Web Services Evaluation Predication Model based on Time Utility

Guisheng Yin, Xiaohui Cui, Yuxin Dong, Jianguo Zhang
College of Computer Science and Technology, Harbin Engineering University, Harbin, China
Email: {yinguisheng, cuixiaohui}@hebru.edu.cn

***Abstract*—Current predication models fail to consider the time utility of the web services evaluation predication and treat the different historical ratings in the same way. To solve this problem, we put forward web service evaluation predication model based on time utility. In the model, naïve quantification method and complex quantification method are proposed to achieve the distinct and proper time utility for the services evaluation predication procedure. Then, the quantification results are used to optimize the length of the predication windows. Also, feedback control strategy is involved to enhance the robust of the model when facing malicious. Experimental results shows our model would calculate the proper time utility and obtain the lower predication error compared with current predication models. Feedback control strategy is an effective method to reduce the impact of malicious ratings and guarantee the lower predication error compared with the model without the feedback control strategy.**

***Index Terms*—web services, time utility, quantification of time utility, predication, feedback control**

I. Introduction

The evaluation predication is the principal application of the web services system. It could help the users to achieve the available web services concerned with their requests. The traditional predication models, such as Quality of service (Qos) in Ref. [1,2,3,4,5], Web services Evaluation System (WES) in Ref. [4], and K Nearest Neighbor (KNN) in Ref. [7,8,9] fail to analyze the time utility of the historical ratings when evaluating web services. In reality, the recent ratings express more valuable than the ratings in the past. Therefore, it is considerable to involve the decay process of the time utility into the predication procedures of the web services evaluation. In the past decade, some researches related with the time utility have been done in other fields of computer sciences. In Machine Learning, Koychev deemed the time utility of ratings would decade gradually with time, and this decay could be presented by the liner function in Ref. [11]. In Recommendation Systems, Ref. [13] and [14] assumed that core function would be the suitable description of the decay process. In Concept Drift System, several scholars applied the exponential function in Ref. [15] to describe the decay phenomenon.

The above models of the decay process of the time utility are hard to directly apply to the web services evaluation predication system since there are some problems to be solved. Firstly, Current models usually assumed that the time utility would decay in a static ratio even for different web services. According to the study of Indre Zliobaite, the ratings are the belief of the subjects to expect the evaluated objects to accomplish a task in Ref. [16]. This belief would be distinct for different web services. It is appropriate to use various decay procedures to describe the time utility for different web services. Secondly, no existing works have ever mentioned how to incorporate the time utility to optimize the predication procedures of the web services evaluation system. Thirdly, the predication procedures heavily rely on the historical ratings, while the web services evaluation system is easy to be attacked by the malicious ratings.

To solve the above problems, we propose a web services evaluation predication model based on time utility (WSEPM-TU). In this model, complex quantification method of the time utility is proposed to unfold the distinct quantification of different web services. Then we apply the quantification results to optimize the length of the predication windows, as to enhance the performance of WSEPM-TU. Finally, WSEPM-TU supplies the feedback control strategy to reduce the side impact of the malicious ratings.

The remainder of the paper is organized as follows: Section II states the architecture of WSEPM-TU. Section II designs the naive quantification method and the complex quantification method of the time utility. Section IV describes the optimization method for the length of the predication windows. Section V provides the feedback control strategy of the malicious ratings. Section VI presents the experiments to analyze the performance of WSEPM-TU. Section VII concludes the paper.

II. Architecture of WSEPM-TU

Web services evaluation predication system (WSEPS) is a prototype system of WSEPM-TU. WSEPS adopts the predication procedures to achieve the predication of the web services evaluation based on the quantification of the time utility. The feedback control strategy filters out the malicious ratings. Fig. 1 shows the architecture of WSEPS.

The detail predication procedures of WSEPS are shown as follows.

(1) Through the user's interface, WSEPS wraps the predication requests of the web services into the request entities and delivers them to the predication model.

doi:10.4304/jsw.7.12.2717-2725

(2) The predication model extracts the identifiers of the web services in the request entities and delivers them to the time utility model.

(3) The time utility would search from the historical ratings table and calculate the current time utility of different web services. Then it returns the quantification results to the predication model.

(4) The predication model applies the quantification results to optimize the length of the predication windows and estimates the evaluation predication of the web services. Then it returns evaluation results to the user's interface.

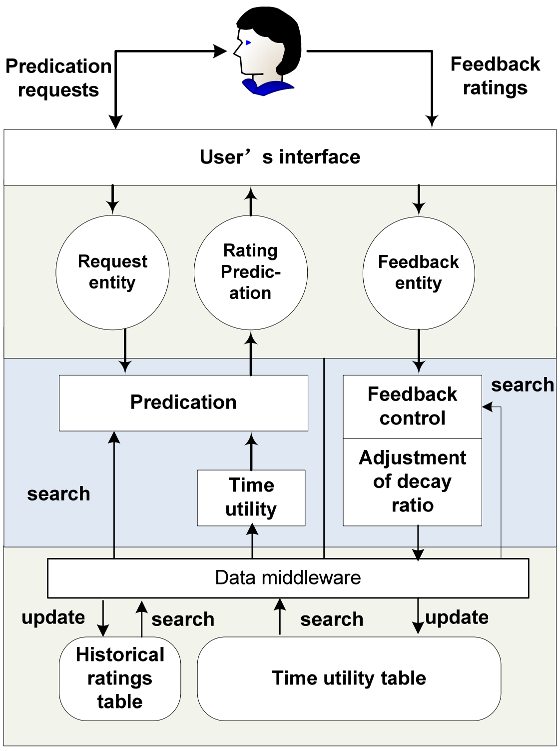

Figure 1. Architecture of WSEPS

The detail feedback procedures of WSEPS are shown as follows.

(1) Through the user's interface, WSEPS wraps the user's feedback ratings of the web services into the feedback entities and delivers them to the feedback control model.

