Evaluating Reputation of Web Services under Rating Scarcity

Xin Zhou, Donghui Lin, Toru Ishida
Department of Social Informatics, Kyoto University
Yoshida-Honmachi,Sakyo-Ku, Kyoto, 606-8501, Japan
xin@ai.soc.i.kyoto-u.ac.jp, {lindh, ishida}@i.kyoto-u.ac.jp

***Abstract*—With the proliferation of Web services, more and more functionally equivalent services are being published by service providers on the Web. Although more services mean more flexibility for consumers, it also increases the risk of choosing as consumers may have little or no past experience with the service they will interact with. Therefore, reputation systems have been proposed and are playing a crucial role in the service-oriented environment. Current reputation systems are mainly built upon the explicit feedback or rating given by consumers after experiencing the service. Unfortunately, services at the cold-start stage, prior to being rated, face the rating scarcity problem. In this paper, we focus on this problem and address it through a novel reputation model that uses the Elo algorithm to consider consumer implicit information in a graph analysis approach. Theoretical analysis is conducted to identify the sufficient and necessary condition for the model to converge to a stable state. Furthermore, experiments confirm our model outperforms the widely adopted reputation algorithm in both accuracy and convergence in the situation of rating scarcity.**

***Keywords*-reputation model; Web services; rating scarcity; implicit information**

I. Introduction

The service-oriented computing paradigm and its realization provide a promising approach to integrate computational resources seamlessly and dynamically across organizational boundaries [1]. In the service-oriented computing environment, such as Amazon Web Services [1] and Language Grid [2], two parties are involved: services offered by service providers and service consumers. Service consumers search and review the description of the services offered by service providers and select a service. With more and more web services being deployed on the Web, service consumers have more alternatives to select. As consumers may have little or no past experience with the service they will interact with, the risk of decision making also increases. Reputation systems were proposed to mitigate the risk that consumers faced when selecting a new service [3]. Existing service reputation systems, mainly based on the ratings given by service consumers, are one of the most important guides that the consumer has in making a decision, as they reveal how other consumers evaluated the services true ability in real scenarios. However, the ratings may be very sparse or unreliable for the following reasons:

1) Ratings are *skewed* towards high values [4]. Consumers cannot express their opinion truthfully if only numerical ratings are used [5]. Moreover, they care about the impact of their feedback on the services future benefits in the marketplace, and so tend to offer a relatively high value unless extremely unsatisfied.
2) Not all customers rate the transaction [6]. As a result, transaction volume is much larger than the number of ratings received. Normal customers, those who pay for the service, have little interest in entering their ratings unless they are extremely satisfied or unsatisfied.
3) No rating is available at the cold-start stage [7]. Upon the introduction of a service, no consumer has interacted with the service, so no historical evidence can be used to derive a reputation score for the service..

The rating scarcity problem is rarely addressed in the literature of the service domain. Some researchers briefly mentioned that it is a weakness common to rating aggregation systems [8]. They argue that the accuracy and stability of the system may be compromised by rating scarcity [9], [10]. To overcome the limitations of existing reputation systems in the service-oriented computing domain, we present *imRep* (implicit Reputation model), a new reputation model that integrates the consumers implicit judgments at the service evaluation moment with the explicit ratings given at the moment of transaction completion. The advantages of implicit judgements offer two benefits. First, implicit judgments are more broadly available since the number of alternative services is usually one or two orders of magnitudes higher than the number of ratings. Second, implicit judgments can more truthfully express the consumers preference for the service as the implicit actions of the consumers are not revealed and, consequently, the consumers do not bias their judgements towards high ratings. As a result, the obtained information is not skewed.

To describe our model, we consider the consumer decision of selecting service A, thus ranking A above some other alternative B, as the input of A defeating B in a match. Consumer decisions can thus be interpreted as a set of match outcomes. There are many algorithms [11], [12], [13] that can be used to aggregate match outcomes. Our reputation system builds upon the Elo ratings system [11],which is widely used to evaluate chess players. In particular, we

[1]https://aws.amazon.com/

978-1-5090-2628-9/16 $31.00 © 2016 IEEE
DOI 10.1109/SCC.2016.35

assign each service an initial rating and we treat each service in the context of consumer judgement as a participant in a chess tournament. Services that are selected will get their scores increased and those that are ignored get their scores decreased. The extent of the increase or the decrease depends upon the scores of the other services, i.e., the better the quality of the ignored service is, the more the scores of the selected services are increased. Similarly, the worse the quality of the selected service is, the more the scores of the ignored services are decreased.

Our major contributions are summarized as follows:

- We adopt implicit consumer judgments of services to address the problem of rating scarcity. Based on the implicit information, our model can assign reputation values to unrated services and boost up the convergence rate towards service ranking stability.
- Both theoretical analysis and extensive experiments in two sets of environments are conducted to demonstrate that our model can not only provide reliable reputation values for unrated services, but also converge to an accurate ranking state.

