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 directly or by providing additional services. Existing approaches for service selection either fail to capture the dynamic relationships between services or assume that the environment is 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 based on Bayesian networks and the other 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—*Web-based services*; I.2.11 [**Computing Methodologies**]: Distributed Artificial Intelligence—*Intelligent agents*

General Terms: Algorithms, Experimentation

Additional Key Words and Phrases: Trust, probabilistic modeling, service-oriented computing

ACM Reference Format:
Hang, C.-W. and Singh, M. P. 2011. Trustworthy service selection and composition. ACM Trans. Auton. Adapt. Syst. 6, 1, Article 5 (February 2011), 17 pages.
DOI = 10.1145/1921641.1921646 http://doi.acm.org/10.1145/1921641.1921646

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 2009]. In many domains, often multiple services provide similar functional properties. For example, several practical services, offered by airlines and travel agencies, provide airline tickets. Therefore, distinguishing and selecting services with the desired nonfunctional characteristics becomes essential, both for direct interaction and for specifying composite services. We address the problem of selecting services based on criteria such as user requirements and service qualities.

A preliminary version of this manuscript, "Selecting trustworthy services: Learning a Bayesian network," was presented at the AAMAS 2009 Workshop on Trust in Agent Societies and appears in the unpublished workshop notes. This submission incorporates substantial revisions and extensions to the formal models, techniques, and results.
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).
Authors' address: C.-W. Hang and M. P. Singh, Department of Computer Science, North Carolina State University, Raleigh, NC 27695-8206; email: chang@ncsu.edu, singh@ncsu.edu.

DOI 10.1145/1921641.1921646 http://doi.acm.org/10.1145/1921641.1921646

Traditional SOC approaches address only service discovery, not service selection as such. Specifically, *Web Service Definition Language (WSDL)* helps describe the functionality (that is, the methods) a service supports, 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 service 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 service instances.

Recent research on trust modeling 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 service of the desired functionality and quality. 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.

Estimating trust from direct experience with a service is not straightforward, because some services may not directly expose details of their composition to their consumers. A consumer may interact with a composite service without knowing much about the qualities of the services that underlie it. In such a case, evaluating the trustworthiness of a service is nontrivial. For example, a consumer may book an itinerary at a travel agency which may use underlying services for flights, hotels, and ground transportation. Suppose the consumer is not satisfied with the composite service because of its late response time. The service selection should penalize the composite service, as well as some or all of the constituent ones. If the hotel service, for instance, is determined to be the cause of an unsatisfactory quality value, the service selection should reflect the changes in the way that consumers or other composite 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.

Thus, service selection should consider service compositions to model how a quality of a component service can affect the whole composition. For example, the reliability of a composite travel service may be affected by the reliability of the constituent hotel and flight services. If a constituent service is not reliable, the composite service is possibly not reliable either. Thus, the composition model should represent not only the relationships between services, but also any dependencies between them. Of course, it may turn out that the qualities of constituent services do not influence the composite service. For example, a composite service may be constructed so that its reliability may not correspond trivially to the trustworthiness of that constituent service. This would make it difficult to collect and evaluate information about service qualities, thereby complicating service selection in the presence of service compositions.

Contributions

This article provides a trust-aware service selection approach that addresses the previously stated challenges, and which supports the following operations.

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

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

This article presents a formal service selection model in probabilistic terms. It develops approaches applying which a consumer may monitor and explore desired service compositions. This article shows how our approach dynamically rewards and punishes the services involved despite incomplete knowledge of the composition. An important contribution is in systematically treating the relationships between some key service composition operators and different types of service qualities.

In this manner, this article addresses 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. For example, suppose service A invokes service B, which may invoke C and D with probabilities p_c and p_d, respectively. Menascé's approach requires knowing the invocation probabilities of the constituent services. But 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 applies Bayesian networks and probability mixtures 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 the 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.

Yue et al. [2007] apply Bayesian networks 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 they can produce service composition guidance 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 subsequently affects the trustworthiness of a service.

Liu [2005] views a service-oriented environment as an ecosystem and distinguishes 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 suggests that an ideal trust representation be: (a) flexible and adaptive to 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 proposes ant crawling for consumers to discover semantically similar services and further classify these into clusters, thus generating a new composition plan. Finally, at the emergent behavior level, the topology of the environment (defined via referrals) evolves based on low-level interactions between agents. 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. We implement and evaluate our approach via simulations.

Paradesi et al. [2009] build a trust framework for Web service compositions. They adopt the trust representation from Wang and Singh [2010] and introduce 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.