(2) The feedback control model resolves the service identifiers and the users' ratings, and investigates whether the ratings are malicious. If the ratings are malicious, the feedback control model adjusts the ratings, and stores them in the historical ratings table. If the ratings are valid, the feedback control model would store the ratings directly into the historical ratings table.

In the procedures of predication and feedback, the time utility model, the predication model, the feedback control model and the adjustment of the decay ratio model are the main focus of our papers.

III. Quantification Method of the Time Utility

System Dynamics is a main way to analyze the complex sequential system. It adopts the quantitative and qualitative methods to confirm the cause and effect of different system factors and constructs the dynamic system equations. The general steps of System Dynamics are Constructing the cause and effect relationship diagram among different system factors, transforming the cause and effect relationship diagram into the system flow diagram, analyzing the characteristics of the variables in the flow diagram to achieve the difference equations, transforming the difference equations to the differential equations, solving the differential equations to obtain the primitive functions. The decay procedures of the time utility can be deemed as a whole system affected by several system factors. To achieve a proper quantification method of the time utility, we use System Dynamics.

A. Naive Quantification Method

The naive quantification system of the time utility assumes that all the decay procedures of the time utility are identical for the web services. In the naive quantification system, the system factors include *time_utility*, *decay_speed* and *decay_ratio*. According to the natural decay characteristics of the time utility, increment of *time_utility* leads to the growth of *decay_speed* of the time utility per unit time, and it means the causal relationship between *time_utility* and *decay_speed* is positive. In turn, the increment of *decay_speed* leads to the decline of *time_utility*, and it means the causal relationship between *decay_speed* and *time_utility* is negative. *Decay_ratio* is a constant in this system. The cause and effect diagram of the naive quantification system is shown in Fig. 2.

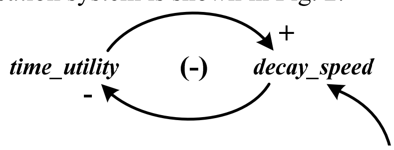

Figure 2. Cause and effect diagram of the naive quantification system

There is a first order negative causal loop in Fig. 2. We assume that the direction into *time_utility* is positive and it can be inferred that *decay_ratio* is less than 0. According to System Dynamics, *time_utility* is a level variable, *decay_speed* is a rate variable and *decay_ratio* is an auxiliary variable. Fig. 3 shows the system flow diagram of the naive quantification system.

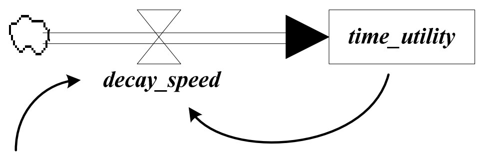

Figure 3. System flow diagram of the naive quantification system

If assuming *J*, *K* and *L* as the sequential time points and *DT* expresses a variance of the sequential time points , the dynamic equations of the naive quantification system are shown as (1) and (2).

$$time\_utility.K=time\_utility.J-decay\_speed.JK*DT. \quad (1)$$

*decay_speed.***KL**= *time-utility.***K** * *decay_ratio.* (2)

The equivalent differential equation of (1) and (2) is (3).

d*time_utility*/**d***t* = *time_utility* * *decay_ratio.* (3)

The primitive function to describe the dynamic characteristics of the naive quantification system is (4).

$$time\_utility = time\_utility_0 * e^{decay\_ratio * t}. \quad (4)$$

In our papers, $time\_utility_0$=1 for all the time utility would decay from 1. To simplify the expression of different system factors, we set *X(t)=time_utility* and *decay_radio=K*. The naive quantification method of the time utility is shown as (5).

$$X(t) = e^{Kt}. \quad (5)$$

The function image of (5) is a curve converging to 0, which meets hypothesis of the naive quantification system. In the web service evaluation system, it is unreasonable to use same decay procedure to describe the time utility of all the web services. A proper way is to adjust the naive quantification method to complex quantification by assigning different decay ratios to different web services, as to provide distinct quantification results.

B. Complex Quantification Method

The complex quantification method of the time utility takes the frequency of users' ratings to affect the decay ratio as to unfold the distinct quantification results for different web services.

When adjusting the decay ratio, we involve the psychological phenomenon of the memory enhancement. Based on the experiments, relearning would enhance the belief and start a new naive decay procedure of the time utility in a lower decay ratio compared with the prior decay procedure. If the time utility of the web services is the object to remember, the whole decay procedures of the time utility are the accumulation of the sequential naive decay procedures. Fig. 4 indicates a general decay procedure of the time utility for a web service.

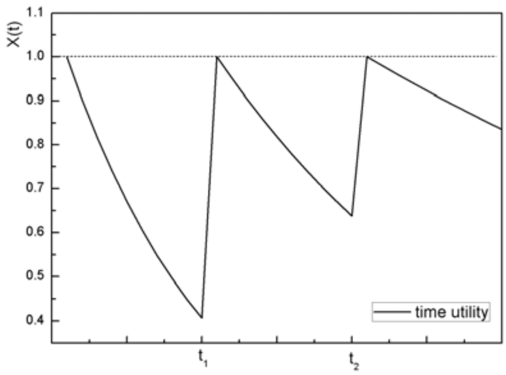

Figure 4. General decay procedure of the time utility

In order to calculate the decay ratio, we select two sequential decay procedures. Assuming the last time point of adjusting decay ratio as t_m, WSEPS receives the user's feedback rating and the decay ratio would change from K_m to K_n at the time point t_n. We make the curve of K_m and K_n share the same starting point to achieve the numeric relationship between K_m and K_n as shown in Fig.5.