The rest of our paper is organized as follows. We discuss related work in Section II and propose our reputation model in Section III. We analyze the convergence conditions of our reputation model in reaching reputation ranking stability in Section IV. The validity and performance of the model are evaluated in Section V. The research is rounded off with a conclusion in Section VI.

II. Related Works

Serval studies have proposed reputation models and analyzed the rating scarcity problem in the service-oriented environment [9], [10]. Rating scarcity or sparsity mainly occurs at the cold-start stage of a service or when a service experiences a long period of inactivity.

The Bayesian reputation system proposed by Jøsang and Quattrociocchi [10] addresses the importance of base rate in the cold-start stage. The system assigns an initial value to new services according to the reputation distribution of the community. Although the approach can assign meaningful initial values to services, we argue that its performance is deficient in two aspects: First, the assigned initial value is unfair to new services. Even though it is clear that the base rate can be biased either negatively or positively, we lack the evidence needed to correct the base rate distribution. Otherwise, assigning an arbitrary initial value may unfair to some services. Second, the Bayesian reputation system cannot boost the convergence rate. The convergence rate is not changed by assigning an initial value, the reputation of the service is unstable until enough ratings have been aggregated.

A prediction model based on historical data is proposed in [9]. In the situation of rating scarcity, reputation is predicted by a Hide Markov model. Another recommendation system derives the reputation of a service by injecting pseudo users into the system [14]. The pseudo users rate the service according to attributes of items or users. Unfortunately, it is difficult to establish a valid relationship between attributes and service reputation.

Our work is inspired by WorkerRank [15], which was proposed to rank workers in crowdsourcing marketplaces. In WorkerRank, worker ranking is built upon the implicit judgement exhibited by employers. Experiments on real data showed that the model is more accurate and a better indicator of worker quality towards hiring decisions. Our model adopts the underlying ranking mechanism, the Elo method [11]. Elo uses a Bayesian update scheme to score chess game players based on past match activity and update their scores by their expected performance in future tournaments. We integrate the implicit reputation score output by the Elo approach with the rating given by consumers to create a more reliable reputation model.

III. The Reputation Model

In this section we describe a reputation model that builds on the service choices of consumers.

A. Notation

We represent the service-oriented computing environment as directed bipartite graph $G = (S, L, A)$; S is the set of services in the environment; L is the set of selections made by consumers. Edge $(s, l) \in A$ represents a consumer actions on the services $s \in S$ based on one selection $l \in L$. We consider the following three consumer actions:

- *interact:* the consumer selects the service for interaction;
- *review:* the consumer reviews the service for detailed information;
- *ignore:* the consumer reviews the service's brief description but takes no action it.

Among these three actions, we consider the first two as positive indications of service performance, and the last one as negative. We also assume that the consumer actions indicate a ranking on the Quality of Service (QoS) in the following decreasing order: interact $>$ review $>$ ignore. For example, a service that is selected for interaction is considered better for the consumer than a service that is ignored. The objective of this paper is to compute a score, $r(s)$, for each service, s, that is informative of its QoS. Score $r(s)$ is considered informative if the relative difference between $r(s)$, $r(s')$ for services s and s' is predictive for the relative ranking of s, s' in subsequent matches.

B. Ranking Web Services with Tournament-based Algorithm

The key idea of our model is to use consumer judgments as implicit information to compute the implicit reputation score $r(s)$ for service s. That is, alternative services $S_l \subset S$ at selection l of the consumer are taken as having competed

in a tournament, and service performance at each selection is examined in pairwise manner. Certainly, services with better actions win over services with weaker actions (for example, interact wins over review, review wins over ignore). Note that draws of services with identical better actions provide useful information about their relative qualities. The same does not necessarily hold for the case of draws of services with negative actions (such as two services that are ignored). Our reasoning is that using judgments of the very brief descriptions would increase the uncertainty in making decisions, and introduce judgment noise to the final result. Hence our algorithm exclude draws among negative action services. The scores are computed via a reputation calculation process on graph $G_l \subset G$ generated by each selection l, using the Elo constants for t_{elo}, K, as shown in Algorithm 1.

