Selecting Trustworthy Service in Service-Oriented Environments

Chung-Wei Hang and Munindar P. Singh

Department of Computer Science
North Carolina State University
Raleigh, NC 27695-8206, USA
{chang,singh}@ncsu.edu

Abstract. Most of current service selection approaches in service-oriented environments fail to capture the dynamic relationships between services or assume the complete knowledge of service composition is known as a prior. In these cases, problems may arise when consumers are not aware of the underlying composition behind services. We propose a distributed trust-aware service selection model based on a Bayesian network for consumers to maintain their knowledge of the environment locally. Results show our model can punish and reward services in terms of QoS properties accurately with incomplete observations so that consumers can prevent themselves from interacting services with unsatisfying QoS.

1 Introduction

In service-oriented computing environments, computing resources are managed as *services*, which can be used directly or composed into larger services. Service-oriented architecture has been widely adopted in modern distributed environments, such as for cloud computing. We address the problem of selecting services based on criteria such as user requirements and service qualities.

The dynamism of quality of service (QoS) is the first challenge of service selection. For example, the number of requests to a shopping service is much higher in the holiday sale season than usual. Traditional approaches, for example, *Web Service Definition Language* or *WSDL*, describe the functionalities of services statically for users to match services to their needs. However, These approaches lack capabilities of capturing the non-functional aspect of services.

The research on *trust* modeling in artificial intelligence provides us with a promising solution to service selection. Trust is a basis of interactions, indicating the relationships between parties in large, open systems. Two parties must trust each other sufficiently to be willing to carry out desired interactions. In service-oriented context, a party Alice trusts another party Bob, because Alice expects Bob will provide desired service under the expected QoS. The trustworthiness of the parties can be defined by both functional and non-functional properties. Selecting desired services based on trust is called *trust aware service selection*.

Maximilien and Singh [1] develop a trust-aware approach to select services based on well-defined ontologies that provide a basis for describing consumers'

requirements and providers' advertisements. Their approach also captures the dynamism by taking QoS properties into account. However, the other aspect of dynamism comes from service composition. Unfortunately, Maximilien and Singh's approach fails to take service composition into consideration. Services may be composed into larger services. The underlying services of composed services may not be shown externally to the consumers. Service composition can be divided into many scenarios [2] and these scenarios can be nested. This makes QoS properties difficult to collect and evaluate. Consequently, our service selection is more complicated than selection without considering compositions because the consumers may not even know with whom they are interacting.

An ideal trust aware service selection should be able to (1) reward/punish underlying services in an appropriate way so that consumers and composed services will become reluctant to interact with low reputation services, and (2) suggest suitable composition.

This paper aims to provide a trust aware service selection model that can capture the dynamism from not only non-functional QoS properties but also service composition in service-oriented environments. We formalize a Bayesian service selection model, develop approaches for consumers to monitor and explore desired service composition. In this paper, we will show that how our approach rewards/punishes the services dynamically with incomplete knowledge of the composition. The suggestion of better service composition will be left as one of our future work.

2 Related Work

Milanovic and Malek [3] compare various modern web service composition approaches. They also conclude four necessary requirements for service composition: connectivity, nonfunctional QoS properties, correctness, and scalability. However, these approaches poorly define QoS properties. Our approach deals with QoS properties separately. No QoS properties are pre-defined.

Menascé [2] studies how QoS properties are aggregated in different service composition scenarios. However, this approach requires the knowledge of the composition. For example, service A invokes service B, which may invoke C and D with probability p_c and p_d. This information is not always available because of two reasons. First, the providers have no incentive to give such information. Second, modeling the invocation probabilities is not trivial. By contrast, our service composition model makes no assumptions. Our approach monitors and explores the desired services dynamically.

Wu *et al.* [4] use Bayesian networks to model a consumer's assessment of a service's QoS. Their approach provides consumers to combine different QoS attributes. Our model uses Bayesian networks to model service composition to evaluate the QoS properties of the composed services. Instead of combining these properties based on the trustworthiness of each QoS property, we may use *multiattribute utility theory* for decision making, which is beyond our scope.