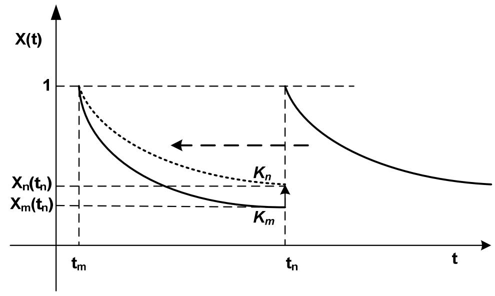

Figure 5. Neighbor naive procedures of the time utility

In Fig. 5, $X_n(t_n)$ and $X_m(t_n)$ represent the time utility curve. At the time point t_n, $\{X_n(t_n)\text{-}X_m(t_n)\}$ means the adjustment degree due to the user's feedback. $\{1\text{-} X_m(t_n)\}$ is the upper bound of the adjustment degree. If δ is the adjustment percentage, the relationship between the $\{X_n(t_n)\text{-}X_m(t_n)\}$ and $\{1\text{-} X_m(t_n)\}$ can be depicted by (6).

$$[1 - X_m(t_n)] / [X_n(t_n) - X_m(t_n)] = \delta. \quad (6)$$

The larger δ is, the fewer effects the users' rating are. Generally, δ is an integer more than 1. Using X(t) in (5) replace X(t) in (6), we would calculate the adjusted decay ratio relationship by (7).

$$K_n = (\ln(1 + (\delta - 1)e^{-k_m(t_n - t_m)}) - \ln\delta) / (t_m - t_n). \quad (7)$$

By (7), if we knowing the initial decay ratio K_0, the decay ratio of the arbitrary procedures can achieve. For a specified web service, assuming the time points sequence of the adjusted decay ratio as $t=\{t_0,t_1,...\}$ and K_m indicates the decay ratio between the neighbor time points(named t_m and t_n), we could use (8) to calculate the complex quantification results of the time utility.

$$X_m(t) = e^{-K_m(t - t_m)} \; t \in [t_m, t_n]. \quad (8)$$

In WSEPS, the time utility model utilizes (8) to calculate the time utility. Meanwhile, the adjustment of the decay ratio model utilizes (7) to update the records of the decay ratio in the time utility table.

IV. Predication of the Web Services Evaluation

In current predication model, KNN is the most common method to fusion the ratings. However, the length of predication windows in KNN should be artificially predefined. The unreasonable length of the predication windows would affect the performance of predication process. In WSEPM-TU, we make use of the quantification results of complex quantification method to optimize the length of predication windows (abbr. *pre_win*).

In the predication system, the system factors includes: the *decay_speed* and the max length of the predication windows (abbr. *max_win*).

Using the time point (named t) as the intermediate variable, we analyze the cause and effect relationship between the *decay_speed* and *pre_win* as follows.

(1) The causal relationship between t and *pre_win*. We assume Δt=t-t_m, while t_m and t indicate the starting point of the current decay procedure and current time point respectively. By the decreasing property of (8), the larger Δt is, the lower $X(t)$ is. To guarantee the reliability of the predication, we need more historical ratings, vice versa. Therefore, *pre_win* $\propto \Delta t$.

(2) The causal relationship between $X'(t)$ and *pre_win*. By the graph of $X'(t)$, the more Δt is, the larger $X'(t)$ is. Therefore, $\Delta t \propto X'(t)$.

For $\propto$ is an equivalence relation, *pre_win* and $X'(t)$ is a positive causal relation under the condition of *pre_win* $\propto \Delta t$, $\Delta t \propto X'(t)$, *pre_win* $\propto X'(t)$ and $X'(t) \propto$ *pre_win*. Fig. 6 shows the cause and effect diagram of the predication system.

Set the direction into the review window as positive, then the *decay ratio* is more than 0.

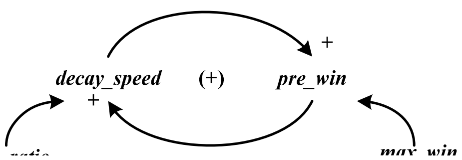

Figure 6. Cause and effect diagram of the predication system

According to the characteristics of the system factors, *max_win* indicates the upper bound of the predication windows, which is a constant. *var_win* is an auxiliary variable. Fig.7 shows the system flow diagram corresponding to Fig.6.

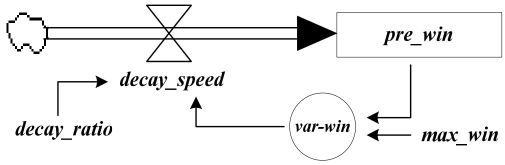

Figure 7. System flow diagram of the predication system

If assuming J, K and L as the sequential time points and DT expresses the variance of the sequential time points, the dynamic equations of the predication system are shown by (9), (10) and (11).

$$pre\_win.K=pre\_win.J+decay\_speed.JK*DT. \tag{9}$$

$$var\_win.K = max\_win - pre\_win.K. \tag{10}$$

$$decay\_speed.KL= var\_win.K * decay\_ratio. \tag{11}$$

Assuming $pre\_win\,|_{t=0} = 0$, we solve (9)-(11) and gain the equivalent function as shown by (12).

$$pre\_win=max\_win*(1-e^{-dcay\_ratio * t}). \tag{12}$$

Unify the expressions of the arguments in (12). n:= *pre_win*. N := *max_win*. The length of the predication windows is described by (13).

$$n=\lceil N(1-X_m(t))\rceil \quad t\in[t_m,t_n]. \tag{13}$$

In WSEPM-TU, the predication model searches the same mount historical ratings in accordance with (13). Assuming $B=\{b_1,b_2,\ldots,b_n\}(n <= N)$ is the historical ratings searched by WSEPM-TU, we could return the expectation of B as the evaluation predication.

V. Feedback Control Strategy

The feedback ratings are the main references for the following predication procedures of WSEPM-TU. If the system is lack of the feedback control strategy, some problems would emerge.

(1) Malicious slander ratings. Users provide the low ratings to slander the performance of a specified web service on purpose, as to enhance the predication of other web services.

(2) Malicious bidding ratings. Users provide the high ratings to bid up the performance of a specified web service on purpose, as to reduce the predication of other web services.

Both above malicious ratings would have side impact on the following predication procedures. Consequently, the feedback control strategy is the indispensible component for a robust web services evaluation predication system.

