Trustworthy Service Selection and Composition

Chung-Wei Hang and Munindar P. Singh
North Carolina State University

We consider service-oriented computing (SOC) environments. Such environments are populated with services that stand proxy for a variety of information resources. A fundamental challenge in SOC is to select and compose services to support specified user needs—or to provide additional services. Existing approaches for service selection either fail to capture the dynamic relationships between services or assume that the environments are fully observable. In practical situations, however, consumers are often not aware of how the services are implemented. We propose two distributed trust-aware service selection approaches. One is based on Bayesian networks; the other is built on a beta-mixture model. We experimentally validate our approach through a simulation study. Our results show that both approaches accurately punish and reward services in terms of the qualities they offer, and further that the approaches are effective despite incomplete observations regarding the services under consideration.

Categories and Subject Descriptors: H.3.5 [**Information Storage and Retrieval**]: Online Information Services

General Terms: Algorithms, Experimentation

Additional Key Words and Phrases: Trust, probabilistic model, service selection, service composition

1. INTRODUCTION

In service-oriented computing (SOC) [Singh and Huhns 2005] environments, computing resources are modeled as *services*, which can be used directly or composed into other services. Services are being widely adopted in modern distributed environments, such as for cloud computing [Amazon.com 2009].

However, there may exist many services with similar functional properties. For example, there are many practical services providing airline tickets, such as the various airlines and travel agencies. Therefore, distinguishing and selecting services with the desired nonfunctional characteristics becomes essential to consumers (and composed services, which consume their underlying services). We address the problem of selecting services based on criteria such as user requirements and service qualities.

We understand quality of service (QoS) in a broad sense to include potentially any service quality of interest to a consumer, not merely the performance-oriented qualities such as throughput and availability. Further, in general, the qualities offered by a service instance would vary over time. For example, the latency of a shopping web service can change with the load it faces.

Traditional SOC approaches do not address service selection as such but only service discovery. Specifically, they seek to capture descriptions of services in a representation such as *Web Service Definition Language (WSDL)*, which characterizes the functionality (that is, the methods) a service supports and the ports where it can be accessed, but not the qualities of service it offers. Even the semantic approaches, such as OWL-S, which characterize the functionality of a service more precisely than WSDL do not address its qualities. In other words, traditional SOC approaches are confined to considering the functional properties of services as a basis for matching services to user needs. The functional properties are generally defined for service types. In a practical setting, however, a successful service enactment episode depends not just on the service types but on the specific service instances involved. Moreover, the qualities offered by a service instance might vary over time, sometimes rapidly. Our approach considers service qualities as they apply to instances.

The research on *trust* modeling in artificial intelligence provides us with a promising starting point for a solution to service selection. Trust is a key basis of interaction in an open setting, indicating the relationships between the parties involved. For example, in a service-oriented context, a party Alice may trust another party

Bob, because Alice expects Bob will provide a desired service with the expected quality. The trustworthiness of the parties can be defined by both functional and nonfunctional properties. We define *trust-aware service selection* as selecting desired services based on the trust placed in their ability to deliver specified values of the specified qualities.

Maximilien and Singh [2004] develop a trust-aware approach to select services based on a well-defined ontology [Maximilien and Singh 2004] that provides a basis for describing consumers' requirements and providers' advertisements. The ontology enables consumers to define nonfunctional properties. Unfortunately, Maximilien and Singh's approach fails to take service composition into consideration. When services are composed, the services underlying a composed service may not be shown externally to the consumers. Service composition can be divided into many scenarios [Menascé 2004] and these scenarios can be nested. This makes information about service qualities difficult to collect and evaluate. Consequently, service selection is more complicated with traditional approaches because the consumers may not even know with whom they are interacting.

Contributions and Relevance to this Issue

An ideal trust-aware service selection approach should support the following operations.

—Selecting service instances to form suitable compositions based on the qualities desired.

—Rewarding and punishing underlying services in an appropriate manner so as to maintain the best information as needed to support successful compositions.

This paper provides just such a trust-aware service selection approach. It presents a formal service selection model in probabilistic terms and develops approaches applying which a consumer may monitor and explore desired service compositions. This paper shows how our approach rewards and punishes the services involved dynamically despite incomplete knowledge of the composition. An important contribution is that this paper shows how to treat the relationships between some key service composition operators and different types of service qualities in a systematic manner.

In this manner, this paper addresses the themes of adaptive service selection, from the standpoint of service composition, which has largely been ignored in the literature.

2. RELATED WORK

Milanovic and Malek [2004] compare various modern web service composition approaches. They identify four necessary requirements for service composition: connectivity, nonfunctional qualities, correctness, and scalability. However, Milanovic and Malek's definition of service qualities is not extensible. Our approach, in contrast, is extensible, and can deal with a changing set of service qualities.

Menascé [2004] studies how qualities of service are aggregated in different service composition scenarios. However, this approach requires the knowledge of the specific service composition under composition. For example, service A invokes service B, which may invoke C and D with probability p_c and p_d, respectively. This information is not always available because of two reasons. First, the providers have no incentive to reveal such information. Second, modeling the invocation probabilities is not trivial. By contrast, our service composition model makes no such assumptions. Our approach monitors and explores the desired services dynamically.

Wu et al. [2007] model a consumer's assessment of a service's quality via a naïve Bayes network, where the root represents the overall capability of a service and a child represents the capability of a particular quality of the service. Wu *et al.* apply a fuzzy representation to express the levels of the service capabilities. Their approach enables consumers to estimate the overall quality assessment. In contrast, our approach uses a Bayesian network to model service composition to evaluate each quality of a service separately. Then a consumer can select services based on its preferences among the various qualities.

Lin et al. [2008] select services according to the consensus of group preference order of various qualities. Consumers express their preferences among the values of the qualities in fuzzy terms. Lin *et al.* use fuzzy logic to resolve the conflicts between the subjective interpretations of service qualities from each consumer. Then they aggregate different fuzzy views from both consumers and providers to reach a consensus of preferred order of quality metrics. Similar to Wu et al. [2007], Lin *et al.* enable consumers to consider more than one quality in combination. Our approach treats each quality separately. Consumers express their subjective preferences in terms of trust. Consumers may show different levels of trust to the same service because of their subjective interpretations of quality metrics. Also, one can combine others' subjective trust by inferring it from their trustworthiness, which reflects the similarity of the subjectiveness.

Yue *et al.*'s work [2007] is the closest to ours. They propose a Bayesian network-based approach to model the relationships between elementary services. Yue *et al.*'s approach constructs web service Bayesian networks (WSBN) based on the invocations between the services. Then the service composition guidance can be made from the *Markov Blanket* [Pearl 1988] of a given service. However, this approach fails to consider the dynamism of service composition because the relationships are fixed. Our model captures the dynamism by updating the Bayesian network, which will subsequently affect the trustworthiness of a service.

Liu [2005] views a service-oriented environment as an ecosystem. Via this analogy, Liu tries to explore the trustworthy service selection and composition problem from three levels: (1) trust: atomic service (service selection), (2) composition: composite service (service composition), and (3) emergent behavior: network economy (organizational behaviors, consumer communities, business alliances, and trusted third parties). At the trust level, Liu points out an ideal trust representation in a service-oriented environment should be (a) flexible and adaptive to suit diversified needs of agents, and (b) exchangeable so agents can help each other. Liu adopts XML Topic Maps for knowledge representation. Then she applies collaborative filtering to select services based on their semantic similarity. At the composition level, Liu suggests using ant crawling for consumers to discover semantically similar services and further classify those into clusters, thus generating a new composition plan. Finally, at the emergent behavior level, the topology of the environment evolves based on low-level interactions between agents. The connections in this topology is defined by a referral network. A service survives only when it is needed by others. New services are born from composition plans. Useless services are eliminated. Unfortunately, Liu does not implement or evaluate her ideas. However, we agree with Liu on the hierarchical view of the trustworthy service selection and composition.

Paradesi *et al.* [2009] build a trust framework for web service compositions. They adopt the trust representation from Wang and Singh [2007] and add operators for combining trust in different types of service compositions including sequence, concurrent, conditional, and loops. In contrast, instead of service compositions, we study how quality is composed in these types of service compositions. Our experiments show our approaches are general enough to deal with various types of quality composition.

3. TRUST-AWARE SERVICE SELECTION MODEL

We imagine trustworthy service-oriented computing as consisting of three important levels.

