liu, P.F., Ma, C., Wang, Z.J., Comerio, M., Xu, X., Batini, C.: Generating global contract for composite services. In: Proceeding of IEEE International Conference on Service Science, pp. 9–16 (2015)
12. Nayyar, P.R.: Information asymmetries: a source of competitive advantage for diversified service firms. *Strateg. Manag. J.* **11**(1), 513–519 (1990)
13. Rajeswary, C.: A survey on efficient evolutionary algorithms for web service selection. *Int. J. Manage., IT Eng.* **2**(9), 177–191 (2012)
14. Rendle, S., Schmidt-Thieme, L.: Pairwise interaction tensor factorization for personalized tag recommendation. In: Proceedings of WSDM 2010, pp. 81–90 (2010)
15. Tian, C.H., Ray, B.K., Lee, J., Cao, R., Ding, W.: BEAM: a framework for business eco-system analysis and modeling. *IBM Syst. J.* **47**(1), 101–114 (2008)
16. Wai, L., Javed, M.: Modeling user interest shift using a bayesian approach. *J. Am. Soc. Inform. Sci. Technol.* **52**(5), 416–429 (2001)
17. Wang, Z.J., Wu, Q., Xu, X.F.: Value-added analysis of bi-lateral e-business services. In: Proceedings of International Conference on e-Business Engineering, pp. 185–192 (2012)
18. Zeng, L., Benatallah, B., Ngu, A.H., Dumas, M., Kalagnanam, J., Chang, H.: QoS-aware middleware for web services composition. *IEEE Trans. Software Eng.* **30**(5), 311–327 (2004)
19. Zhang, Y., Zheng, Z., Lyu, M.R.: WSPred: a time-aware personalized QoS prediction framework for Web services. In: IEEE 22nd International Symposium on Software Reliability Engineering, pp. 210–219 (2011)