Maximilien and Singh [2004] develop a trust-aware approach to select services based on a well-defined ontology that provides a basis for describing consumers' requirements and providers' advertisements. The ontology enables consumers to define nonfunctional properties. Their approach does not consider service composition whereas ours does. We model service compositions to deal with partial observability of the services behind compositions.

Wang and Singh [2010] develop a trust model for multiagent systems that formalizes how how agents map evidence to trust and vice versa. Our work studies how to estimate the trustworthiness of services that are constituents of a composite service. We develop a systematic way to represent qualities of service compositions via operators that correspond to typical ways of composing services. Our contribution is a way to handle service selection in the context of service compositions.

3. TRUST-AWARE SERVICE SELECTION MODEL

We represent trust based on the *beta probability distribution* [Evans et al. 2000], which can be integrated with Wang and Singh's model [2006, 2010]. 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. Previously, Hang et al. [2009] show how to model trust from indirect experience. This is beyond our present scope.

Figure 1 shows our approach schematically. Several services exist in the computational environment. Each consumer maintains its own local model to guide itself to reward or penalize services based on its direct interactions with them. 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 with respect to the

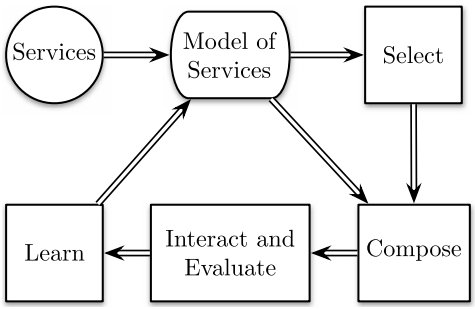


Fig. 1. Our trustworthy service selection approach, schematically.

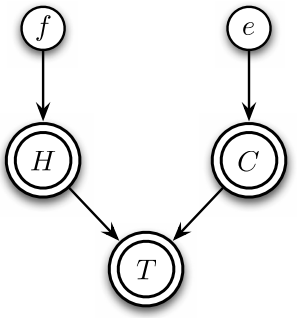
service qualities of interest. Based on the evaluation, the consumer applies a learning method to update the model it is maintaining for the services. In an alternative scenario, the consumer may not be responsible for composing services and would simply select an atomic service or a composite service that another party has composed. In this case, it would need to learn about the services from less information than in the first scenario. Our approach handles both of these scenarios.

We introduce two service selection approaches which can construct models from *incomplete* observations (direct experience) of a consumer. 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 $D^t = (x_1^t, x_2^t, \ldots, x_d^t)$, where d is the number of services in the composition.

Section 3.1 presents our *Bayesian* approach which models service compositions via Bayesian networks in partially observable settings. The Bayesian approach captures the dependency of providing a good service quality between the composite and the constituent services. It also adaptively updates trust to reflect the most recent quality. Section 3.2 describes our *beta-mixture* approach. This approach can learn not only the distribution of the composite quality, but also the *responsibility* of a constituent service in the composite quality without actually observing the constituent's performance. These two approaches provide different information about the services. The Bayesian approach uses online learning to track service behavior and shows how the composite service's quality depends upon its constituents' quality. The beta-mixture model learns the quality distribution of the services and provides how much each constituent service contributes to the composition.

3.1 Bayesian Approach

A Bayesian network is a directed acyclic graph $G = \langle \mathbf{V}, \mathbf{R} \rangle$ with random variables $\mathbf{V}$ as nodes, and edges $\mathbf{R}$ as the direct relationships between variables. We denote atomic and composite services with lowercase and uppercase, 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 more typical composition hierarchy where a composite service would be a parent (or ancestor) of its constituent services. We use the Bayesian network terminology in this article.



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

Fig. 2. Service composition example. The trust the consumer places in service T is the marginal probability $P(T)$: that is, the probability of obtaining a satisfactory quality from T. $P(T)$ can be calculated by marginalizing over all the parents of T. In this example, $P(T)$ is computed as shown.

Note that an edge can originate from either an atomic or a composite service but must terminate on a composite service. A conditional probability associated with each node represents trust (a probability) of the node variable given an estimate of its parent's trustworthiness. Let each node in the Bayesian network equal trust, that is, the probability of obtaining a good outcome from the service corresponding to the node. The good outcome in this case depends on a specified quality. An edge represents composition. For example, in Figure 2, the composite hotel service H is composed of the Four Seasons Hotel service f: that is, f is a parent of H. Then the trustworthiness of 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 Rent-A-Car service e.