—*Trust.* At the trust level, consumers and atomic services discover and interact with each other. Trust is built from those interactions. Also, trust information (referrals) is exchanged. Consumers and services require a trust framework that can formalize, aggregate, combine, and update trust. Our previous work supports this level [Wang and Singh 2006; 2007; Hang et al. 2008; 2009].

—*Composition.* At the composition level, atomic services are composed into composite services, which can be further composed into composite services. A trust-aware service composition model is required so that consumers can express trust information through relationships in the compositions; incorporate unobserved underlying components; suggest better compositions based on past observations; and integrate with the trust framework. This paper develops probabilistic solutions for this level.

—*Community.* After trust is built up at the trust and composition levels, new composite services are born and

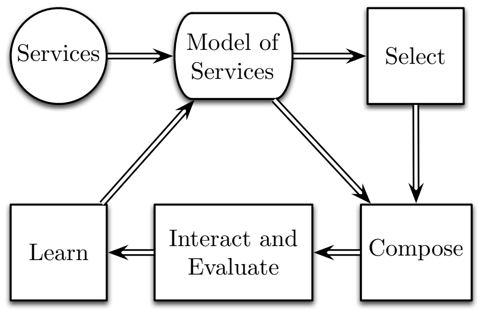

Fig. 1. Trustworthy service selection architecture

low reputation services are forgotten. Consumers prefer to interact with reputable services, and avoid those with a poor reputation. As a result, the system evolves into communities where services interact highly with each other. Besides, *authorities*—the dominating services in particular areas—are identified. At this level, a good measurement (for example, *entropy*, as Liu [2005] suggests) is needed to determine if the emergent behavior driven by the trust mechanism improves the overall performance. Although this paper seeks to lay the groundwork for such communities, it defers specific study of communities to future work.

3.1 Overview

We represent trust based on the *beta probability distribution* [Evans et al. 2000], which can be integrated with Wang and Singh's model [2006; 2007]. Intuitively, the trustworthiness of a service should be estimated based on both direct and indirect experience. Direct experience means the previous quality of service received from the service, whereas indirect experience comes from referrals by peers. Some of our previous work addresses how to model trust from indirect experience [Hang et al. 2009]. This is beyond our present scope.

Estimating trust from direct experience is not straightforward, because some services may not directly expose details of their composition to their consumers. A consumer may interact with a composed service without knowing about the services that underlie it. In such a case, evaluating the trustworthiness of a service is no longer easy. For example, a consumer 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 consumer is not satisfied with the composed service because of its late response time. The model should penalize the composed service, as well as some of the underlying ones. If the hotel service, for instance, is reported to be the cause of an unsatisfactory quality value, the model should reflect the changes in the way that consumers or other composed services would become reluctant to interact with it. Also, as the amount of experience of the rater (as captured in the model) increases, the model should be able to suggest superior compositions.

Each consumer maintains its own local model to guide itself to reward or penalize services based on its direct interactions with them. Figure 1 shows our architecture. Several services exist in the computational environment. In one scenario, a consumer maintains models of some or all of the available services. Using this model, it selects some services and composes them into a composite service. Next, the consumer interacts with and evaluates the composite service in terms of the outcomes with respect to the service qualities of interest. Based on these outcomes, the consumer applies a learning method to update the model it is maintaining for the services. In an alternative scenario, the consumer may not be responsible for composing services and would simply select an atomic service or a composite service that another party has composed. In this case, it would need to learn about the services with less information than in the first scenario. Our approach handles both of these scenarios.

We now introduce our proposed approaches. Section 3.2 presents the *Bayesian* approach, which models service compositions via Bayesian networks in partially observable settings. Bayesian approach captures the dependency of providing good service between composite and underlying services. It also adaptively updates trust to reflect most recent quality. Section 3.3 describes the *Beta-Mixture* approach. This approach can learn not only the distribution of composite quality, but also the underlying services' "responsibility" in composite quality without actually observing the underlying performance. These two approaches provide different information about services. Bayesian approach uses online learning to track the service behavior over time. It also tells consumers how good service they can expect from composition when the underlying services are good. Beta-mixture model learns the quality distribution of services and provides how much each underlying service contributes in the composition.

3.2 Bayesian Approach

3.2.1 *Bayesian Network-Based Graph Representation.* The purpose of modeling service composition is to model how a certain quality 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 and flight services. If the underlying service is not reliable, the composed service is possibly not reliable either. Thus, the composition model should be able not only to represent the relationships between services, but also to capture the dependency between them. Of course, it may turn out that the qualities of underlying services do not influence the composed service. 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 would not correspond trivially to the trustworthiness of that underlying service.

We introduce a Bayesian network-based service selection approach, which can construct models from the *incomplete observations* (direct experience) of a consumer. Here, we emphasize incomplete observations because not all qualities are observable from the consumers' point of view. For simplicity, we normalize the qualities to the real interval $[0, 1]$. Thus we represent an observation of a particular quality of a service instance d at time t as a real number x_d^t between 0 and 1. Some qualities, say, error, can be simply considered as 1 (positive) or 0 (negative). We write an observation D^t of the whole composition at time t as $D^t = (x_1^t, x_2^t, \dots, x_d^t)$, where d is the number of services in the composition.

A Bayesian network is a directed acyclic graph $G = \langle V, R\rangle$ with random variables V as nodes, and edges R as the direct relationships between variables []. We denote atomic and composite services with uppercase and lowercase, respectively. An edge from service a to B means B is composed of a. In Bayesian network terminology, the source node of an edge is the parent of its target. Thus, a is B's *parent* and B is a *child* of a. Notice that this terminology is opposite to the typical service composition rendition as a figure where the composite service would be a parent (or ancestor) of the constituent services. We use the Bayesian network terminology in this paper.

Note that an edge can be only arise from either an atomic or a composite service to a composite service, because atomic services cannot be composed. A conditional probability associated with each node represents trust (a probability) of the node variable given its parent trust values. Let each node in the Bayesian network equal trust, that is, the probability of obtaining a good outcome from the service corresponding to node. The good outcome in this case depends on a specified quality. An edge represents the relationship of composition. For example, in Figure 2, the composed hotel service H is composed of the Four Seasons hotel service f, that is, f is a parent of H. Then the trust of node H is the probability of obtaining a good outcome in terms of a particular quality value from H, given f provides a good outcome. T, a travel service, is composed of hotel service H and car rental service C, which is itself composed of the Enterprise Car Rental service e.

The Bayesian network models the relationships between services. The conditional probability table associated with each node provides consumers a basis for determining how much responsibility to assign a service that underlies a service composition. Thus, the consumers can view the conditional probabilities as the level of trust they place in the services in the composition. One can view trust as a probability, a real number

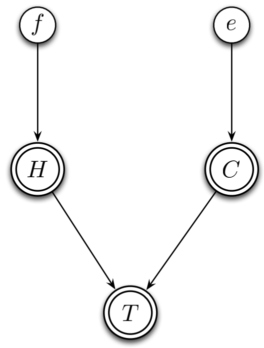

Fig. 2. Service composition example

from 0 to 1. For example, in Figure 2, trust of the consumer in service T has two parts. One part is the conditional probability of obtaining a satisfactory quality value from T given obtaining a satisfactory quality value from H. The other part is the conditional probability of obtaining a satisfactory quality value from T given obtaining a satisfactory quality value from H. The overall trust placed in T is the marginal probability $P(T)$ (that is, the probability of obtaining a satisfactory quality value from T). $P(T)$ can be calculated by marginalizing over all the parents of T. That is, we have

$$
\begin{aligned}
P(T) = \; & P(T|H=0,C=0)P(H=0,C=0) && (1)\\
& +P(T|H=0,C=1)P(H=0,C=1) && (2)\\
& +P(T|H=1,C=0)P(H=1,C=0) && (3)\\
& +P(T|H=1,C=1)P(H=1,C=1) && (4)
\end{aligned}
$$

The network can be constructed and trustworthiness of the various providers can be learned from the consumers' direct experience. Section 3.2.2 explains how trustworthiness as a real number from 0 to 1 can be learned. Section 3.2.3 describes some drawbacks of this representation and introduces trust as a beta distribution, which is our main trust representation in the proposed approach.