In WSEPM-TU, we treat some amounts of recent ratings as the sample data set (named Y) of the whole ratings set. Based on the sample data set, we estimate the relative confidence interval of whole ratings set by statistic of Y. If the feedback ratings are not in the relative confidence interval, the ratings would be converted to the random numbers in the relative confidence interval and stored in the historical ratings table. If the feedback ratings are in the relative confidence interval, the ratings will be directly stored in the historical ratings table. The calculation process of the relative confidence interval is shown as follows.

(1) Calculate the confidence interval of the whole ratings set. According to Central Limit Theory, the whole ratings set are normal distribution. Though we have no idea of the variance (named σ^2) of the whole ratings set, the variance (named S^2) of Y is an unbiased estimation of σ^2. By the relationship between t-distribution and normal distribution by (14):

$$T=\overline{Y}-\mu\big/(S/\sqrt{n}) \sim t(n-1). \tag{14}$$

The confidence interval of the whole ratings set is (15) under the confidence level of 1-α.

$$[\overline{Y}-t_{\alpha/2}(n-1)S\big/\sqrt{n},\overline{Y}+t_{\alpha/2}(n-1)S\big/\sqrt{n}]. \tag{15}$$

In (15), $t_{\alpha/2}$(n-1) is the $\alpha/2$ quantile of $t(n-1)$ distribution. $\alpha=5\%$ means the malicious ratings are the small probability events.

(2) Construct the relative confidence interval from the confidence interval. If there is no relative confidence interval, the feedback control strategy would be too strict to filter out the ratings when the system received several similar ratings concerned with (15). Therefore, we extend the confidence interval to the relative confidence interval for reducing the impact of over-fitting. In WSEPM-TU,

ratings are range from m to n. The relative confidence interval is described by (16). In (16), a and b mean the lower and the upper bound of the confidence interval of the whole ratings set.

$$\left[min\{\lfloor a-0.1\times(m-n+1)\rfloor, m\}, min\{\lceil b+0.1\times(m-n+1)\rceil, n\}\right]. \quad (16)$$

VI. Experiments and Analysis

A. Data Preparation

To evaluate the performance of our model, we use the ratings of application in App Store [18]. App store is the most mature market and rating platform for software. The way to bind credit card guarantees the reliability of the ratings in App Store.

By Search API we capture the 100,000 ratings of application in the form of json. Based on the preprocessing to the raw ratings, we choose the 5600 sequential ratings of application 105, 661 and 1084 as data sets (named D_1, D_2 and D_3 respectively) for the following experiments. The statistics of these 3 datasets are shown in table I. *Me*, *Var*, *Avg* and *IQR* mean *Median*, Variance, Average and Inter-quartile range respectively.

TABLE I.
STATISTICS OF SIMULATION DATASETS

	Num	*Me*	*Var*	*Avg*	*Min*	*Max*	*IQR*
D_1	5600	3.00	1.249	3.38	1	5	1
D_2	5600	3.00	0.932	3.41	1	5	1
D_3	5600	4.00	0.769	3.96	1	5	2

B. Evaluation Metrics and Simulation Parameters

We use Mean Absolute Error (MAE) in [19] to measure the performance of various predication models by (17). In (17), U is the length of data segment, $b(n)$ is the predicated value and $b^p(n)$ is the standard value. The lower MAE is, the better the predication models perform. Table II presents the simulation parameters of WSEPS.

$$MAE = \sum_{n=1}^{U} \left| b(n) - b^p(n) \right| \Big/ U \cdot \quad (17)$$

TABLE II.
PARAMETERS OF WSEPS

Name	Meaning	Value
$\|Y\|$	feedback sample	30
K_0	Initial decay ratio	1.0
N	*max_win*	10

C. Experimental Procedures

We adopt incremental learning to do the experiments. The procedures are shown as follows.

(1) Select the top-|Y| ratings from the data sets by the ascending order of the ratings' time stamp and put the selected ratings into the historical ratings table as the initial records.

(2) For the each rating left in the data sets, we treat them as the predication requests of the evaluation and carry out the different predication models to achieve the predicated evaluation.

(3) Measure the variance by (18) and insert the feedback ratings into the historical ratings table. Continue to go to step 2 until the last record in the data sets.

D. Parameters Analysis

This experiment displays the impact of various settings of δ. When δ is equal to 10, 50, 100 and 300, Fig. 8, 9 and 10 indicate the MAE analysis of WSEPM-TU on D_1~D_3. U=560.

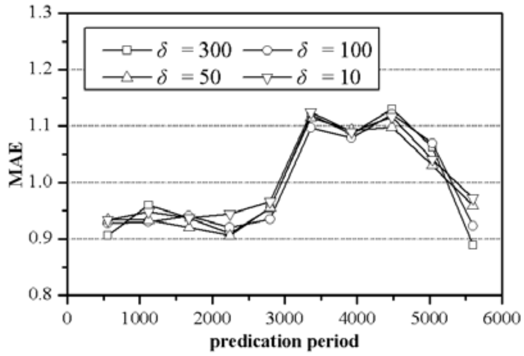

Figure 8. MAE analysis of various δ settings on D_1

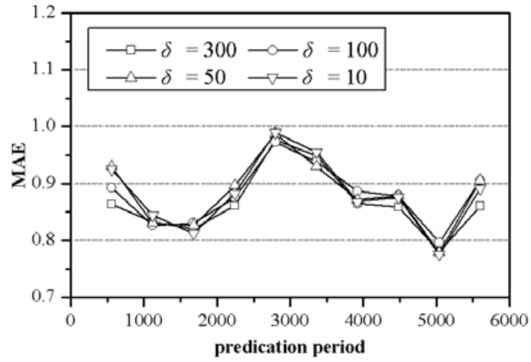

Figure 9. MAE analysis of various δ settings on D_2

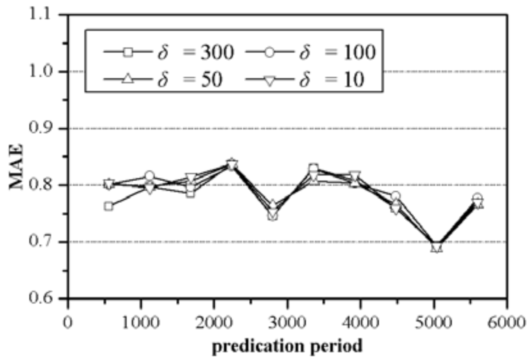

Figure 10. MAE analysis of various δ settings on D_3

In Fig. 8, the MAE values fluctuate slightly in a small range. For instance, when $\delta = 300$, the maximum of the MAE values emerges at the predication periods from 3921 to 4480 and the minimum of the MAE values arise in the predication periods from 5041 to 5600. In other predication periods, the MAE values maintain at 0.99 ± 0.1. Meanwhile, the average of the MAE values keeps at 0.88 ± 0.1 and 0.78 ± 0.1 on D_2 and D_3 respectively, as shown in Fig. 9 and Fig. 10.