Yue *et al.*'s approach is the closest work to ours. Yue *et al.* [5] propose a Bayesian network-based approach to model the causal relationships between elementary services. Their approach construct a web service Bayesian networks (WSBN) based on the invocations between the services. Then the service composition guidance can be made from the *Markov Blanket* of a given service. However, this approach fails to consider the dynamism because the guidance remains unchanged if the causal relationships are fixed. Our model captures the dynamism by updating the Bayesian network, which will eventually affect the trustworthiness of a service.

3 Service Selection Model

We propose a trust-aware service selection model based on a Bayesian network. We represent trust based on the *beta distribution*, which can be integrated with Wang and Singh's model [6, 7]. The trustworthiness of services should be estimated based on both direct and indirect experience. Direct experience is referred to the previous quality of service received from the target, whereas indirect experience comes from referrals by peers. To model trust from indirect experience, which can be found in [8], is beyond our scope.

Estimating trust from direct experience is not straightforward in a service composition setting, because some services may not expose details of their composition to their clients directly. A client may interact with a composed service without knowing other underlying services. In such a case, evaluating the trustworthiness of services is no longer an easy task. For example, a client books an itinerary from a composed travel agent service, which interacts with other underlying services like flight services, hotel services, and transportation services. Suppose the client is not satisfied with the composed service because of its late response time. The model should penalize the composed service, as well as the underlying ones. If the hotel service, for instance, is reported to be the cause of unsatisfying QoS, the model should reflect the changes in the way that clients or other composed services become reluctant to interact with it. Also, as the experience increases, the model should be able to suggest appropriate composition.

Our service selection method models causal relationships between services with a Bayesian network. Each consumer maintains its own local model to guide itself to reward or penalize services based on direct interactions. The trust information can be also aggregated with referrals from other consumers. Figure 1 shows the architecture of our trust-aware service selection model. Our model is two-fold. First, a consumer keeps interacting with the services, constructs and updates its local service composition model, and get composition suggestions from the model. Second, consumers may exchange referrals with each other. This indirect evidence can be aggregated with the trust information in our service selection model, helping consumers discover strangers and identify desired services more quickly. The integration of the indirect evidence is our future work.

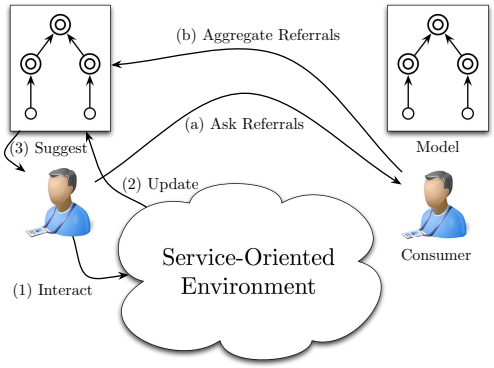

Fig. 1. Trustworthy service selection architecture

3.1 Bayesian Networks

The purpose of modeling service composition is to model how a certain QoS property of a component service can affect the whole composition. For example, the reliability of a composed travel service may be affected by the reliability of the underlying hotel service and flight service. If the underlying service is not reliable, the composed service is very likely not reliable either. Thus, the composition model should be able to not only represent the relationships between (composed) services, but also capture the causal factors between them. Of course, QoS properties of underlying services may not have influence on the composed services. For example, the reliability of the composed service may stay the same no matter how a particular underlying service performs. In other words, the trustworthiness regarding the reliability of the composed service should not correspond to the trustworthiness of that underlying service.

We introduce a Bayesian network-based service selection model, which can be constructed from the *incomplete observations* (direct experience) of a consumer. Here, we emphasize incomplete observations because not all QoS properties are observable from the consumers' point of view. An observation of a particular QoS property of a service d at time t can be represented as a number x_d^t between 0 and 1. Some QoS properties, say, error, can be simply considered as positive 1 or negative 0. Other quantitative QoS properties like up-time should be further projected to an real number from 0 to 1. An observation D^t of the whole composition at time t can be written as $D^t = (x_1^t, x_2^t, \ldots, x_d^t)$, where d is the number of services in the composition.