3.2.2 *Parameter (Trust) Estimation.* Given an acyclic Bayesian network graph G over d variables, x_1, x_2, …, x_d, the associated joint distribution is written as

$$P(x_1, \ldots, x_d) = \prod_{i=1}^{d} P(x_i|x_{pa_i}) = \prod_{i=1}^{d} \theta_i \tag{5}$$

where θ_i is the conditional probability $P(x_i|x_{pa_i})$, and x_{pa_i} is the set of parent variables of x_i. Suppose the consumer obtains n complete observations, $D = \{(x_1^t, \ldots, x_d^t), t = 1, \ldots, n\}$. In a fully observable environment, θ_i can be learned from the observed data by *maximum likelihood estimation* (MLE) [Buntine 1994].

In our model, each parameter θ_i represents trust—the conditional probability of obtaining a good outcome from x_i given obtaining a good outcome from each of the services in x_{pa_i}. We assume that all variables x_i are pairwise independent and identically distributed (i.i.d.). θ is the set of all parameters θ_i. The likelihood function is defined as the probability of the observations given the parameters. Following Bishop [2006], we write this as:

$$
\begin{aligned}
P(D|\theta) &= \prod_{t=1}^{n} P(x_1^t, \ldots, x_d^t|\theta) && (6)\\
&= \prod_{t=1}^{n}\prod_{i=1}^{d} \theta_i && (7)\\
&= \prod_{i=1}^{d}\prod_{x_i, x_{pa_i}} \theta_i^{n(x_i, x_{pa_i})} && (8)\\
&= \prod_{i=1}^{d} \theta_i^{m_i}(1-\theta_i)^{l_i} && (9)
\end{aligned}
$$

where $n(x_i, x_{pa_i})$ is the number of observations that satisfy the variable assignment, $m_i = n(x_i, x_{pa_i})$, and $l_i = n(x_{pa_i}) - m_i$. Then, given the observations, the parameters that maximize the likelihood are

$$\hat{\theta}_i = \frac{m_i}{m_i + l_i}.$$

For example, let a consumer have 10 good outcomes out of 15 interactions with service x_i, given that x_{pa_i} provides good services. Then we have, $m_i = n(x_i = 1, x_{pa_i} = 1) = 10$ and $l_i = n(x_{pa_i} = 1) - m_i = 15 - 10 = 5$. The consumer can calculate that the estimated trust value $\hat{\theta}_i$ from these observations is $\frac{10}{15}$. That is, using MLE, a consumer can estimate the trust value of a service from the consumer's observations of it.

3.2.3 *Bayesian Inference.* Note that when the number of observations is small, MLE may yield overfitted results. Consider an extreme case where $x_i^t = 1$ for $t = 1, \ldots, n$. That is, all the observations are the best possible. The parameter $\hat{\theta}_i$ maximizing the likelihood is $\frac{n}{n} = 1$, 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 [Bishop 2006, chap. 2].

$$P(\theta_i) = \frac{\Gamma(\alpha_i + \beta_i)}{\Gamma(\alpha_i)\Gamma(\beta_i)}\theta_i^{\alpha_i - 1}(1-\theta_i)^{\beta_i - 1} \tag{10}$$

Here α_i and β_i are *hyperparameters* controlling the distribution of the parameter θ_i, and $\Gamma(x) = \int_0^\infty u^{x-1}e^{-u}du$. The coefficient $\frac{\Gamma(\alpha_i+\beta_i)}{\Gamma(\alpha_i)\Gamma(\beta_i)}$ in Equation 10 ensures $\int_0^1 P(\theta_i)d\theta_i = 1$. We simplify the coefficient to a function B of the hyperparameters α_i and β_i, yielding,

$$P(\theta_i) = B(\alpha_i, \beta_i)\theta_i^{\alpha_i - 1}(1-\theta_i)^{\beta_i - 1} \tag{11}$$

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.

$$P(\theta_i|D) = \frac{P(D|\theta_i)P(\theta_i)}{P(D)} \tag{12}$$

That is, the posterior distribution $P(\theta_i|D)$ is proportional to the multiplication of the prior $P(\theta_i)$ and the likelihood function $P(D|\theta_i)$. Now we combine Equations 9, 11, and 12 to obtain

$$P(\theta_i|D) \;=\; B(m_i + \alpha_i, l_i + \beta_i)\theta_i^{m+\alpha_i - 1}(1-\theta_i)^{l_i + \beta_i - 1} \tag{13}$$

Note that the posterior distribution is also a beta distribution with hyperparameters $m_i + \alpha_i$ and $l_i + \beta_i$. Here we assume the values of x_i are independent of θ_i, that is, $P(D|\theta_i) = \theta_i$. Then the predictive distribution of x_i given the observations D is defined by the mean of θ_i given the observations D. This enables consumers to learn the parameters from the observations without the problems caused by MLE in some extreme cases.

$$\begin{aligned} P(x_i|D) &= \int_0^1 P(x_i|\theta_i)P(\theta_i|D)d\theta_i && (14) \\ &= \int_0^1 \theta_i P(\theta_i|D)d\theta_i && (15) \\ &= E(\theta_i|D) && (16) \\ &= \frac{m_i + \alpha_i}{m_i + \alpha_i + l_i + \beta_i} && (17) \end{aligned}$$

Bayesian inference provides an intuitive way to update trust (a beta distribution) of a service. For example, let a consumer's current trust value of service x_i be $\theta_i = (\alpha_i, \beta_i) = (5, 5)$. Suppose the consumer observes two new good outcomes and one bad outcome. The consumer can update the trust value by simply adding the new observations to the previous value. That is, $\hat{\theta}_i = (\hat{\alpha}_i, \hat{\beta}_i) = (7, 6)$. Then the consumer can predict that the probability of obtaining a satisfactory quality value from the next interaction is $\frac{7}{13}$.

Additionally, to incorporate the dynamism of service behavior, a discount factor γ reduces the impact of the old information when we calculate the posterior distribution. In other words, instead of Equation 17 we have:

$$P(x_i|D) = \frac{m_i + \gamma\alpha_i}{m_i + \gamma\alpha_i + l_i + \gamma\beta_i}. \qquad (18)$$

The notion of a discount factor is common in trust and reputation systems. The estimate reflects the overall behavior if it is high; otherwise, the estimate depends more on the recent behavior. Hang et al. [2008] study the effect of the discount factor on updating trust estimates. Section 4.2 shows how our approach keeps track of dynamic service behavior in a service composition.

3.2.4 *Dealing with Incomplete Data.* Quite often in service-oriented settings, some variables may not be observable, meaning that the data would be incomplete. In this case, we can use *expectation maximization* (EM) to calculate optimal parameter estimation [Lauritzen 1995; Singh 1997].

The idea here is that, since some variables are not observable, we can consider the 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 we can 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 using *exact inference* like *variable elimination* [Zhang and Poole 1996]. We can complete the *counts* (that is, m_i and l_i) by $P(x_i^t|D_{observed}, \theta_i^t)$. This is called the *E* step of the EM algorithm.

For example, suppose there is a travel service T, which is composed of an underlying hotel service h. If a consumer observes that T has reliability 1 at time-step *t* (that is, $x_T^t = 1$) but does not observe the reliability of h at time t, then we can use the expected reliability of h, $P(h = 1, T = 1)$, as the observation (that is, $x_h^t = P(h = 1, T = 1)$). The completed data, that is, $(x_T^t, x_h^t) = (1, P(h = 1, T = 1))$, can be used as the observation in the *M* step to update the parameter estimates by Bayesian inference as described in Section 3.2.3. 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 estimation converges [Dempster et al. 1977]. This EM process, which can be viewed as a sequential (on-line) learning method, can be repeated whenever the consumer makes new observations.

Table I. An example observation derived from a consumer's experience

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

3.2.5 *Extended Example.* We can implement a sequential approach to construct and learn the service composition model from observations. Taking the scenario of Figure 2 as an example, Table I shows the incomplete observations from a consumer in terms of its response time. In the first observation, the consumer interacts with the hotel service H and obtains a satisfactory response time. The consumer is also aware of the existing underlying Four Seasons 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. Here the consumer is not aware of any underlying services. In the third observation, the consumer directly interacts with the travel service T with a positive experience. Here the consumer also realizes the presence of the two underlying services H and C. T reports service H as offering good outcomes and service C as offering bad outcomes. Service C further reports its bad response time as having been caused by its underlying Enterprise service e.