Algorithm 1 Update implicit reputation scores

1: **procedure** IMRANK
2: Inputs: Graph $G_l = (S_l, l, A_l)$.
3: Output: Implicit reputation score for $s \in S$.
4: **for** $s \in S_l$ **do**
5: $T_{s,l} = 0, X_{s,l} = 0$
6: **for** $s, s' : (s, l) \in A_l, (s', l) \in A_l, s \neq s'$ **do**
7: $T_{s,l} += t(s, s', l)$
8: $T_{s',l} += t(s', s, l)$
9: $X_{s,l} += t_{elo}(s, s', l)$
10: $X_{s',l} += t_{elo}(s', s, l)$
▷ Update Competition scores:
11: $\tau(s, l) = T_{s,l} - X_{s,l}$
12: $\tau(s', l) = T_{s',l} - X_{s',l}$
▷ Update implicit reputation scores:
13: $r^i(s) = r^{i-1}(s) + \frac{K}{n-1} \cdot \tau(s, l)$
14: $r^i(s') = r^{i-1}(s') + \frac{K}{n-1} \cdot \tau(s', l)$

In the algorithm, we first initialize reputation score $r(s)$ for services s to 1.0. Then, when a consumer makes decision in selecting a service in process l, we update the service reputation score by considering every pair (s, s') of services available to the consumer c for judgment as a game in a tournament with possible outcome of matches:

$$t(s, s', l) = \begin{cases} 0, & \text{if } s \text{ lost against } s' \text{ at } l; \\ 0.5, & \text{if } s \text{ came to draw with } s' \text{ at } l; \\ 1, & \text{if } s \text{ won against } s' \text{ at } l. \end{cases} \quad (1)$$

After general initialization, for each selection made by consumers, we set the outcome variables $T_{s,l}, X_{s,l}$ to 0 for each alternative service s at selection process l made by consumer c. We compute $T_{s,l}$ as the sum of the actual points that s scored in selection process l against the other service candidates. Also, we compute the sum of expected points $X_{s,l}$ that s would earn against other service candidate $s' \neq s$ in the selection of process l, according to Elo's formula [11]. As each service competes with other services (lines 6-10 in Algorithm 1), the accumulated expected points and the actual points earned in selection l can be derived.

Finally, we update the reputation score of service s in an iterative way in line 13. $r^{i-1}(s)$ is the current reputation score while $r^i(s)$ is the updated reputation scored based on current value. K-factor represents the maximum possible adjustment per game (set here to 32), and is normalized by $n-1$, where n is the number of the services in one selection process. Without normalization, we would have dramatic inflation/deflation of scores with each selection process. The reputation score of a service is updated according to the average competition results between other services in each selection process. This averaging ensures that services gain according to their relative position in the service candidate ranking, rather than the number of alternative services.

C. The Proposed Model

Our Algorithm 1 yields the implicit ranking score of the services, but explicit ratings will be available for some services. We expect that a hybrid reputation model that combines both types of information would yield better results. This subsection introduces the *imRep* model; it integrates the ranking yielded by implicit reputation scores and rating-based ranking into a service list to predict the true ranking with higher accuracy. Along with the ranking, the reliability of the reputation score for each service is allocated under the rating criteria.

When mapping the implicit reputation scores into rating scores, we consider both the implicit reputation ranking and the ranking from feedback. For all ratings of service s, we define two parameters to evaluate the ratings: mean of the ratings, μ_s, and the number of ratings, n_s. The higher of n_s is, the more reliable is the average rating, μ_s, of service s. Thus, for s, the final reputation score will approach μ. Assuming that the rating value is lies in the range [0.0, 1.0], the $imRep$ method is illustrated in Algorithm 2.

Algorithm 2 Generate normalized reputation value

1: **procedure** IMREP
2: Inputs: S; implicit reputation score ri_s for $s \in S$.
3: Average rating μ_s for $s \in S$.
4: Number of ratings n_s for $s \in S$.
5: Output: Normalized reputation value for $s \in S$.
6: Let services with $max(ri_{s_i}), min(ri_{s_i})$ as s_x, s_y.
7: $ri_{max} = (ri_{s_x} < 1.0 \ ? \ 1.0 : ri_{s_x})$
8: $ri_{min} = (ri_{s_y} > 1.0 \ ? \ 1.0 : ri_{s_y})$
9: $\mu_{max} = (\mu_{s_x} \neq 0 \ ? \ \mu_{s_x} : 1.0)$
10: $n_{max} = (n_{s_x} \neq 0 \ ? \ n_{s_x} : \infty)$
11: **for** $s_i \in S$ **do**
12: $v_i = \mu_{max} \cdot \frac{ri_{s_i} - ri_{min}}{ri_{max} - ri_{min}}$
13: $\alpha = 1/e^{n_{max}/(n_{s_i}+1)}$
14: $r(s_i) = v_i \cdot (1 - \alpha) + \mu_{s_i} \cdot \alpha$

Initially, we iterate over the implicit reputation score of each service and record the services with maximized score as s_x, and with minimized score as s_y. The first ranked service is taken as a fiducial reputation value. Other service reputations are calculated according to it. To avoid meaningless values, lines 7 - 10 of Algorithm 2 check and adjust the maximum or minimum value. Finally, for each service, the final reputation value determined from two parts. The first part is a normalized value from the implicit reputation score, and the second part is the average rating value. To balance these two parts, we use the relevant value between the number of ratings. As a service accumulates more ratings, more trust is laid on its averaged rating value.