A Bayesian network is an acyclic directed graph $G = \langle V, R \rangle$ with random variables V as nodes, and edges R as the direct relationships between variables. A conditional probability associated to each node represents the probability of the node variable given its parent variable value. Let each node in the Bayesian network be the probability of getting good service (in terms of a particular QoS property) captured from a composed or elementary service. An edge represents the relationship of composition. For example, in Figure 2, a composed hotel

service H is composed of Four Season hotel service f, i.e., f is a child of H. Then the conditional probability of node H is the probability of getting good service in terms of a particular QoS property from H, given f provides good service. T, a travel service, is composed of composed hotel and car rental services H and C, which is also a composed service composed of Enterprise car service e.

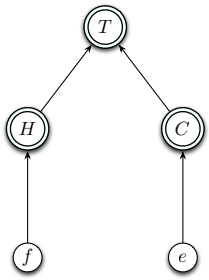

Fig. 2. Service composition example

The Bayesian network models the causal relationships between services. The conditional probability table associated with each node provides consumers a basis of how much responsibility an underlying service should take behind a service composition. The network can be constructed and the conditional probabilities can be learned from the consumers' direct experience.

3.2 Parameter (Trust) Estimation

Given an acyclic Bayesian network graph G over d variables, x_1, x_2, $\ldots x_d$, the associated joint distribution is written as

$$P(x_1, \ldots, x_d) = \prod_{i=1}^{d} P(x_i | x_{pa_i}) = \prod_{i=1}^{d} \theta_i \tag{1}$$

The conditional probability $P(x_i|x_{pa_i}) = \theta_i$ can be estimated by n observations, $D = \{(x_1^t, \ldots, x_d^t), t = 1, \ldots, n\}$, where x_{pa_i} is the set of parent variables of x_i. In fully observable environments, θ_i can be learned from the observed data by *maximum likelihood estimation* (MLE) [9].

In our model, each variable θ_i represents the probability of getting a good service from x_i given getting a good service from x_{pa_i}. The likelihood function can be then defined as [10],

$$P(D|\theta) = \prod_{t=1}^{n} P(x_1^t, \ldots, x_d^t | \theta) \tag{2}$$

$$= \prod_{t=1}^{n} \prod_{i=1}^{d} \theta_i \tag{3}$$

$$= \prod_{i=1}^{d} \prod_{x_i, x_{pa_i}} \theta_i^{n(x_i, x_{pa_i})} \tag{4}$$

$$= \prod_{i=1}^{d} \theta_i^{m_i} (1-\theta_i)^{l_i} \tag{5}$$

where $n(x_i, x_{pa_i})$ is the number of observations that satisfy the variable setting, and $m_i = n(x_i, x_{pa_i})$ and $l_i = n(x_{pa_i}) - m_i$. Then, the parameters that maximize the likelihood is $\hat{\theta}_i = \frac{m_i}{m_i + l_i}$.

3.3 Bayesian Inference

Note that, when the number of observations is small, MLE may yield over-fitted results. Consider an extreme case where $x_i^t = 1$ for $t = 1, \ldots, n$. The parameter $\hat{\theta}_i$ maximizing the likelihood is 1, which is not reasonable. Thus, we use *Bayesian inference* to treat this problem by introducing a beta distribution $P(\theta_i)$ over the parameter θ_i as a conjugacy prior.

$$P(\theta_i) = \frac{\Gamma(\alpha_i + \beta_i)}{\Gamma(\alpha_i)\Gamma(\beta_i)} \theta_i^{\alpha_i - 1} (1-\theta_i)^{\beta_i - 1} \tag{6}$$

where α_i and β_i are *hyperparameters* controlling the distribution of parameter θ_i. The expected value or mean of θ_i is given by $E(\theta_i) = \frac{\alpha_i}{\alpha_i + \beta_i}$. Bayesian inference uses observations to update the prior. The parameters θ_i can be learned using Bayes rule.

$$P(\theta_i | D) = \frac{P(D|\theta_i) P(\theta_i)}{P(D)} \tag{7}$$

That is, the posterior distribution $P(\theta_i|D)$ is propositional to the multiplication of the prior $P(\theta_i)$ and the likelihood function $P(D|\theta_i)$. Now we can put equations 5, 6, and 7 together and normalize it,

$$P(\theta_i | D) = \frac{\Gamma(m_i + \alpha_i + l_i + \beta_i)}{\Gamma(m_i + \alpha_i)\Gamma(l_i + \beta_i)} \theta_i^{m + \alpha_i - 1} (1-\theta_i)^{l_i + \beta_i - 1} \tag{8}$$