Table II shows the parameters estimation using Bayesian inference. The parameters are represented as a pair of hyperparameters α_i, β_i of the corresponding beta distribution. The numbers in the parentheses in Table I are the inferred counts to complete the missing data in the 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 we can infer $n(x_H^2 = 1)$ as follows.

$$\begin{aligned} n(x_H^2 = 1) &= n(x_H^2 = 1|x_f^2 = 1) + n(x_H^2 = 1|x_f^2 = 0) \\ &= P(x_H^2 = 1|x_f^2 = 1)P(x_f^2 = 1) + P(x_H^2 = 1|x_f^2 = 0)P(x_f^2 = 0) \\ &= 0.5 \times 0.33 + 0.67 \times 0.67 = 0.61 \end{aligned}$$

Subsequently, we use the completed data 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

$$\begin{aligned} &(\alpha_{H|f=1}^2, \beta_{H|f=1}^2) \\ &= (\alpha_{H|f=1}^1 + n(x_H^2 = 1, x_f^2 = 1), \beta_{H|f=1}^1 + n(x_H^2 = 0, x_f^2 = 1)) \\ &= (2 + P(x_H^2 = 1|x_f^2 = 1) \times x_f^2, 1 + P(x_H^2 = 0|x_f^2 = 1) \times x_f^2) \\ &= (2.44, 1.22) \\ &(\alpha_{H|f=0}^2, \beta_{H|f=0}^2) \\ &= (\alpha_{H|f=0}^1 + n(x_H^2 = 1, x_f^2 = 0), \beta_{H|f=0}^1 + n(x_H^2 = 0, x_f^2 = 0)) \\ &= (1 + P(x_H^2 = 1|x_f^2 = 0) \times (1 - x_f^2), 1 + P(x_H^2 = 0|x_f^2 = 0) \times (1 - x_f^2)) \\ &= (1.17, 1.17) \end{aligned}$$

Note that some parameters may not exist until a particular observation because the consumer may not be aware of the corresponding random variables. For example, service C is not reported until the second observation. Further the conditional dependencies may change because some underlying services may be observed later. For example, $\theta_{C|e=0}^1$ actually means θ_C^1 in the first observation because service e is not reported. However, θ_C^2 changes to $\theta_{C|e=0}^2$ and $\theta_{C|e=1}^2$ 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

Table II. Parameter estimation based on the observations of Table I

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)

time to reflect the new discovery.

3.3 Beta-Mixture Approach

3.3.1 *Finite Mixture Models.* Finite mixture models are powerful statistical probabilistic tools for modeling complex data [McLachlan and Peel 2000]. They have been widely used in machine learning, bioinformatics, computer vision, and pattern recognition domains. One of the most popular mixtures for continuous data is the Gaussian mixture.

In general, finite mixture models can be viewed as the superposition of multiple probability density components. Suppose there are component distributions. Then the finite mixture model can be formulated as

$$p(D) = \sum_{k=1}^{K} \pi_k p_k(D|\theta_k), \tag{19}$$

where $D = \{x_1, \ldots, x_N\}$ are the observations, p_k is the k^{th} component distribution with parameter θ_k, and π_k is the mixing coefficient. Mixing coefficients, which are also probabilities, control the portion of each component in the linear combination of the whole mixture, that is, $\sum_{k=1}^{K} \pi_k = 1$ and $0 \leq \pi_k \leq 1$.

The mixture distribution is governed by π and Θ. These parameters can be estimated by maximizing the log likelihood function using the EM algorithm

$$L(\Theta) = \ln p(D|\pi\Theta) = \sum_{i=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k p(x_i|\theta_k) \right\}. \tag{20}$$

Define binary latent random variable z_k as the indicator of if an observation is from component k, where only one particular $z_k = 1$ while others are zero. Thus, $p(z_k = 1) = \pi_k$, $p(z) = \prod_{k=1}^{K} \pi_k^{z_k}$, and $p(D|z_k = 1) = p_k(D|\theta_k)$. Then the distribution can be rewritten as

$$p(D) = \sum_z p(z)p(x|z) = \sum_{k=1}^{K} \pi_k p_k(D|\theta_k) \tag{21}$$

The E step first uses current parameters Θ^{old} to compute the posterior distribution $p(z|D, \Theta^{old})$. Then it uses the posterior distribution to calculate the expectation of the log likelihood function as

$$Q(\Theta, \Theta^{old}) = E_{\Theta^{old}}(L|D) = \sum_z p(z|D, \Theta^{old}) \ln p(D, z|\Theta). \tag{22}$$

In the M step, the expectation is maximized to determine the new parameter $\Theta^{new} = \arg\max_\Theta Q(\Theta, \Theta^{old})$. Then the log likelihood with new parameters is checked for convergence; otherwise repeat E and M steps.

3.3.2 *Beta-Mixture Model.* Although our observations (that is, trust values) are continuous, instead of Gaussian mixture, we use beta-mixture model [Bouguila et al. 2006] for two reasons. First, our trust values lie between 0 and 1. The beta distribution is designed for the distribution in a certain interval. Second, the beta distribution can be integrated with our trust framework, which is also based on beta distribution.

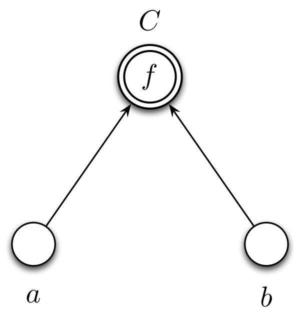

Fig. 3. Composition operator f defines how quality is composed in a composition

Quality	**Sequence**	**Flow**	**Case**
Latency	SUM	MAX	SWITCH
Throughput	MIN	SUM	SWITCH
Failure	PRODUCT	PRODUCT	SWITCH

Table III. Composition operator examples of different qualities and their interaction types

For each composition, we use a beta mixture to model the trust distribution. The number of components is the number of the direct underlying services in the composition. Each component is a beta distribution.

4. EXPERIMENTAL EVALUATION

To simulate different types of compositions, we consider composition operators as commonly defined in leading business process and scientific workflow approaches. Specifically, we consider the Web Services Business Process Execution Language (WS-BPEL) [BPEL 2007]. WS-BPEL defines three types of interactions between web services, including *sequence*, *case*, and *flow* (which executes the constituents in parallel). Let a *composition operator* be denoted by a function f. That is, $x^S = f(x^{s_i})$ means that S is a composite service and s_i are its (direct) children. For example, $x^C = f(x^a, x^b)$ is a composition operator of a service C, which is composed of service a and b, as shown in Figure 3. A composition operator specifies how quality is composed in a composition. Depending on the type of interactions and quality, composition operators can be defined differently.

Table III show some examples how some quality metrics are composed in these types of interactions. Let us briefly discuss five *composition operators*:

—SWITCH chooses exactly one among all children. The SWITCH operator simulates the composite quality inherits from one of children. SWITCH chooses a child based on a predefined multinomial distribution. This corresponds broadly to the *case* interaction type.

—MAX composes quality by inheriting from the child with the highest quality value. This relates to *latency* for *flow*.

—MIN composes quality by inheriting from the child with the lowest quality. This relates to *response time* for *flow*.

—SUM yields the composite quality value as the sum of the quality values obtained from all children. This relates to *response time* for *flow*.

—PRODUCT yields the composite quality value as the product of the quality values obtained from all children. This relates to *failure* (which we can think of as the inverse of *availability*) for *flow*.

Note that our approach is not limited to the above operators.

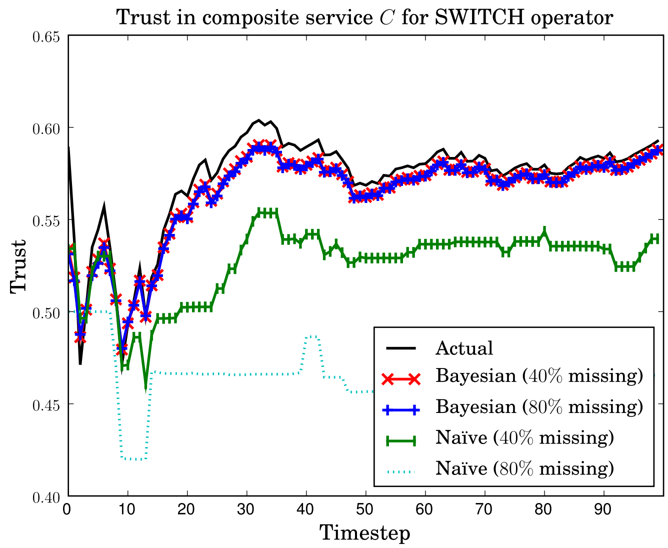

Fig. 4. Trust estimation of composite service C for the SWITCH operator

In following experiments, we consider a basic scenario as shown in Figure 3, where f can be SWITCH, MAX, MIN, SUM, or PRODUCT. In each experiment, the underlying services a and b are first initialized to separate beta distributions. At each time-step, the quality values of a and b are sampled based on these distributions. Then the composite quality is calculated by the composition operator f.