Reputations should converge quickly, and be stable [16]. In the next section, we will analyze the convergence conditions and the convergence speed of our model.

IV. Convergence Analysis

In this section, we discuss the situations in which the competition scores can converge and the convergence rate. First, we define the convergence of the proposed reputation model here as the result that ranking is asymptotical to the ranking of QoS for all services with increasing competition. Suppose that in the service-oriented computing environment, the number of services in S is n, $S = \{s_1, s_2, ..., s_n\}$, and P is a family of subsets. Each item in P is a set of services ranked based on consumer selection. The extreme limit is when consumers select from the same set of services every time; no new subset is created in P.

Theorem 1: We define the relation $\leq$ on set S as less than, by Algorithm 1, $(S, \leq)$ is a linearly ordered set. Set S can converge to the correctly ranked set if all the following conditions hold:

1) P does not contain empty set.
2) Any subset X of S, also exists in P.
3) The judgement number for each subset A of P is large enough.

In mathematical notation, these conditions can be summarized as:

1) $|P| = 2^n - 1$ && $\emptyset \notin P$
2) $\forall A \in P, |A| \to \infty$

Proof: It is easy to show that $\forall A \in P, (A, \leq)$ is a linearly ordered set. When consumer preference is not considered, the ranking result is convergent as any two elements in S are comparable: $\forall s_i, s_j \in S$, as $|P| = 2^n - 1$, either $s_i \leq s_j$ or $s_j \leq s_i$. That is, $\forall s_i, s_j \in S, \exists A \in P$ satisfied $s_i \in A$ and $s_j \in A$.

When we consider consumer preference, $\forall A \in P$ of the judgement result for a particular consumer, the order of $(A, \leq)$ may disagree with the correct order, but as $|A| \to \infty$, the consumer preference is offset and $(A, \leq)$ settles on a general ranking for each element in B. This process is consistent with the definition of reputation. Thus, set S not only can converge to a ranked set, but also can converge asymptotically to the correct order given the above conditions. ■

The above analysis addresses the sufficient conditions for convergence, we discuss the necessary conditions for convergence as follows:

1) In the ideal condition, there are at least $n1$ matches. In the ranked list, the front service s is exactly compared with the service that follows s. But, within $n - 1$ competitions, service ranking is highly dependent on the preference of the consumer and inaccurate.
2) In our model, at least $n(n-1)/2$ matches are needed. Each service is compared with all other services. That is, the total number of matches is C_n^2. Ranking is not assured of converging if there are fewer matches than C_n^2.
3) Algorithm 1 will not converge if a certain percentage of consumer judgements is irrational. We will discuss this condition below.

The above analysis finds that the minimum acceptable condition for the convergence of Algorithm 1 is that the number of matches between services must not be less than $n(n-1)/2$.

The factors that control the convergence rate are:

- Service number n. More services mean more matches are needed to rank the services.
- Dependability of consumer judgement. If the judgment of most consumers is rational, then Algorithm 1 can converge because random judgments are offset and the rational judgments will form the ranking. If, however, consumers try to game the rating system consistently, the algorithm will fail to converge. In our research, the consumers are supposed to be dependable.
- The diversity, d, of consumer preference. Popular service usually has a outstanding performance on every QoS metric, an environment with various types of consumers will make the system converge quickly.
- Competition number m. The number of competitions between services contribute proportionally to the convergence rate.

In our paper, we assume that all the consumers are dependable one and so the competition result reflects the QoS of the service and preference. With this assumption, convergence rate λ can be written as:

$$\lambda \propto \frac{m \cdot d}{n} \tag{2}$$

Theoretically, the convergence rate is inversely proportional with the number of services. In the following section, we test the convergence and the accuracy of the proposed model with experimental evaluations using data of simulated and actual web services.

Table I
QOS METRIC FOR VARIOUS AVAILABLE MAIL VERIFICATION WEB SERVICES.

ID	Service Provider & Name	RT	TP	AV	AC	IA	C
1	StrikeIron Email Verification	710	12.00	98	96	100	1
2	ServiceObjects DOTS Email Validation	391	9.00	99	99	90	5
3	StrikeIron Email Address Verification	912	10.00	96	94	100	7
4	CDYNE Email Verifier	910	11.00	90	91	70	2
5	XMLLogic ValidateEmail	720	6.00	85	87	80	1.2
6	Webservicex ValidateEmail	1232	4.00	87	83	90	0
7	XWebservices XWebEmail-Validation	1110	1.74	81	79	100	1

V. EVALUATION

To evaluate the proposed reputation model, we compare $imRep$ with both the real reputation value calculated by the WsRF algorithm [17] and the explicit average reputation model. The average algorithm takes the mean of all explicit ratings as the reputation value and is widely used in commercial services like Amazon [3].