Note that the posterior distribution is also a beta distribution. Here we assume the values of x_i are independent of θ_i, i.e., $P(D|\theta_i) = \theta_i$. Then the predictive distribution of x_i given the observations D is defined by the mean of θ_i given the observations D.

$$P(x_i | D) = \int_0^1 P(x_i|\theta_i) P(\theta_i|D) d\theta_i \tag{9}$$

$$= \int_0^1 \theta_i P(\theta_i|D) d\theta_i \tag{10}$$

$$= E(\theta_i|D) \tag{11}$$

$$= \frac{m_i + \alpha_i}{m_i + \alpha_i + l_i + \beta_i} \tag{12}$$

3.4 Dealing with Incomplete Data using Expectation Maximization

In service-oriented settings, some variables may not be observable, which means data is incomplete. In this case, we can use *expectation maximization* (EM) algorithm to calculate a optimal parameter estimation [11, 12].

The idea is that, since some variables are not observable, we can consider those variables without data as latent variables and calculate the expected values of those variables. Let $D_{observed}$ and $D_{missing}$ be the observed and missing data, respectively. Then probabilistic inference can be used to infer $P(x_i^t|D_{observed}, \theta_i^t)$, where $x_i^t \in D_{missing}$ and θ_i^t is the current parameter estimation. We can complete the *counts* (i.e., m_i and l_i) by $P(x_i^t|D_{observed}, \theta_i^t)$. This is called the E step of EM algorithm. For example, suppose there is a composed travel service T, which is composed of an underlying hotel service h. If a consumer observes T has reliability 1 at timestep t (i.e., $x_T^t = 1$) but does not observes the reliability of h, then we can use the expected reliability of h, which is $P(h = 1, T = 1)$, as the observation (i.e., $x_h^t = P(h = 1, T = 1)$). The completed data, i.e., $(x_T^t, x_h^t) = (1, P(h = 1, T = 1))$, can be used as the observations in the M step to update parameter estimation by Bayesian inference. The new parameter estimation of θ_i^{t+1} can be calculated by the posterior mean of θ_i^t. The E and M steps are executed iteratively until the convergence of the estimation. This EM process, which can be viewed as a sequential (on-line) learning method, can be repeated whenever the consumer has new observations.

3.5 Example

We can implement a sequential approach to construct and learn the service composition model from observations. Take the scenario in Figure 2 as an example, Table 1 shows the incomplete observations from a consumer in terms of response time. In the first observation, the consumer interacts with hotel service H with a satisfying response time. The consumer is also aware of the existing underlying Four Season hotel service f and its good response time. In the second observation, the consumer interacts with the car rental service C but with a bad response time. This time the consumer is not aware of any underlying services behind C. In the third observation, the consumer directly interacts with the travel service T with positive experience. It also realizes the presence of the two underlying services H and C. Service H is reported good, whereas service C is reported bad. Service C further reports the bad response time is caused by its underlying Enterprise service e.

Table 2 show the parameters estimation using Bayesian inference. The parameters are represented as a pair of hyperparameters α_i, β_i of the corresponding

t	x_f^t	x_e^t	x_H^t	x_C^t	x_T^t
1	1		1		
2	(0.67)		(0.61)	0	
3	(0.67)	0	1	0	1

Table 1. An example of observation from a consumer's experience

beta distribution. The numbers in the parentheses in Table 1 are the inferred counts to complete the missing data in E step. For example, $n(x_f^2 = 1) = E(\theta_f^1) = \frac{\alpha_f^1}{\alpha_f^1+\beta_f^1} = 0.67$. Then $n(x_H^2 = 1)$ can be inferred by

$$n(x_H^2 = 1) = n(x_H^2 = 1|x_f^2 = 1) + n(x_H^2 = 1|x_f^2 = 0) \tag{13}$$
$$= P(x_H^2 = 1|x_f^2 = 1)P(x_f^2 = 1) + P(x_H^2 = 1|x_f^2 = 0)P(x_f^2 = 0) \tag{14}$$
$$= 0.5 \times 0.33 + 0.67 \times 0.67 = 0.61 \tag{15}$$