According to the experimental results, the trends of fluctuations are similar under the various settings of δ and the fluctuated range is [-0.1, +0.1]. In conclusion, the performance of WSEPM-TU is not related to settings of δ. In the following experiments, we set δ=50.

E. Performance of Various Predication Models

By quantification methods of the time utility presented in [15] and [17], we construct a web service evaluation predication models based on the exponential function (WSEPM-E) and analyze the performance of WSEPM-E and WSEPM-TU on the data sets. WSEPM-E uses static quantification procedures and fails to consider the distinct and dynamic of the time utility.

Fig. 11, Fig. 12 and Fig. 13 indicate the MAE values of WSEPM-TU and WSEPM-E with different static decay ratios (WSEPM-E(0.5), WSEPM-E(1) and WSEPM-E(2)) on the data sets. U=30.

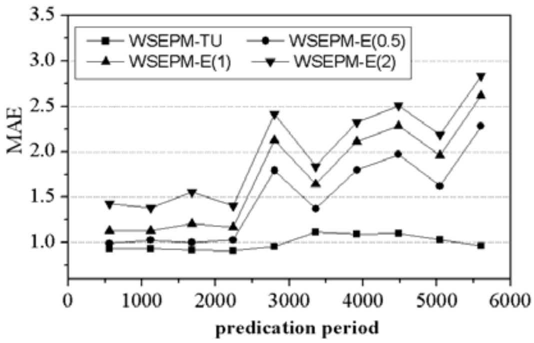

Figure 11. MAE Analysis of WSEMP-TU and WSEPM-E on D_1

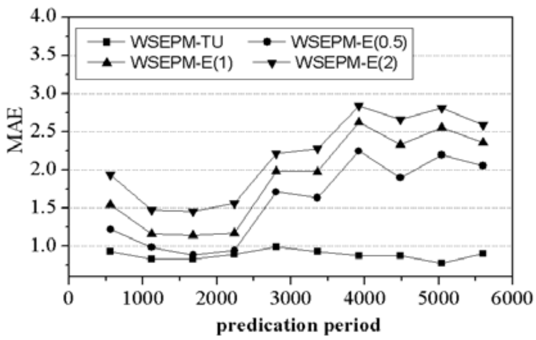

Figure 12. MAE Analysis of WSEMP-TU and WSEPM-E on D_2

In Fig. 11, the MAE values preserve at 0.99 ± 0.1 among all the predication periods. For WSEPM-E, on one hand, its MAE values would fluctuate with the predication periods and appear a rising trend. For instance, the MAE value of WSEPM-E (0.5) is 0.99002 at the predication periods from 1 to 560 and increases to 2.28268 at the predication periods from 5046 to 5600. On the other hand, owing to the lack of distinct quantification procedures for various web services, WSEPM-E (0.5) uses the static decay ratios to simulate decay procedures of the time utility, which performs poorly. The analysis of MAE on D_2 and D_3 would be similar as it is shown in Fig. 12 and Fig. 13.