A. Environment setting

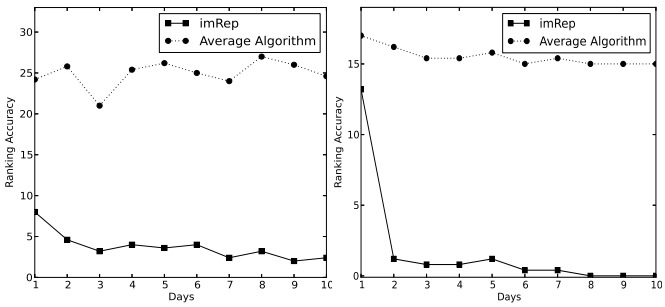
First, we consider a simulated environment with 100 service consumers and 50 services, the QoS parameters of the service are generated randomly. Second, seven real services listed in Table I are used to evaluated the performance of the proposed model [17]. In the following subsection, we try to evaluate the performance of *imRep* on services in the real world. We selected QoS parameters following earlier research [17]:

1) Response Time (RT): the time taken to send a service request and receive a response (unit: milliseconds)
2) Throughput (TP): the maximum number of requests that can be handled per given unit of time (unit: requests/min)
3) Availability (AV): a ratio of the time period which a Web service is available (unit: %/3-day period).
4) Accessibility (AC): the probability that a system is operating normally and can process requests without any delay. (unit: %/3-day period).
5) Interoperability Analysis (IA): a measure indicating whether a Web service is in compliance with a given set of standards. (unit: % of errors and warnings reported).
6) Cost of Service (C): the cost per Web service request or invocation (cents per service request).

To facilitate service selection of a consumer, the preference of the consumers is simulated by weighting the above QoS parameter. The weight is uniformly selected from the range of [0, 1.0]. The simulation ran for 10 days, in each day, the probability of each service consumer searching for a service was 0.5. They first viewed the brief search results for further actions. They may review some services and select one for interaction. The services are reviewed and selected according to the user-centric QoS-based service discovery model [17]. For example, if the cost of service s_1 and s_2 is 0 cent and 5 cents respectively. And the response time of service s_1 and s_2 is 710ms and 391ms respectively. Suppose consumer c_1 sets all his weights to zero except cost, in this situation, he intend to minimize cost since it represents 100% significance to him. Under the QoS-based service discovery model, s_1 will be selected by consumer c_1. After the interaction, the consumer rates the performance of the service.

B. Evaluation of simulated Web services

Our proposed model is compared with WsRF and the average reputation algorithm. We assume that the rating criteria is [0, 1.0], and the rating offered by rational consumers on a service s after interaction follows $N(WsRF(s), \sigma)$ with probability $Pr = 0.4$. where WsRF(s) is the reputation value of service s calculated by WsRF. The ratings follow a normal distribution, while the other ratings skew towards high values with probability of $1 - Pr$.



(a) (b)

Figure 1. Evaluation results of ranking accuracy of imRep. a) Compared with the average algorithm on 10 simulated web services. b) Compared with the average algorithm on 7 real web services.

Scenario 1: Ranking accuracy test on 10 simulated web services. The accuracy of the reputation model is measured as the ranking accuracy because the real reputation value is unavailable. The ranking accuracy of the model is evaluated as the sum of the deviation between current ranking and the correct ranking for each service; lower values are better. The ranking accuracy for *imRep* and the average algorithm is updated every day to show the detailed behavior of the model. Given that more data is accumulated each day, the ranking should converge to the correct ranking. The ranking results of *imRep* and the average algorithm at the 10th day are shown in Table II. In the table, we denote the ranking

Table II
RANKING COMPARISON FOR *imRep* AND THE AVERAGE ALGORITHM.

Web Services	WsRF		imRep			Average Algorithm		
	Value	Rank	Value	Rank	∇ Rank	Value	Rank	∇ Rank
s_1	0.82	1	0.98	1	0	0.98	2	-1
s_2	0.66	2	0.55	2	0	0.99	1	1
s_3	0.55	3	0.42	3	0	0.0	Ω	Ω
s_4	0.53	4	0.35	5	-1	0.0	Ω	Ω
s_5	0.50	5	0.37	4	1	0.0	Ω	Ω
s_6	0.48	6	0.24	6	0	0.0	Ω	Ω
s_7	0.46	7	0.15	8	-1	0.0	Ω	Ω
s_8	0.45	8	0.18	7	1	0.0	Ω	Ω
s_9	0.43	9	0.0	10	-1	0.0	Ω	Ω
s_{10}	0.40	10	0.01	9	1	0.0	Ω	Ω

accuracy of a service without scores as Ω. The table shows that most services cannot be ranked because lack of data. Hence, it is hard for consumers to decide between those services. This situation happens because at the cold-start stage of a service, if the number of potential consumers is not large enough, most consumers will tend to select the service that has a reputation value. As a result, it is hard for newly deployed services to be accepted, and thus rated, by consumers. Although the $imRep$ model wrongly ranked some services, the distance between the correct ranking is relatively small. The small QoS performance difference between those service pairs, such as s_7 and s_8, make it hard to distinguish between the two services. Given rating scarcity, with the limited amount of data available, this result is acceptable to consumers. Note the ranking accuracy for $imRep$ is $\sum_{s \in S}(|\nabla Rank|_s) = 6$ on the 10th day.