Then the completed data can be used to update the parameter estimation. For example, the new estimation θ_H^2 (including $\theta_{H|f=0}^2$ and $\theta_{H|f=1}^2$) is given by

$$(\alpha_{H|f=1}^2, \beta_{H|f=1}^2) \tag{16}$$
$$= (\alpha_{H|f=1}^1 + n(x_H^2 = 1, x_f^2 = 1), \beta_{H|f=1}^1 + n(x_H^2 = 0, x_f^2 = 1)) \tag{17}$$
$$= (2 + P(x_H^2 = 1|x_f^2 = 1) \times x_f^2, 1 + P(x_H^2 = 0|x_f^2 = 1) \times x_f^2) \tag{18}$$
$$= (2.44, 1.22) \tag{19}$$
$$(\alpha_{H|f=0}^2, \beta_{H|f=0}^2) \tag{20}$$
$$= (\alpha_{H|f=0}^1 + n(x_H^2 = 1, x_f^2 = 0), \beta_{H|f=0}^1 + n(x_H^2 = 0, x_f^2 = 0)) \tag{21}$$
$$= (1 + P(x_H^2 = 1|x_f^2 = 0) \times (1 - x_f^2), 1 + P(x_H^2 = 0|x_f^2 = 0) \times (1 - x_f^2)) \tag{22}$$
$$= (1.17, 1.17) \tag{23}$$

t	θ_f^t	θ_e^t	$\theta_{H\|f=0}^t$	$\theta_{H\|f=1}^t$	$\theta_{C\|e=0}^t$	$\theta_{C\|e=1}^t$
0	(1,1)		(1,1)	(1,1)		
1	(2,1)		(1,1)	(2,1)	(1,1)	
2	(2.67,1.33)	(1,1)	(1.17,1.17)	(2.44,1.22)	(1,2)	(1,2)
3	(3.33,1.67)	(1,2)	(1.5,1.17)	(3.11,1.22)	(1,3)	(1,2)

Table 2. The parameter estimation based on the observations

Note that some parameters may not exist until particular observation because the consumer is not aware of the corresponding random variables. For example, service C does not exist until the second observation. The conditional dependencies may change because some underlying services may not be discovered in the first place. For example, $\theta^1_{C|e=0}$ actually means θ^1_C in the first observation because service e does not exist. However, θ^2_C changes to $\theta^2_{C|e=0}$ and $\theta^2_{C|e=1}$ is initialized because service e and the dependency on service C are discovered in the third observation. In these cases, the Bayesian network is updated at the same time to reflect new discovery.

4 Evaluation

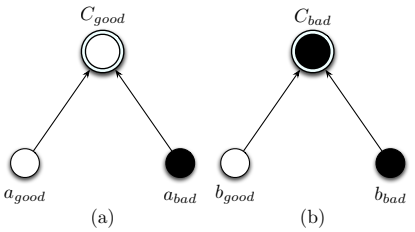

Fig. 3. Two basic experiment scenarios

4.1 Service Selection with Missing Data

Now we evaluate our trust aware service selection model by showing it can reward/punish underlying services in an appropriate way so that consumers and composed services will become reluctant to interact with low reputation services. Two basic scenarios are considered as shown in Figure 3. Shaded nodes represent bad services, which provides unsatisfiable QoS with high probability 0.8, whereas good services provide 80% satisfying QoS. To enable our service composition assumption that underlying services may not be exposed to the consumer, we introduce δ as the percentage of missing data. In each scenario, the consumer interacts with the composed service for d times. Each time the composed service may report the QoS metric of each of its underlying service with independent probability δ. The service selection model is updated sequentially. We measure the quality of the estimation by *root mean square error* (RMS). We also show how the trust (i.e., parameter θ) in services changes over time, and how the performance of underlying services affect the composed services and the whole composition by comparing the parameter θ and the joint probability.