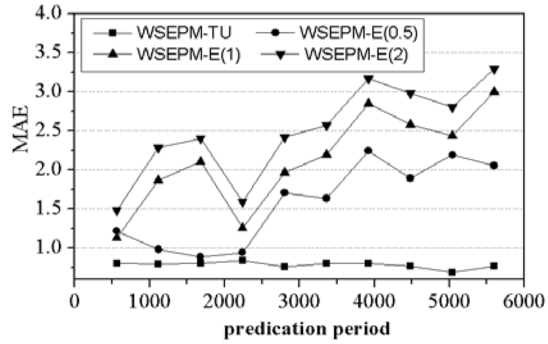

Figure 13. MAE Analysis of WSEMP-TU and WSEPM-E on D_3

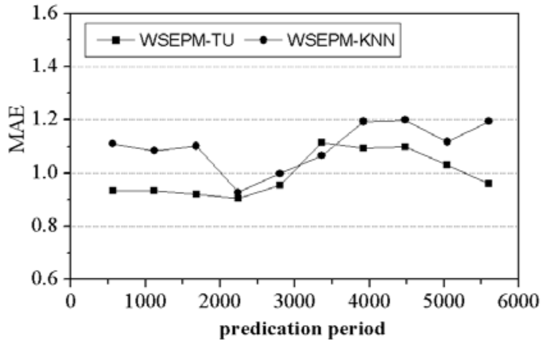

Figure 14. MAE analysis of WSEPM-TU and WSEPM-KNN on D_1

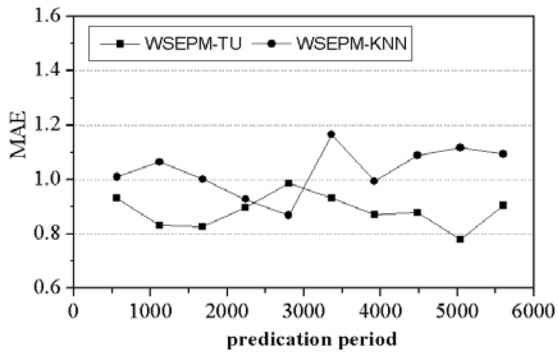

Figure 15. MAE analysis of WSEPM-TU and WSEPM-KNN on D_2

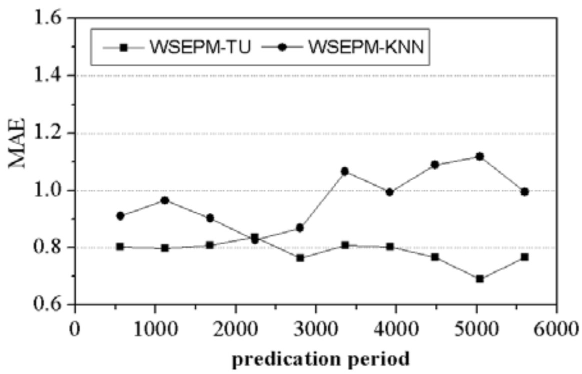

Figure 16. MAE analysis of WSEPM-TU and WSEPM-KNN on D_3

According to the experimental results, WSEPM-E fluctuates in a large range and performs more poorly than WSEPM-TU. In conclusion, WSEPM-TU would better simulate the time utility and perform better than the models with static quantification methods.

WSEPM-KNN [7-9] uses the static length of the predication windows to predicate the web services evaluation, while WSEPM-TU uses the dynamic length of the predication windows. This experiment shows the performance of WSEPM-KNN and WSEPM-TU.

Fig. 14, Fig. 15 and Fig. 16 indicate the MAE values of WSEPM-KNN (n=10) and WSEPM-TU on the data sets. U=560.

In Fig. 14, the MAE values of WSEPM-TU and WSEPM-KNN both fluctuate with the predication periods. For WSEPM-TU, the maximum and the minimum of the MAE values are 0.90536 and 1.11518, which appear at the predication periods from 1168 to 2240 and the periods from 2800 to 3360 respectively. For WSEPM-KNN, the maximum and the minimum of the MAE values are 0.92679 and 1.9929, which appears at the predication periods from 1680 to 2240 and the periods from 2920 to 4480 respectively.

Only at predication periods from 2800 to 3360, the MAE values of WSEPM-TU are 4.61% more than WSEPM-KNN. At other predication periods, WSEPM-TU gains the lower MAE values than WSEPM-KNN. Results are similar at D_2 and D_3, as shown in Fig. 15 and Fig. 16.

According to the experimental results, WSEPM-KNN could not properly reach the users' expectation, and consumes more computational sources than WSEPM-TU. In conclusion, WSEPM-TU utilizes the quantification results of the time utility to effectively optimize the length of the predication windows and show the better performance on various data sets.

F. Feedback Control Strategy Analysis

This experiment is to analyze the performance after introducing the feedback control strategy of the malicious ratings. As a comparison, we remove the feedback control strategy from WSEPM-TU and allow the malicious ratings to store in the historical ratings table directly.

To express the distributed change with the impact of the feedback control strategy, we utilize the box-plot to describe the statistics from D_1 to D_3. In the box-plot, the middle line of boxes means Median, the upper and lower bound line is the maximum and the minimum and the isolated points are the malicious data. In our experiments, each box possesses 560 ratings.

Fig. 17, Fig. 18, Fig. 20, Fig. 21, Fig. 23 and Fig. 24 show the distributed change of the historical ratings after running WSEPM-TU with and without the feedback control strategy. Fig. 19, Fig. 22 and Fig. 25 indicate the MAE values of WSEPM-TU with and without feedback control strategy.

In Fig. 17, the overall ratings without the feedback control strategy are distributed from 3 to 4. There are 6 boxes with the isolate points among all the predication periods. In Fig. 18, there are only 3 boxes with the isolate points, and it unfolds that the feedback control strategy would effectively filter out the malicious ratings. Meanwhile, the feedback control strategy has no impact on the valid ratings for the box shapes between Fig. 17 and Fig. 18 are similar. In Fig. 19, WSEPM-TU with the feedback control strategy performs better than the model without the feedback control strategy.

For the experiment results on D_2, the feedback control strategy would filter out the malicious ratings and provide reliable historical ratings.

For the experiment results on D_3, it possesses more malicious ratings as shown in Fig. 23. After the feedback controlling, the number of the malicious ratings reduce as it is shown in Fig. 24.