4.1 Bayesian Approach Evaluation

To evaluate our Bayesian approach, we follow the setting described in Section 4 and initialize the hyperparameters (α, β) of the underlying services a and b as $(10, 5)$ and $(2, 8)$, respectively. (Table I shows an example observation.) Service a on average offers better quality than service b. For the SWITCH operator, the probabilities of choosing service a and b are 0.8 and 0.2, respectively. There are total 100 observations. The Bayesian approach goes through the partial observations in order and learns the quality and dependencies of all services online.

4.1.1 *Comparison: Naïve Approach.* We introduce a naïve approach for comparison. The naïve approach is the same as the Bayesian approach except that it does not use EM algorithm. Consequently, it lacks the ability of dealing with missing observations. With the naïve approach, the (conditional) trustworthiness cannot be learned if quality is not observed. Although the composite quality is always observable, the naïve approach still fails to learn it because the composite trust is marginalized from conditional trust. We show how the naïve approach suffers with missing observations.

4.1.2 *Experimental Results.* Figure 4 shows that the Bayesian approach outperforms the naïve approach for the SWITCH operator. The Bayesian approach estimates the trustworthiness well regardless of the amount of missing observations. In contrast, the accuracy of the naïve approach, which is low for 40% missing observations and quite low for 80% missing observations. Similar results are shown in Figure 5 and 6 for conditional trust. Figure 7 shows the average errors of the observations for all composition operators with 40% and 80% missing observations using the Bayesian and naïve approaches.

Now we evaluate how the Bayesian approach identifies the underlying services' influence on the composition based on conditional trust. In order to highlight the difference, we choose different hyperparameters of

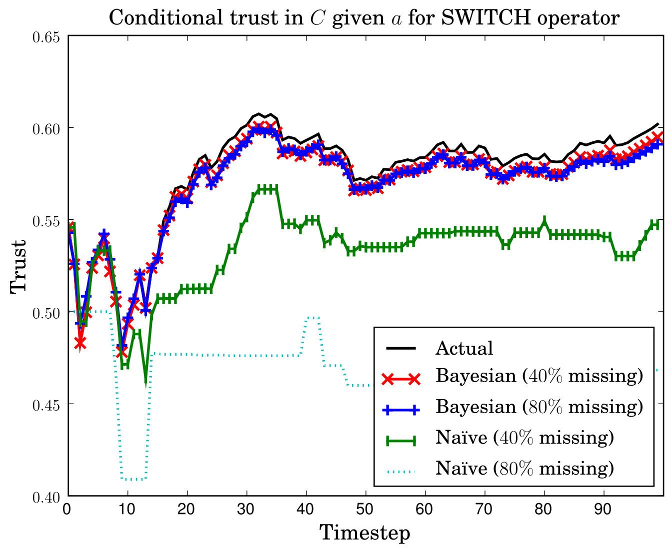

Fig. 5. Conditional trust estimation of composite service C and good service a for the SWITCH operator

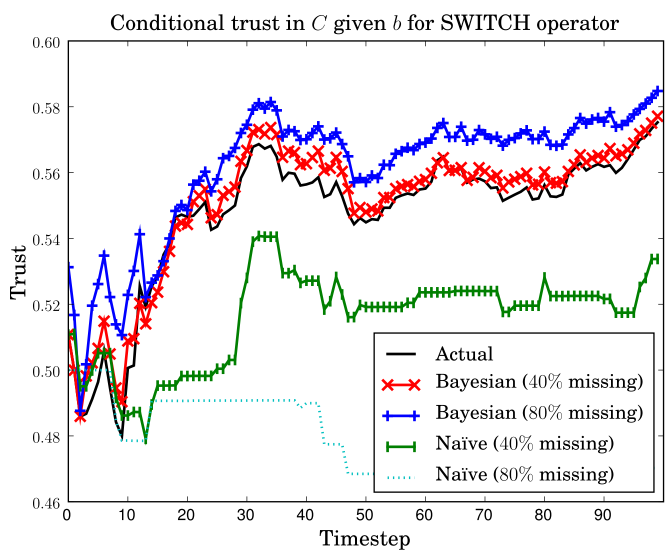

Fig. 6. Conditional trust estimation of composite service C and bad service b for the SWITCH operator

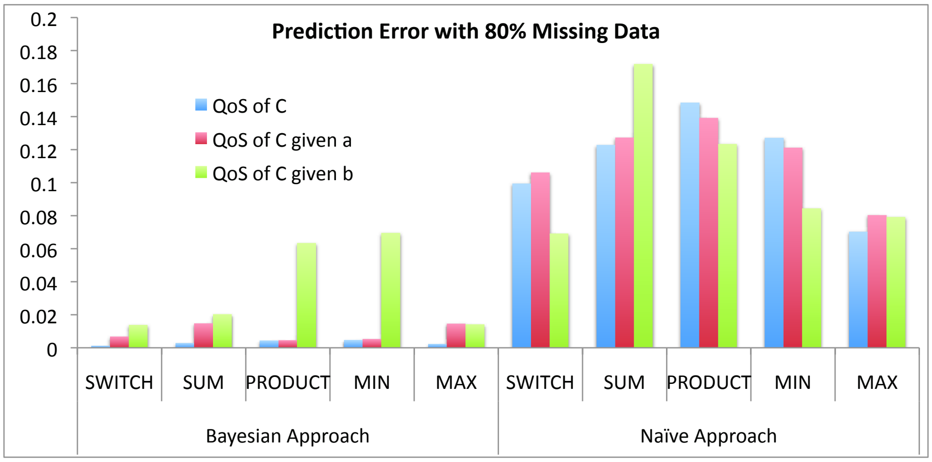

Fig. 7. Prediction errors of Bayesian and Naïve approach with 80% missing data for all composition operators

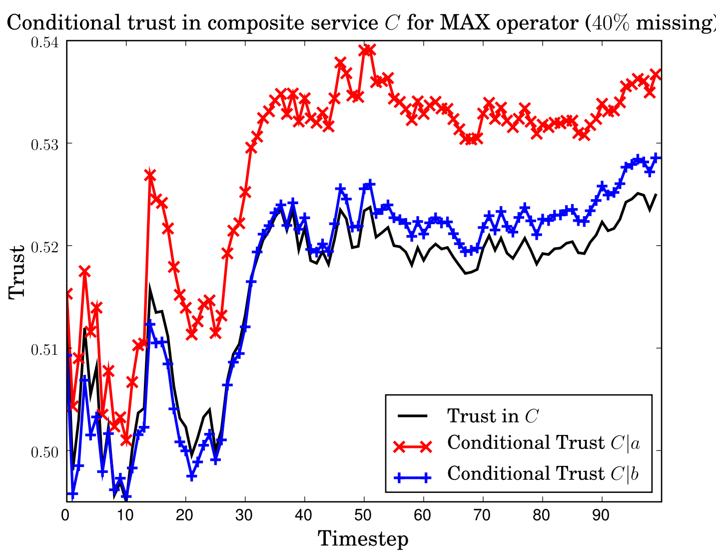

Fig. 8. Conditional trust in composite service C for the MAX operator

services a and b than in the above; specifically, we a's hyperparameters to $(10, 10)$ and b's to $(6, 8)$. Figure 8 compares the conditional trust in C given a and b with overall trust in C for the MAX operator. In 77% of the observations, service a yields better performance than service b. In other words, in the MAX composition, 77% of the composite quality comes from service a. The conditional trust in C given a means if service a performs well, what the probability of service C performs well is. We know when a performs well, the MAX operator tends to select a more. Therefore, the conditional trust in C given a is much higher than overall trust in C. In contrast, since the MAX operator mostly selects a, the conditional trust in C given b is extremely close to the overall trust in C. However, those 23% observations that come from b make the conditional trust in C given b higher than overall trust in C.