Figure 4 shows the error of trust in a_{good} service in the first experiment scenario for 20% and 40% missing data (i.e., $\delta = 0.2$ or 0.4). The total number of observations d is 100. The trust learned from 40% missing data captures a_{good}'s behavior more slowly than the one learned from 20% missing data. Also, other results show that our approach successfully reward or punish underlying services based on incomplete observations. For example, with $\delta = 0.4$, $P(C_{good} = 1|a_{bad} = 0)$ and $P(C_{good} = 1|a_{good} = 1)$ are 0.78 and 0.76, respectively. $P(C_{bad} =$

$1|b_{good} = 1)$ and $P(C_{bad} = 1|b_{bad} = 0)$ are 0.12 and 0.13, respectively. This result indicates our model correctly evaluates underlying services with 40% missing data, regardless of the goodness or badness of the composed service.

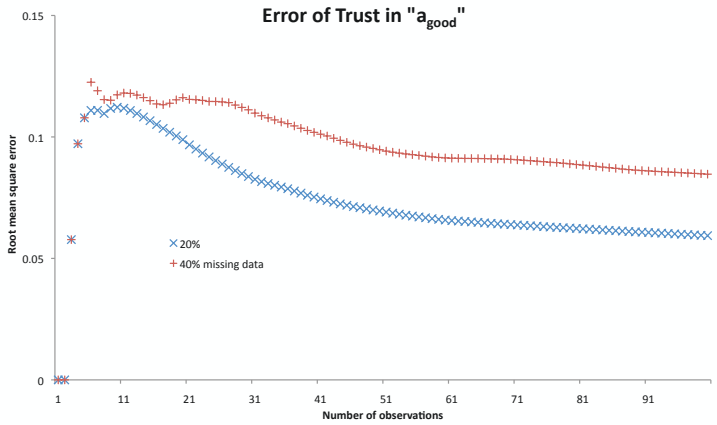

Fig. 4. Error of trust in a_{good} for 20% and 40% missing data

4.2 Service Selection with Dynamic Service Providers

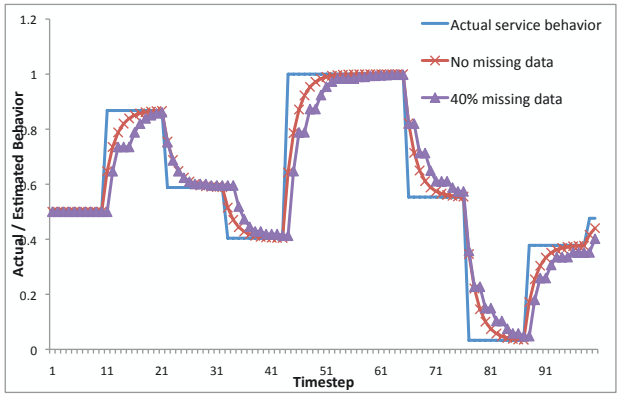

Fig. 5. Tracking a random walk service for different percentages of missing data

Our second evaluation examines the ability of tracking the dynamic behavior of services. We introduce two behavior profiles: *random walk* and *damping*. The random walk profile models the general predictable behavior. The random walk service changes behavior every certain period. Its current behavior x^t depends on the previous behavior x^{t-1}, defined as $x^t = x^{t-1} + \gamma U(-1, 1)$, where γ is a

real number between 0 and 1, and $U(-1,1)$ represents the uniform distribution from -1 to 1. In our settings, the random walk service changes behavior every ten timesteps, and $\gamma = 0.8$. The damping profile models the service who turns bad once its reputation is built. Its behavior is defined as $x^t = 1$ when $t \leq T$, and $x^t = 0$ otherwise, where T is the total number of timesteps. Additionally, a discount factor ϕ is used when we calculate posterior distribution in Equation 7, which becomes $P(x_i|D) = \frac{m_i+\phi\alpha_i}{m_i+\phi\alpha_i+l_i+\phi\beta_i}$. The discount factor is a common idea in trust and reputation systems. The estimate reflects the overall behavior if it is high; otherwise, the estimate depends more on the recent behavior. The study of the effect of the discount factor can be found in [13]. Here we set $\phi = 0.6$.

Figure 5 shows how our trust values track the actual behavior of the random walk service with 0% and 40% missing data. The result shows our approach captures the dynamism of the random walk service, although the missing data does slow down the convergence. Figure 6 shows the similar result of tracking damping service.