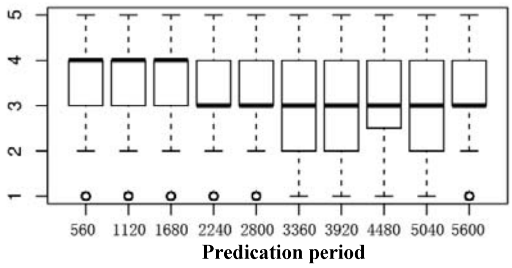

Figure 17. Data distribution on D_1 without the feedback control

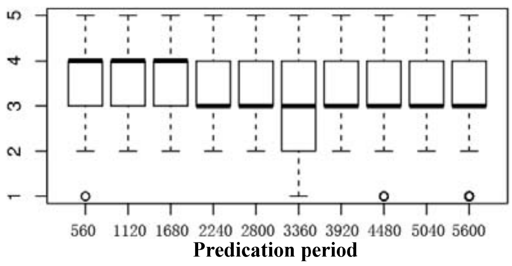

Figure 18. Data distribution on D_1 with the feedback control

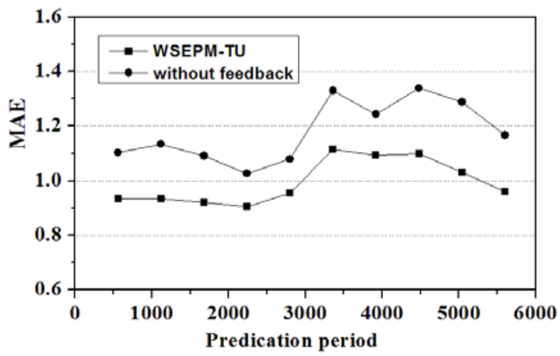

Figure 19. MAE analysis on D_1 with various feedback control strategies

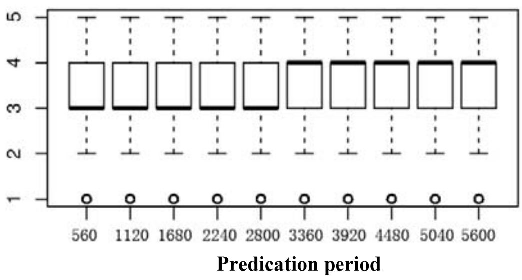

Figure 20. Data distribution on D_2 without the feedback control

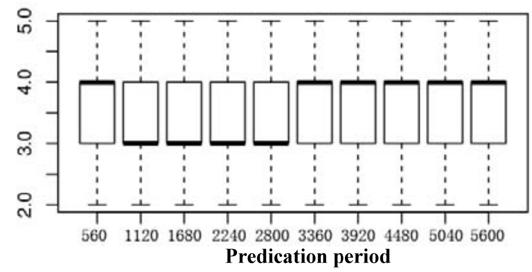

Figure 21. Data distribution on D_2 with the feedback control

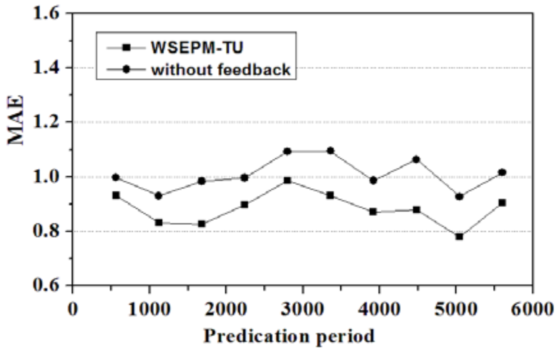

Figure 22. MAE analysis on D_2 with various feedback control strategies

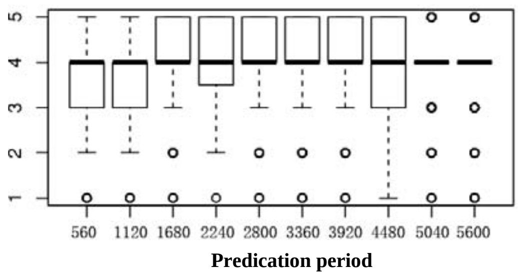

Figure 23. Data distribution on D_3 without the feedback control

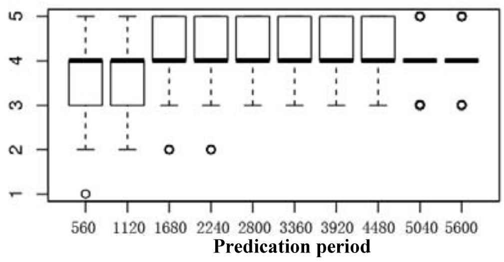

Figure 24. Data distribution on D_3 with the feedback control

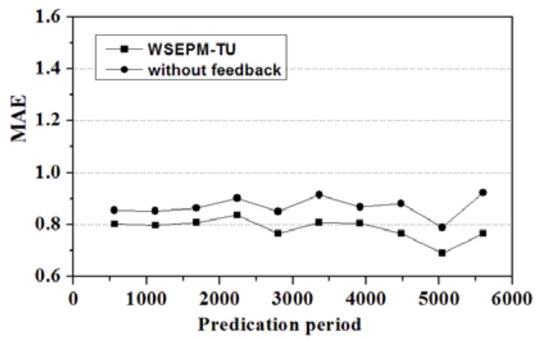

Figure 25. MAE analysis on D3 with various feedback control strategies

In Fig. 18, Fig. 21 and Fig. 24, the initial isolate points at the predication periods from 1 to 560 originate from the initial $|Y|$ ratings. In the experiments, we directly store the initial $|Y|$ ratings in the historical ratings table. If there are isolate points in these $|Y|$ ratings, the malicious ratings would be drawn in the first box plot. In fact, the malicious ratings only affect the following $|Y|$ predication periods. From the whole predication periods, the malicious ratings in the first box-plot have less impact on the predication performance of WSEPM-TU.

According to the experimental results, WSEPM-TU with the feedback control strategy would filter out the malicious ratings and out-perform compared with the model without the feedback control strategy.

VII. Conclusion

This paper proposes a web services evaluation predication model based on time utility. The model uses the complex quantification method reflecting the distinction quantification results for different web services. Then, the quantification results are used to optimize the length of predication windows. Also, the feedback control strategy is involved in WSEPM-TU to filter out the malicious ratings. According to the experimental results, WSEPM-TU with the feedback control strategy would filter out the malicious ratings and out-perform compared with other predication model.

WSEPM-TU adopts the memory enhancement phaenomenon to obtain the distinct quantification results. In the filed of psychology, there are more controversial model that can be used to gain the distinct quantification results. In the future, we would compare these models with complex quantification model in practical ratings datasets, and achieve the more suitable quantification model for web service predication process.

Acknowledgment

This work is sponsored by the National Natural Science Foundation of China under Grant No. 60973075, the Provincial Natural Science Foundation under Grant No.F200937 and No. F201110, the Foundation of Harbin Science and Technology Bureau under Grant No. RC2009XK010003.