The conditional probability table associated with each node provides 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. Modern approaches model trust as two values: probability and certainty. In our Bayesian approach, currently we only consider probability values. We defer incorporating certainty, for example, as formalized by Wang and Singh [2010], to future work.

In a fully observable setting, we can estimate the trustworthiness θ_i of service x_i by applying *Maximum Likelihood Estimation* (MLE) to maximize the likelihood of the observations Buntine [1994]. For example, let a consumer have $m_i = 10$ good outcomes and $l_i = 5$ bad outcomes with service x_i. Then the consumer can calculate the trustworthiness θ_i of service x_i as $\frac{m_i}{m_i+l_i} = 2/3$.

To address overfitted results produced by MLE (e.g., when $m_i = 10$, $n_i = 0$; then $\theta_i = 1$), we further apply *Bayesian inference* by introducing a beta distribution $P(\theta_i)$ associated with two *hyperparameters* (α_i, β_i) over trust θ_i as a conjugacy prior [Bishop 2006] to update trust (now a beta distribution) of service x_i. Let the consumer's current trust in service x_i be $(\alpha_i, \beta_i) = (5, 5)$. Suppose the consumer observes three additional outcomes: two good and one bad. The consumer can update the trust estimate by adding the new observations $(2, 1)$ to the previous estimate $(5, 5)$. That is, $(\alpha_i, \beta_i) = (7, 6)$. The consumer can predict that the probability of obtaining a satisfactory quality from the next interaction is $\frac{\alpha_i}{\alpha_i+\beta_i}$. To incorporate the dynamism of service behavior, a discount factor $\gamma \in [0, 1]$ reduces the impact of the old information when we update the trust estimate [Hang et al. 2008]. In other words, in the previous example, we would

obtain a new trust estimate $(\alpha_i, \beta_i) = (5\gamma + 2, 5\gamma + 1)$. The appendix provides additional details of parameter estimation using MLE and Bayesian Inference.

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 optimally estimate parameters [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 apply *exact inference*, for example, *variable elimination* [Zhang and Poole 1996], 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 estimate. We can use $P(x_i^t|D_{observed}, \theta_i^t)$ to estimate the missing *counts* (that is, m_i and l_i). The preceding is the E step of the EM algorithm.

For example, suppose there is a travel service T which includes a 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 nominal 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 using 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 estimation converges [Dempster et al. 1977]. The EM process is in essence a sequential (online) learning method: it can be repeated whenever the consumer makes new observations. The appendix provides an extended example.

3.2 Beta-Mixture Approach

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, and computer vision. 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

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

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 $0 \leq \pi_k \leq 1$. We can understand each mixing coefficient as an indicator of the corresponding component's *responsibility*, indicating how much contribution the component makes toward the composite quality. The mixture distribution is governed by parameters π and Θ which can be estimated by maximizing the log-likelihood function using the EM algorithm. Specifically,

$$L(\Theta) = \ln p(D|\pi\,\Theta) = \sum_{i=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k p(x_i|\theta_k) \right\}. \tag{2}$$

Let us define binary latent random variables z_k, each an indicator of whether an observation is from component k. Exactly one of the z_k equals 1; the rest 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

$$p(D) = \sum_z p(z)p(x|z) = \sum_{k=1}^{K} \pi_k p_k(D|\theta_k). \quad (3)$$

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

$$Q(\Theta, \Theta^{old}) = E_{\Theta^{old}}(L|D) = \sum_z p(z|D, \Theta^{old}) \ln p(D, z|\Theta). \quad (4)$$

In the M step, we first maximize the expectation to determine the new parameter $\Theta^{new} = \arg\max_\Theta Q(\Theta, \Theta^{old})$. Then we verify if the log-likelihood with new parameters has converged; otherwise, we repeat the E and M steps.

Beta-Mixture Model

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

For each composition, we use a beta mixture to model the trust distribution. The number of components is the number of the direct constituent 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 [BPEL 2007]. BPEL defines three types of interactions between Web services, including *sequence*, *case*, and (parallel) *flow*. Let a *composition operator* be denoted by a function f. That is, $x^S = f(x^{s_0}, \ldots, x^{s_k})$ means that S is a composite service and the s_i are its (direct) children.

In our evaluation, we use the following basic scenario wherein a service C is composed of constituent services a and b by applying a composition operator f. We can express how the quality of C is composed from the qualities of a and b as $x^C = f(x^a, x^b)$. Depending on the type of interactions and quality, composition operators can be defined suitably.