As we would expect, the MIN operator selects b 77% of the time. As Figure 9 shows, the conditional trust

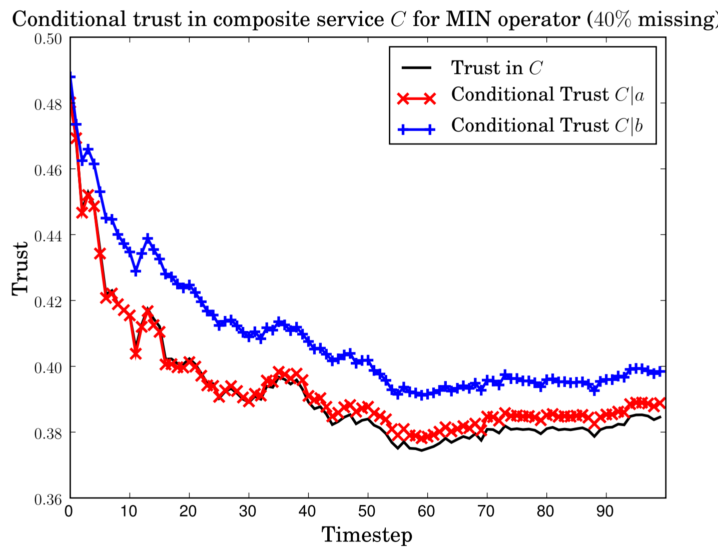

Fig. 9. Conditional trust in composite service C for the MIN operator

placed in C given b is much higher than the conditional trust in C given a and the overall trust in C. The conditional trust in C given a is slightly higher than overall trust in C because of those 23% observations from a.

4.2 Dealing with Dynamic Behavior

This experiment examines the Bayesian approach's ability of tracking the dynamic behavior of services. We introduce two dynamic behavior profiles:

—The *random walk* profile models the general predictable behavior of a service. The random walk service changes behavior every period. Its current behavior x^t depends on the previous behavior x^{t-1}, and is defined as $x^t = x^{t-1} + \psi U(-1, 1)$, where ψ is a real number between 0 and 1, and $U(-1, 1)$ represents the uniform distribution from -1 to 1. In our setting, the random walk service changes behavior every ten time-steps, and $\psi = 0.8$.

—The *cheating* profile models a service that turns bad once its reputation is built. Its behavior is defined as $x^t = 1$ when $t \leq d/2$, and $x^t = 0$ otherwise, where d is the total number of observations. We set the discount factor $\gamma = 0.6$. The total number of observations is 100.

We follow the setting from Section 4 but replace underlying service b with a random-walk service. Figure 10 shows how our trust values predict the actual behavior of the random walk service with 0%, 20%, and 40% missing data. The result shows that our approach captures the dynamism of the random walk service, although the missing data does slow down the convergence noticeably. Figure 11 shows a similar result of tracking a cheating service.

4.2.1 *Summary of Bayesian Approach.* The above experiments show our Bayesian approach can (1) model the relationships of the service composition; (2) distinguish the good and bad services in partial observable setting; and (3) extract the conditional probabilities in the relationships. In the second simulation, our approach tracks the *random walk* and *cheating* underlying services adaptively.

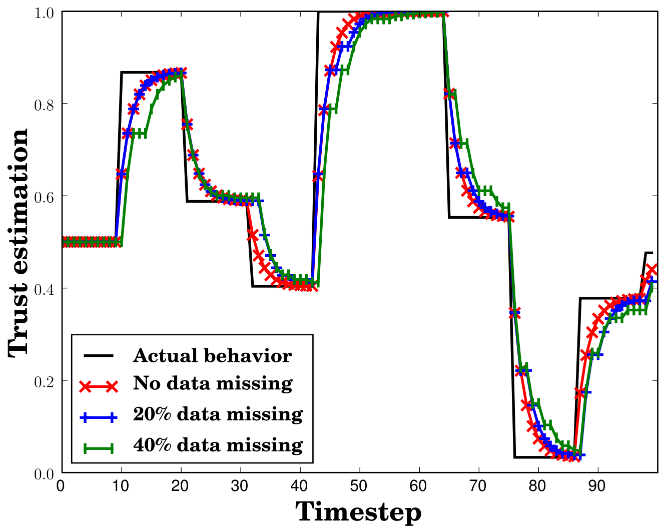

Fig. 10. Trust in a random-walk underlying service

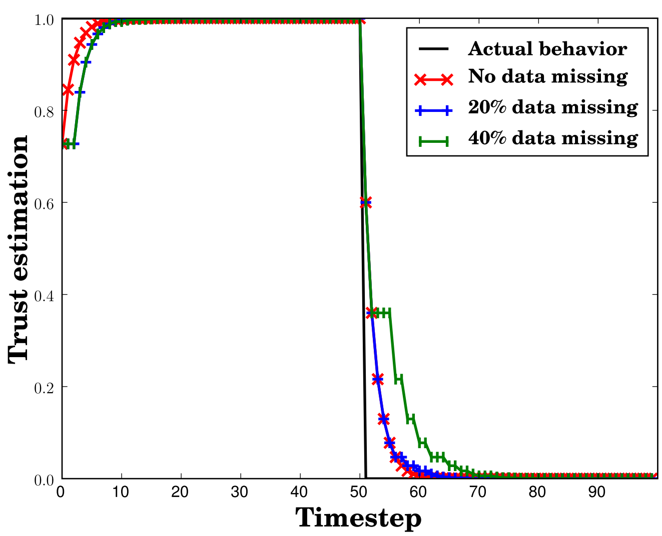

Fig. 11. Trust in a cheating underlying service

	Actual			FCM-MM			Beta-Mixture		
	α	β	π	α	β	π	α	β	π
SWITCH	20	20	.45	17.46	18.31	.52	19.87	20.37	.5
	2	9	.55	3.24	19.36	.48	2.39	13.46	.5
(K-S test)			.90			.99			.99
SUM	20	20		16.42	5.59	.41	17.33	6.22	.31
	1	6		42.04	32.08	.59	44.49	29.63	.69
(K-S test)			N/A			.31			.40
PRODUCT	20	20		10.43	65.45	.64	10.43	65.45	.43
	5	9		15.44	45.91	.36	12.50	48.85	.57
(K-S test)			N/A			.25			.65
MIN	3	4		5.82	15.32	.54	4.42	6.52	.96
	5	4		17.01	15.28	.46	5.38	69.52	.04
(K-S test)			N/A			.54			.97
MAX	3	4		13.47	18.29	.39	107.09	295.96	.06
	4	4		28.47	15.05	.61	9.98	7.13	.94
(K-S test)			N/A			.61			.91

Table IV. Actual and estimated parameters by FCM-MM and Beta-Mixture, and their K-S test goodness-of-fit measurements

4.3 Beta-Mixture Approach Evaluation

Following the setting described in Section 4, we apply our beta-mixture approach to model the composite distribution with different types of composition operators. The number of mixture components is known to be two because C has two direct children a and b. For each experiment we sample 100 observations. Note that in this experiment, the quality of the underlying services a and b is totally unobservable. The only information from which the beta-mixture approach can learn is the composite quality.

4.3.1 *Comparison: FCM-MM Approach.* To enable comparison, we introduce the *FCM-MM* approach. This approach uses *Fuzzy C-Means Clustering* (or FCM) [Bezdek 1981] to partition the observations into two clusters. The portion of each cluster is calculated as our π. Then the *Method of Moments* (MM) [Fielitz and Myers 1975] is adopted to estimate the beta parameters α and β of each component based on clustered observations.

4.3.2 *Evaluation Measurement.* Here we introduce Kolmogorov-Smirnov test (or K-S test) for goodness-of-fit measurement. If the *p-value* from K-S test is higher, the distribution explains the data better. In general, a p-value higher than 0.05 is considered a good fit. Figure 12 shows the comparison of our beta-mixture approach and the FCM-MM approach.

4.3.3 *Experimental Results.* Table IV summarizes the results for all composition operators using the beta-mixture and FCM-MM approaches. Figure 13, 14, 15, 16, and 17 show the actual quality histogram and the learned distribution for the SWITCH, SUM, PRODUCT, MIN, and MAX composition operators, respectively.

Since the SWITCH operator follows the setting of a mixture model (that is, each observation comes from one of the components with a probability), the beta-mixture approach performs quite well in this case, yielding a p-value close to one. The parameters of each component distribution and the mixing coefficients are estimated accurately.