References

[1] WU Minghui, XIONG Xianghui, YING Jing,et al, "QoS-driven global optimization approach for large-scale web services composition," *Journal of Computers*. vol. 6, no. 7, pp. 1452-1460, July, 2011. "doi: 10.4304/jcp.6.7.1452-1460"

[2] GUO Guangjun, "A Method for Semantic Web Service Selection Based on QoS Ontology," *Journal of Computers*. vol. 6, no.2, pp. 337-386, February, 2011. "doi: 10.4304/jcp.6.2.377-386"

[3] SHAO Ling-Shuang, ZHOU Li, ZHAO Jun-Feng, XIE Bing, MEI Hong, "An extensible management framework for web service qos," *Journal of Software*. vol. 20, no. 8, pp. 2062-2073, April 2009. (in chinese)

[4] M. Serhani, A. Benharref, "Enforcing Quality of Service within Web Services Communities," *Journal of Software*.

vol 6, no. 4, pp. 554-563, April, 2011. "DOI: 10.4304/jsw.6.4.554-563"

[5] E. Badidi, L. Esmahi, "A Scalable Framework for Policy - based QoS Management in SOA Environments, " *Journal of Software*. vol. 6, no. 4, pp. 544-563, April, 2011. "DOI: 10.4304/jsw.6.4.544-553"

[6] YANG Wenjun, LI Juan-Zi, WANG Ke-Hong, "Domain-adaptive service evaluation model," *Chinese Journal of Computers*. vol. 28, no. 4, pp. 514-523, August, 2005. (in chinese)

[7] QIAO Baiyou, DING Linlin, WEI Yong, WANG Xiaoyang, "A KNN query processing algorithm over high-dimensional data objects in P2P systems," *in Proc. of the 2011 2nd International Congress on Computer Applications and Computational Science, Heidelberg, Berlin, Springer Press,* vol. 144, pp. 133-139, 2012. "DOI: 10.1007/978-3-642-28314-7_19 "

[8] Wikipedia. "*k*-nearest neighbor algorithm," 2012-3-18; Available from: *http://en.wikipedia.org/wiki/Knearest_neighbor_algorithm.*

[9] YE Tao, ZHU Xue-Feng, LI Xiang-Yang, SHI Bu-Hai. "Soft sensor modeling based on a modified k-nearest neighbor regression algorithm," *ACTA AUTOMATICA SINICA*. vol. 33, no. 9, pp. 996-999, September 2007. (in chinese)

[10] LI Xiaoyong, GUI Xiaolin, "Cognitive model of dynamic trust forecasting," *Journal of Software*. vol. 21, no. 1, pp. 163-176, January 2008. (in chinese)

[11] I. Koychev, I. Schwab, "Adaptation to Drifting User’s Interests," i*n Proc. of ECML 2000 Workshop: Machine Learning in New Information Age*, Barcelona, Spain, IEEE Press, pp. 39-46, 2000.

[12] J. Krücker, XU Sheng, V. Aradhana, K. Locklin, A. Hayet , G. Neil, et al, "Clinical utility of real-time fusion guidance for biopsy and ablation," *Journal of Vascular and Interventional Radiology*, vol. 22, no. 4, pp. 515-524, April 2011. "doi: 10.1016/j.jvir.2010.10.033"

[13] M. Kukar, S. Ljubljana, "Drifting concepts as hidden factors in clinical studies," *in Proc. of Artificial intelligence in Medicine 9th Conference on Artificial Intelligence in Medicine in Europe, AIME 2003,* Protaras, Cyprus, IEEE Press, pp. 355-364, October 2003.

[14] SUN Chao, JIANG Bo, "The research on model of network teaching system based on agent and recommendation technology," *Journal of Zhengzhou University*, vol. 40. no. 3, pp. 84-87, September 2009. (in chinese)

[15] YIN Guisheng, WANG Shuyin, "Dynamic trust model of internet-ware based on task classification," *in Proc. of 2nd International Conference on Mechanical Engineering and Green Manufacturing, MEGM 2012*, Chongqing, China, IEEE Press, March 2012. "doi: 10.4028/www.scientific.net/AMM.155-156.221"

[16] I. Zliobaite, "Learning under Concept Drift: an Overview," *Technical Report. 2010, Faculty of Mathematics and Informatics*, Vilnius University: Vilnius, Lithuania, 2010.

[17] H. Ebbinghaus, "Memory: A Contribution to Experimental Psychology," *New York by Teachers College, Columbia U niversity*, 1885. http://psy.ed.asu.edu/~classics/Ebbinghaus/index.htm.

[18] Apple. 2011-05-12. Available from http://www.apple.com.cn/mac/app-store.

[19] ZHANG Jinbo, LIN Zhi-qing, XIAO Bo, ZHANG Chuang, "An optimized item-based collaborative filtering recommendation algorithm," in Proc. Of *Network Infrastructure and Digital Content, IC-NIDC 2009*, Beijing, China, IEEE Press, pp.414-418, November 2009. "doi: 10.1109/ICNIDC.2009.5360986"

[20] J. Haddad, M. Maude1, R. Guillermo, R. Marta, "QoS-driven selection of web services for transactional composition," *in Proc. Of the IEEE International Conference on Web Services, ICWS 2008*, Beijing, China, IEEE Press, pp. 653-660, September 2008. "doi: 10.1109/ICWS.2008.116"

[21] KIM, Youngae, PHALAK Rasik, "A trust prediction framework in rating-based experience sharing social networks without a web of trust," *Information Science*, vol. 191, no. 15, pp. 128-145, January 2012. "doi: 10.1016/j.ins.2011.12.021"

[22] ZHAO, Laiping, REN Yizhi, LI Mingchu, S. Kouichi, "Flexible service selection with user-specific QoS support in service-oriented architecture," *Journal of Network and Computer Applications*, vol. 35, no. 3, pp. 962-973, May 2012. "doi: 10.1016/j.jnca.2011.03.013"

Guisheng Yin was born in China. He received the PhD degree in automatic control from Harbin Engineering University, where he is a Full Professor and Doctoral Advisor, the Acting Dean of College of Computer Science and Technology and the Dean of School of Software Engineering. He ever worked in Tokyo University before he joined the current university. His research interests include database and web services discovery.

Xiaohui Cui was born in China. He got his bachelor’s degree in Harbin Engineering University in 2007. Now he is a PhD Candidate of computer application in College of Computer Science and Technology of Harbin Engineering University. His research interests include web services discovery and web services evaluation.

Yuxin Dong was born in China. She received the PhD degree in Computer Application from Harbin Engineering University, where she is an associate professor. Her research interests include Internetware and trust evaluation.

Jianguo Zhang was born in China. His research interests include trust evaluation.