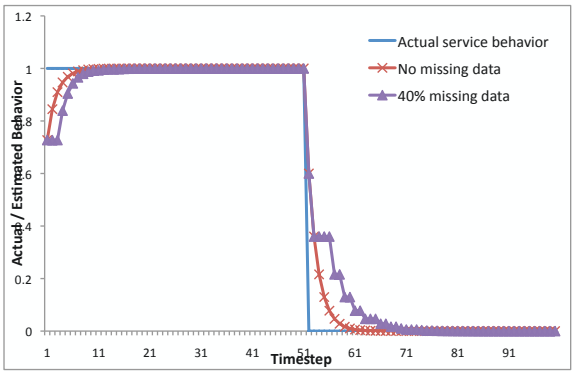

Fig. 6. Tracking a damping service for different percentages of missing data

5 Conclusion

This paper present a trust-aware service selection model in service-oriented environments. The model is built on a Bayesian network to capture the relationships between services. The trust information, which can be integrated with our previous trust model, is learned sequentially from both direct observations and indirect evidence in terms of QoS properties. The main feature of this model is it can deal with incomplete observations, which is as a result of the fact that the underlying services behind service composition may not be observable. Consumers maintain its own knowledge of the environment locally and exchange information each other. Our model rewards services with good QoS metrics and punishes those with bad metrics in the way that consumers will be reluctant to interact with services with low reputation.

Our future work is to refine and enhance an existing QoS ontology from [1] to fit it into our approach. This ontology will be able to capture SLAs as well as

the requirements of consumers and advertisements from services regarding SLAs. Both domain-independent and domain-specific QoS properties can be defined in our ontology. Thus, we can further evaluate the QoS properties by comparing the QoS metrics and SLAs, and the sociability of referrers by our trust framework. Knowing the sociability can yield more accurate trust information from referrals. We will study how referrals improve the convergence of trust estimation. We will also apply multiattribute utility theory for decision-making, based on the trust information. Finally, the EM-based parameter estimation in our model can be upgraded to *Structural EM* [14], which can not only learn the trust information but also the graph structure. The learned structure can be used as a suggestion of service composition.

Acknowledgement

This work is supported by the U.S. Army Research Office (ARO) under grant W911NF-08-1-0105 managed by NCSU Secure Open Systems Initiative (SOSI).

References

1. Maximilien, E.M., Singh, M.P.: A framework and ontology for dynamic web services selection. IEEE Internet Computing **8**(5) (September 2004) 84–93
2. Menascé, D.A.: Composing web services: A QoS view. IEEE Internet Computing **8**(6) (2004) 88–90
3. Milanovic, N., Malek, M.: Current solutions for web service composition. IEEE Internet Computing **8**(6) (2004) 51–59
4. Wu, G., Wei, J., Qiao, X., Li, L.: A Bayesian network based QoS assessment model for web services. In: IEEE International Conference on Services Computing. (2007) 498–505
5. Yue, K., Liu, W., Li, W.: Towards web services composition based on the mining and reasoning of their causal relationships. In: APWeb/WAIM. (2007) 777–784
6. Wang, Y., Singh, M.P.: Trust representation and aggregation in a distributed agent system. In: Proc. of the AAAI, Menlo Park, (2006) 1425–1430
7. Wang, Y., Singh, M.P.: Formal trust model for multiagent systems. In: Proc. of IJCAI, Detroit, (2007) 1551–1556
8. Hang, C.W., Wang, Y., Singh, M.P.: Operators for propagating trust and their evaluation in social networks. In: Proc. of AAMAS. (2009)
9. Buntine, W.L.: Operations for learning with graphical models. Journal of Artificial Intelligence Research **2** (1994) 159–225
10. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer (August 2006)
11. Lauritzen, S.L.: The EM algorithm for graphical association models with missing data. Computational Statistics & Data Analysis **19**(2) (1995) 191–201
12. Singh, M.: Learning Bayesian networks from incomplete data. In: Proc. of AAAI, AAAI Press (1997) 534–539
13. Hang, C.W., Wang, Y., Singh, M.P.: An adaptive probabilistic trust model and its evaluation. In: Proc. of AAMAS. (2008)
14. Friedman, N.: The Bayesian structural EM algorithm. In: Proc. of UAI '98, San Francisco, CA, USA. (1998) 129–138