For SUM and PRODUCT composition, the beta-mixture approach approximates the composite distribution fairly well with solid p-values, but yields inaccurate parameter estimation of the underlying components. Note that the observation histograms from the SUM and PRODUCT operators in Figure 14 and 15 tend to follow a unimodal distribution, that is, one only one maximum. In this case, it is harder to estimate the parameters from each component. However, if the quality from the underlying services can be partially observed, the accuracy of the component parameters can be improved. This is left as future work.

The MIN and MAX operators are similar to the SWITCH operator in the sense that the composite quality

Fig. 12. Kolmogorov-Smirnov test comparison for FCM-MM and Beta-Mixture

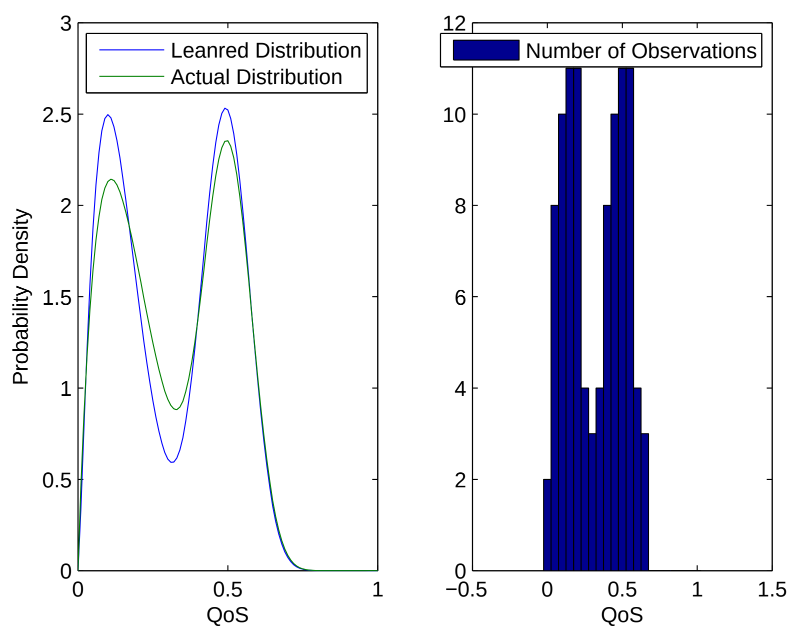

Fig. 13. Estimated beta mixture and actual distribution and samples of trust in quality for a SWITCH composition

inherits from one of the components, except that the mixing coefficients are unknown. The p-values show the beta-mixture approach is still highly promising in estimating the composite quality distribution. Note that, different from the SWITCH operator, the MIN and MAX operators tend to yield a dominant component, whose corresponding distribution has highest or lowest means. In this case, the mixing coefficient of that component is close to one, making the remainder of the mixing coefficients extremely small. In other words, the distributions of these weaker components are not learned well because of the lack of evidence. For example, in Table IV, the second component in the MIN case and the first component in the MAX case are

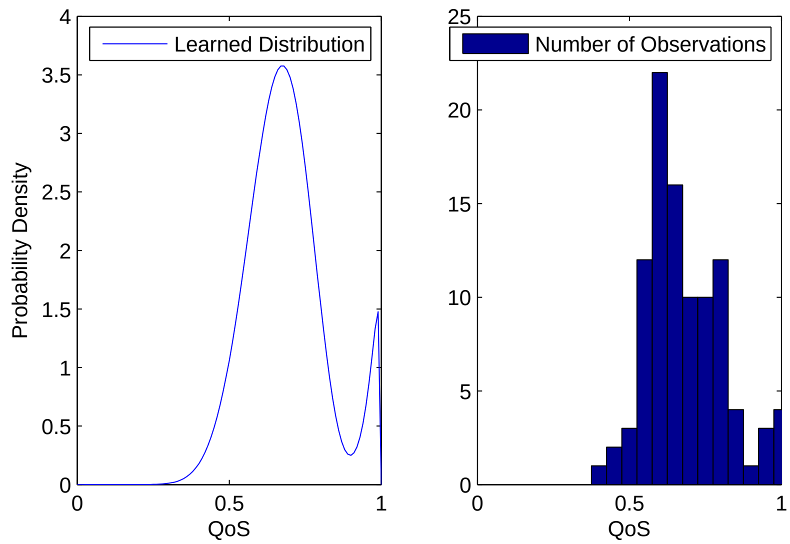

Fig. 14. Estimated beta mixture and actual distribution of trust in quality for a SUM composition

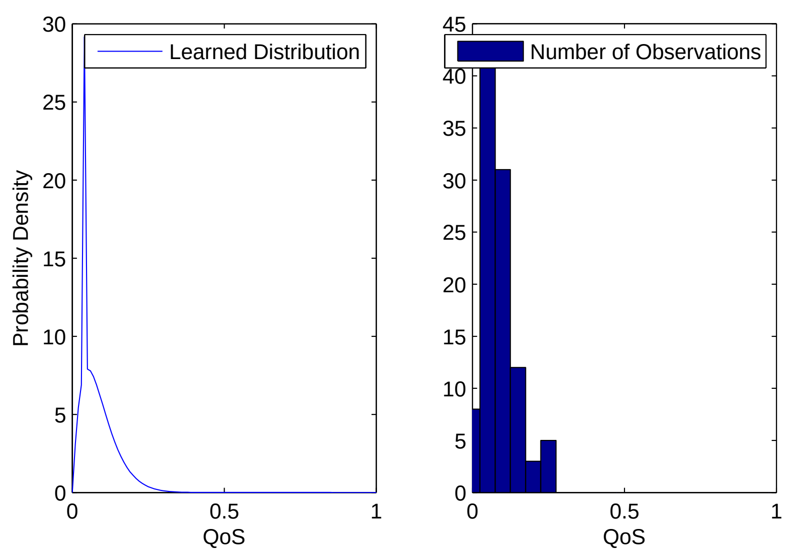

Fig. 15. Estimated beta mixture and actual distribution of trust in quality for a PRODUCT composition

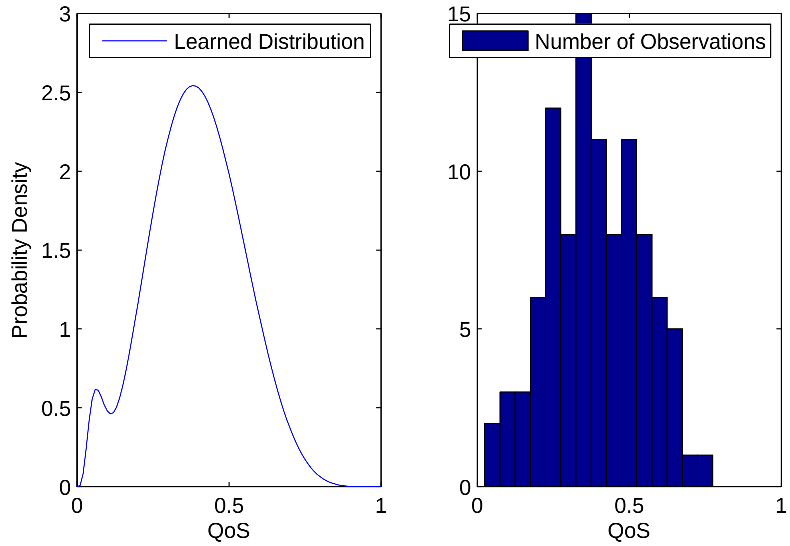

Fig. 16. Estimated beta mixture and actual distribution of trust in quality for a MIN composition

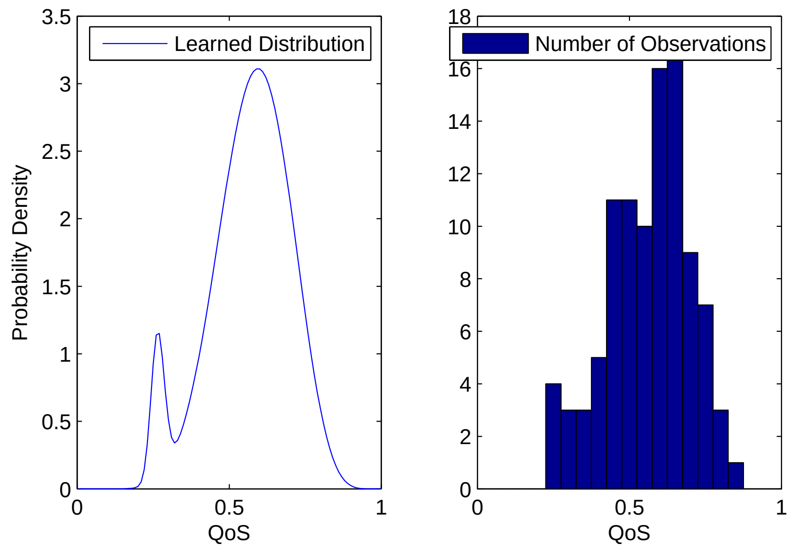

Fig. 17. Estimated beta mixture and actual distribution of trust in quality for a MAX composition

dominated. Their corresponding α and β are not accurate. However, the beta-mixture approach can still distinguish the strong components from weak ones by estimated mixing coefficients. This enables us to know which of the underlying services are better than others.