Besides considering the ranking on the final day, we calculate the ranking accuracy over time. To mitigate randomness, the simulation was run for multiple times, and the average results are plotted in Figure 1-(a). On day 10, some services may not receive any implicit or explicit scores, making it impossible to compute $\nabla Rank$ for those services. We denote the ranking accuracy of each such service as Ω to simplify the comparison and we set $\Omega = 3$ to plot the figure. The actual data is logged in column scenario 1 of Table III. The ranking accuracy of $imRep$ tends to converge on the zero point. However, on some days, the decisions of some consumers may affect the convergence rate, which we discussed before.

In the previous section, we analyzed the impact of service number n on the convergence rate. We used different service numbers to test the average ranking accuracy of the proposed model against the number of evaluation days. The result is plotted in Figure 2. In the figure, the convergence rate is inversely proportional to n approximately, which validates the equation (2). The average ranking accuracy falls as the number of days increases, however, it seems that the model cannot converge to the correct ranking. The key factor in this situation is the diversity d of consumer preference in equation (2). As consumers did not change their preference, they will always choose the same set of services. This breaks the sufficient condition $|P|$ discussed in Section IV and so service ranking is not assured of converging on the correct ranking.

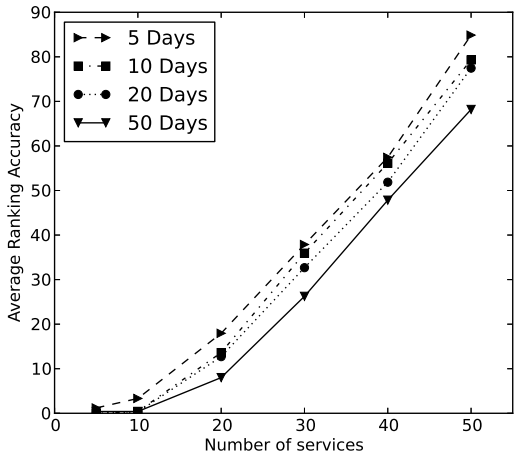


Figure 2. Convergence test of $imRep$ on the simulated services.

C. Evaluation on real services

With the popularity of Web services, it is easy to access various services. We adopt the dataset used in paper [17], all services are intended to validate e-mail. Details of the dataset are listed in Table I. Unlike the original service order in [17], we reorder the services according to their WsRF values here for ease of comparison. As the QoS parameters have different units, we use min-max normalization to unify the value of each QoS parameter into the range [0, 1.0], as is widely used [18], [19], [20]. Based on our dataset, we use the following equation to normalize the QoS values.

For positive parameters (*TP , AV, AC and IA*):

$$q'_{s_i} = \frac{q_{s_i} - q^{min}}{q^{max} - q^{min}} \tag{3}$$

For negative parameters (*RT and C*):

$$q'_{s_i} = \frac{q^{max} - q_{s_i}}{q^{max} - q^{min}} \tag{4}$$

Table III
RANKING ACCURACY FOR *imRep* AND THE AVERAGE ALGORITHM ON DIFFERENT TEST SCENARIOS. 1) SCENARIO 1: TEN SIMULATED WEB SERVICES. 2) SCENARIO 2: SEVEN REAL WEB SERVICES. 3) SCENARIO 3: SEVEN REAL WEB SERVICES AND ONE SERVICE UPDATES ITS PERFORMANCE ON DAY SIX. ($\Omega = 3$ *when plotting figure 1 and 2*).

Days	Test Scenario 1		Test Scenario 2		Test Scenario 3	
	imRep	Average Algorithm	imRep	Average Algorithm	imRep	Average Algorithm
1	3.8+1.4Ω	3.8 + 6.8Ω	1.2 + 0.4Ω	2 + 5Ω	2	5Ω
2	3.4 + 0.4Ω	6 + 6.6Ω	1.2	1.2 + 5Ω	2	5Ω
3	3.2	7.8 + 4.4Ω	0.8	0.4 + 5Ω	0	5Ω
4	4.0	6.2 + 6.6Ω	0.8	0.4 + 5Ω	0	5Ω
5	3.6	8.8 + 5.8Ω	1.2	0.8 + 5Ω	0	5Ω
6	4.0	7.6 + 5.8Ω	0.4	5Ω	12	4Ω
7	2.4	7.2 + 5.6Ω	0.4	0.4 + 5Ω	0	4Ω
8	3.2	10.2 + 5.6Ω	0	5Ω	0	12 + 4Ω
8	2.0	9.2 + 5.6Ω	0	5Ω	0	4Ω
10	2.4	7.8 + 5.6Ω	0	5Ω	0	4Ω

where q_{min} and q_{max} are the minimum and maximum values, respectively, for one QoS requirement. q'_{s_i} is the normalized value for service s_i.

Scenario 2: Ranking accuracy test on 7 real web services. As the dataset holds only service configuration, we need to simulate consumer preference. The consumer configuration is the same as in the total simulation environment. 100 consumers are active in the environment and they try to choose the service according to their preference and the QoS values of a service. We ran the experiments multiple times and plot the averaged results in Figure 1-(b); details are given in Table III(*Test scenario 2*). In Figure 1-(b), $imRep$ has faster convergence than in figure 1-(a). The differences between these two experiments are: 1) the number of services; 2) the performance gap between those services. It reasonable that a larger difference between services will make it easier for the algorithm to rank the services. However, the results in Figure 1-(a) and Figure 1-(b) also validate equation (2) to some extent.

Scenario 3: Convergence of reputation model in dynamic environment. Usually, the performance of a service dynamically changes with time. Is the reputation model stable enough to catch the changes and reflect the changes correspondingly? Although some changes are caused by consumers deliberately, such as collective attacks discussed in the literature [21], [22], [23], we assume the consumers are rational and legal.

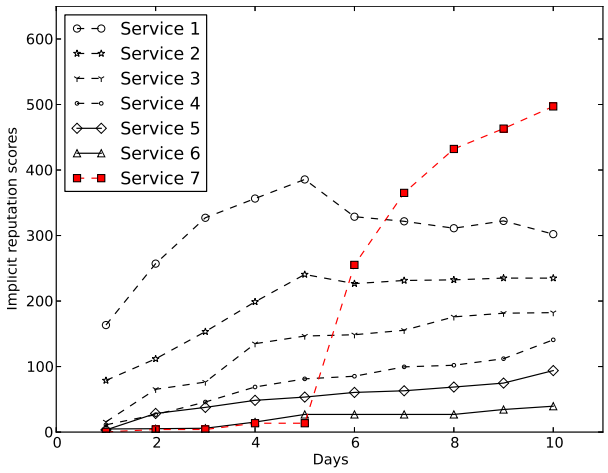


Figure 3. The daily changes of imRank scores changes when the 7-th service updates its performance.

A previous experiment showed that service 7 had the lowest reputation. The day-to-day results of scenario 2 in Table III show that $imRep$ basically converged on the correct ranking by the 5th day. In the robustness experiment, we update the QoS values for service 7 as (300, 10.00, 98, 95, 100, 1) at day 6. The updated QoS values for service 7 are not the best for every criterion, but have a competitive overall performance.

The experiment ran for ten days, to study the changes in ranking, and we used the implicit competition score output by Algorithm 1. The result is plotted in Figure 3. The changes in the scores of service 7 are shown by the solid line. When compared with scenario 2 in Table III, on the fifth day, the scores of the services indicate that the service ranking is stable. On the sixth day, when we updated the QoS values for service 7, the scores of service 7 in Figure 3 indicate that the implicit information based algorithm detected the changes. More consumers tended to consider service 7 because the performance of service 7 better matched the consumer preference. From the sixth day to the final day, the model reduced the scores of service 1 and increased the reputation of service 7. The distribution of the implicit scores converged quickly.

D. Analysis and discussion

In our experiments, we assumed that consumers with different preferences would select services with different QoS values. Jøsang et al. [3] defined reputation as what is generally said or believed about a services performance. Hence, reputation is an aggregate value from various consumers. By clustering the ratings by consumer preference, a consumer centered reputation system can be built and is

able to provide more accurate reputation values for a specific consumer cluster.

Although the proposed model is focused on the cold-start stage of a service, Algorithm 2 combines the implicit scores with the ratings given by consumers to generate the reputation value for services. As more and more ratings are accumulated, the number of ratings will lead the reputation value to the rating value, line 13. All reputation systems that are based on ratings can benefit from the implicit scores to reach a more accurate reputation result.

In Scenario 2, only 7 real Web services were used to test the reputation accuracy of the models. To obtain a comprehensive assessment, we simulate service numbers ranging from 5 to 50. However, we will test our model on large scale, real world Web services in future work.