4.3.4 *Summary of the Beta-Mixture Approach.* Our experiments show the beta-mixture approach provides a powerful way to estimate the quality distribution of composite services without knowing the underlying quality. It also accurately estimates the "responsibilities" of each underlying service in contributing the overall composite quality. However, sometimes the beta-mixture approach has two drawbacks when learning the parameters of the underlying services. First, when the composite distribution is unimodal, it is difficult to learn the component distributions. The accuracy in this case may be improved if the observations of the underlying services are partially observable. Second, the underlying services that are rarely contributing are difficult to learn because of lack of evidence, but the beta-mixture can correctly identify those services.

5. CONCLUSIONS AND FUTURE WORK

This paper presents two probabilistic approaches for trust-aware service selection that accommodates service composition. The approaches capture the relationships between the qualities offered by a composite service and the qualities offered by it constituent services. The trust information can be integrated with our previous trust model; is learned sequentially from the directed observations; and further combined with indirect evidence in terms of service qualities. Our approaches can deal with incomplete observations, arising from the fact that the services underlying a composed service may not be observable. Each consumer maintains its own knowledge of the environment locally and monitors the quality metrics of the parties with whom it is interacting. Our model rewards services with good quality values and punishes those with bad quality values. This paper shows how to model the relationship between service qualities and important service composition operators.

Our approach is able to accommodate a variety of service composition operators in a uniform manner, thus covering the situations that arise in scientific and business applications. Our neutral with respect to the specific qualities considered as long as they can be measured. In particular, it would apply to subjective qualities such as the quality of user experience or system-level qualities such as privacy preservation of user data. We would define the appropriate mixtures for the composition operators with respect to such qualities, and then our approach would apply equally well.

This work suggests important directions for future work. In particular, we expect to study situations where the inherent nature of the composition operators in consideration has the effect of hiding or diminishing the information about the constituent services. Section 4.3.3 discusses this situation. We will address this challenge by considering multiple service compositions, each potentially involving different but overlapping sets of constituent services, thereby acquiring further information about additional constituent services, even if they cannot be readily observed directly.

Another direction of interest is to apply *Structural EM* [Friedman 1998] instead of parameter estimation, which would learn not only the trust information but also the graph structure. The learned structure can be used as a basis for suggesting new service compositions.

A third direction of interest is to expand the above methods to deal with situations where the consumers participate in a social network wherein they may exchange referrals and ratings about services. This indirect evidence can be aggregated with the trust information, thus helping consumers discover strangers and identify desired services more quickly than otherwise.

Acknowledgments

This work was partially supported by the U.S. Army Research Office (ARO) under grant W911NF-08-1-0105 managed by the NCSU Secure Open Systems Initiative (SOSI). We thank Carla Savage for helpful comments on an earlier version.

REFERENCES

1995. –.

AMAZON.COM. 2009. Amazon elastic compute cloud (Amazon EC2) http://aws.amazon.com/ec2/http://aws.amazon.com/ec2/.

BEZDEK, J. C. 1981. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA.

BISHOP, C. M. 2006. *Pattern Recognition and Machine Learning*. Springer, New York, NY, USA.

BOUGUILA, N., ZIOU, D., AND MONGA, E. 2006. Practical Bayesian estimation of a finite beta mixture through Gibbs sampling and its applications. *Statistics and Computing 16,* 2, 215–225.

BPEL. 2007. Web services business process execution language, version 2.0. http://docs.oasis-open.org/wsbpel/2.0/.

BUNTINE, W. L. 1994. Operations for learning with graphical models. *Journal of Artificial Intelligence Research 2*, 159–225.

DEMPSTER, A. P., LAIRD, N. M., AND RUBIN, D. B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological) 39,* 1, 1–38.

EVANS, M., HASTINGS, N., AND PEACOCK, B. 2000. *Statistical Distributions*, 3rd ed. Wiley-Interscience, New York, NY, USA.

FIELITZ, B. D. AND MYERS, B. L. 1975. Estimation of parameters in the beta distribution. *Decision Sciences 6,* 1, 1–13.

FRIEDMAN, N. 1998. The Bayesian structural EM algorithm. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI)*. Morgan Kaufmann, San Francisco, CA, USA, 129–138.

HANG, C.-W., WANG, Y., AND SINGH, M. P. 2008. An adaptive probabilistic trust model and its evaluation. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems*. IFAAMAS, Columbia, SC, 1485–1488.

HANG, C.-W., WANG, Y., AND SINGH, M. P. 2009. Operators for propagating trust and their evaluation in social networks. In *Proceedings of the 8th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. IFAAMAS, Columbia, SC, 1025–1032.

LAURITZEN, S. L. 1995. The EM algorithm for graphical association models with missing data. *Computational Statistics & Data Analysis 19,* 2, 191–201.

LIN, W.-L., LO, C.-C., CHAO, K.-M., AND YOUNAS, M. 2008. Consumer-centric QoS-aware selection of web services. *Journal of Computer and System Sciences 74,* 2, 211–231.

LIU, W. 2005. Trustworthy service selection and composition—reducing the entropy of service-oriented web. In *Proceedings of the 3rd IEEE International Conference on Industrial Informatics (INDIN)*. IEEE Computer Society, Los Alamitos, CA, USA, 104–109.

MAXIMILIEN, E. M. AND SINGH, M. P. 2004. A framework and ontology for dynamic web services selection. *IEEE Internet Computing 8,* 5 (Sept.), 84–93.

MCLACHLAN, G. AND PEEL, D. 2000. *Finite Mixture Models*. Wiley-Interscience Wiley-Interscience Wiley-Interscience, New York, NY, USA.

MENASCÉ, D. A. 2004. Composing web services: A QoS view. *IEEE Internet Computing 8,* 6, 88–90.

MILANOVIC, N. AND MALEK, M. 2004. Current solutions for web service composition. *IEEE Internet Computing 8,* 6, 51–59.

PARADESI, S., DOSHI, P., AND SWAIKA, S. 2009. Integrating behavioral trust in web service compositions. In *Proceedings of the 7th IEEE International Conference on Web Services (ICWS)*. IEEE Computer Society, Los Alamitos Los Alamitos Los Alamitos, CA, USA.

PEARL, J. 1988. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

SINGH, M. 1997. Learning Bayesian networks from incomplete data. In *Proceedings of the 14th National Conference on Artificial Intelligence (AAAI)*. AAAI Press, Menlo Park, CA, USA, 534–539.

SINGH, M. P. AND HUHNS, M. N. 2005. *Service-Oriented Computing: Semantics, Processes, Agents*. John Wiley & Sons, Chichester, UK.

WANG, Y. AND SINGH, M. P. 2006. Trust representation and aggregation in a distributed agent system. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI)*. AAAI Press, Menlo Park, 1425–1430.

WANG, Y. AND SINGH, M. P. 2007. Formal trust model for multiagent systems. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*. IJCAI, Detroit, 1551–1556.

WU, G., WEI, J., QIAO, X., AND LI, L. 2007. A Bayesian network based QoS assessment model for web services. In *IEEE International Conference on Services Computing*. IEEE Computer Society, Los Alamitos, CA, USA, 498–505.

YUE, K., LIU, W., AND LI, W. 2007. Towards web services composition based on the mining and reasoning of their causal relationships. In *Advances in Data and Web Management*. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, Berlin, 777–784.

ZHANG, N. L. AND POOLE, D. 1996. Exploiting causal independence in Bayesian network inference. *Journal of Artificial Intelligence Research 5*, 301–328.