VI. Conclusion

To overcome the rating scarcity problem, we proposed a reputation model based on the implicit behaviors of consumers. The proposed model considers the judgement actions from consumers on alternative services as a competition tournament among services, where service ranking is updated with each match. The convergence of the model was analyzed and experiments demonstrated the accuracy and convergence of the proposed model. This research provides ranking support for services without ratings at the cold-start stage and can boost the convergence rate towards the correct ranking. In future work, we plan to deploy our proposed model on the service-oriented computing platform Language Grid [2].

Acknowledgment

This research was partly supported by a Grant-in-Aid for Scientific Research (S) (24220002, 2012-2016) from Japan Society for Promotion of Science (JSPS).

References

[1] M. Alrifai and T. Risse, "Combining global optimization with local selection for efficient qos-aware service composition," in *Proceedings of the 18th international conference on World wide web*. ACM, 2009, pp. 881–890.

[2] T. Ishida, *The language grid: Service-oriented collective intelligence for language resource interoperability*. Springer Science & Business Media, 2011.

[3] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision." *Decision Support Systems*, vol. 43, no. 2, pp. 618 – 644, 2007.

[4] N. Hu, J. Zhang, and P. A. Pavlou, "Overcoming the j-shaped distribution of product reviews," *Commun. ACM*, vol. 52, no. 10, pp. 144–147, Oct. 2009.

[5] H. Ramn, R. Centeno, and M. Fasli., "From blurry numbers to clear preferences: A mechanism to extract reputation in social networks." *Expert Systems with Applications*, vol. 41, no. 5, pp. 2269–2285, 2014.

[6] L. Cabral and A. Horiacǫsu, "The dynamics of seller reputation: Evidence from ebay*," *The Journal of Industrial Economics*, vol. 58, no. 1, pp. 54–78, 2010.

[7] O. Arazy, N. Kumar, and B. Shapira, "Improving social recommender systems," *IT Professional*, vol. 11, no. 4, pp. 38–44, July 2009.

[8] M. Chen and J. P. Singh, "Computing and using reputations for internet ratings," in *Proceedings of the 3rd ACM conference on Electronic Commerce*. ACM, 2001, pp. 154–162.

[9] Z. Malik, I. Akbar, and A. Bouguettaya, "Web services reputation assessment using a hidden markov model," in *Service-Oriented Computing*. Springer Berlin Heidelberg, 2009, pp. 576–591.

[10] A. Jøsang and W. Quattrociocchi, *Advanced features in bayesian reputation systems*. Springer, 2009.

[11] A. E. Elo, *The rating of chessplayers, past and present*. Arco Pub., 1978.

[12] M. E. Glickman, "The glicko system," *Boston University*, 1995.

[13] R. Herbrich, T. Minka, and T. Graepel, "Trueskill(tm): A bayesian skill rating system," in *Advances in Neural Information Processing Systems 20*. MIT Press, January 2007, pp. 569–576.

[14] S.-T. Park, D. Pennock, O. Madani, N. Good, and D. DeCoste, "Naïve filterbots for robust cold-start recommendations," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 699–705.

[15] M. Daltayanni, L. de Alfaro, and P. Papadimitriou, "Workerrank: Using employer implicit judgements to infer worker reputation," in *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*. ACM, 2015, pp. 263–272.

[16] C. J. Hazard and M. P. Singh, "Macau: A basis for evaluating reputation systems." in *IJCAI*, 2013.

[17] E. Al-Masri and Q. H. Mahmoud, "Qos-based discovery and ranking of web services," in *Computer Communications and Networks, 2007. ICCCN 2007. Proceedings of 16th International Conference on*. IEEE, 2007, pp. 529–534.

[18] C. Shi, D. Lin, and T. Ishida, "User-centered qos computation for web service selection," in *Web Services (ICWS), 2012 IEEE 19th International Conference on*, June 2012, pp. 456–463.

[19] D. Lin, C. Shi, and T. Ishida, "Dynamic service selection based on context-aware qos," in *Services Computing (SCC), 2012 IEEE Ninth International Conference on*. IEEE, 2012, pp. 641–648.

[20] Y. Wang, Q. He, and Y. Yang, "Qos-aware service recommendation for multi-tenant saas on the cloud," in *Services Computing (SCC), 2015 IEEE International Conference on*. IEEE, 2015, pp. 178–185.

[21] X. Zhou and S. Matsubara, "Towards robust reputation system based on clustering approach," in *Services Computing (SCC), 2015 IEEE International Conference on*. IEEE, 2015, pp. 33–40.

[22] X. Zhou, T. Ishida, and Y. Murakami, "Dynamic sliding window model for service reputation," in *Services Computing (SCC), 2015 IEEE International Conference on*. IEEE, 2015, pp. 25–32.

[23] X. Wang, L. Liu, and J. Su, "Rlm: A general model for trust representation and aggregation," *Services Computing, IEEE Transactions on*, vol. 5, no. 1, pp. 131–143, 2012.