Table I shows some examples of how some quality metrics are composed in these types of interactions. Let us briefly discuss five composition operators.

— SWITCH chooses exactly one of its children based on a predefined multinomial distribution. It simulates the composite quality based on one of the children. 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*.

Table I. Composition Operator Examples of Different Qualities and Their Interaction Types

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

— MIN composes quality by inheriting from the child with the lowest quality. This relates to *throughput* for *sequence*.
— SUM yields the composite quality value as the sum of the quality values obtained from all children. This relates to *throughput* 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 preceding operators.

In the following experiments, we consider the basic scenario introduced earlier, wherein we let f be SWITCH, MAX, MIN, SUM, or PRODUCT. In each experiment, the constituent 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 using the appropriate composition operator f.

4.1 Bayesian Approach Evaluation

To evaluate our Bayesian approach, we initialize the hyperparameters (α, β) of the constituent services a and b as $(10, 5)$ and $(2, 8)$, respectively. Thus a on average offers better quality than b. For SWITCH, the probabilities of choosing a and b are 0.8 and 0.2, respectively. There are a total of 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 the purpose of comparison. The naïve approach is the same as the Bayesian approach except that it does not use the EM algorithm. Consequently, it lacks the ability of dealing with missing observations. With the naïve approach, (conditional) trustworthiness cannot be learned if the 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 the conditional trust. We show how the naïve approach suffers in the face of missing observations.

4.1.2 Experimental Results. Figure 3 shows that the Bayesian approach outperforms the naïve approach for SWITCH. The Bayesian approach estimates the trustworthiness well regardless of the amount of missing observations. In contrast, the accuracy of the naïve approach becomes low for 40% missing observations and quite low for 80% missing observations. Similar results (Figure 4) are observed for conditional trust. The accuracy of the naïve approach is significantly reduced by the incomplete observations. The Bayesian approach deals with partial observability better by using the EM algorithm. Figure 5 shows the average errors of the observations for all composition operators with 40% and 80% missing observations using both the Bayesian and the naïve approaches.

Now we evaluate how the Bayesian approach identifies the constituent services' influence on the composition based on conditional trust. In order to highlight the difference, we choose different hyperparameters of services a and b than in the

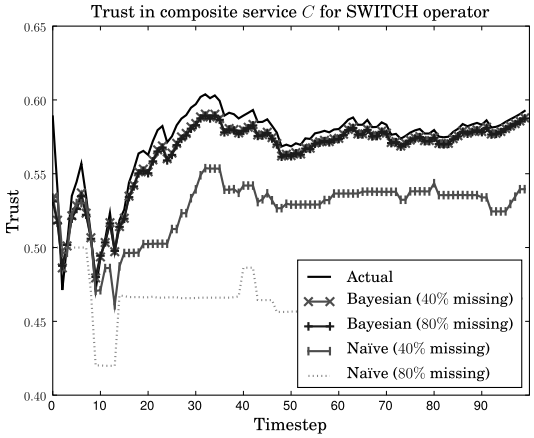


Fig. 3. Trust estimation of composite service C for the SWITCH operator.

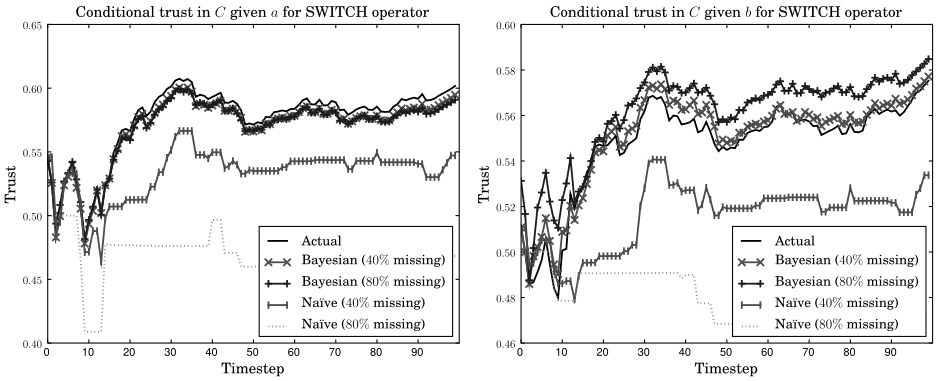


Fig. 4. Bayesian approach: conditional trust estimation of composite service C given good service a (left), and conditional trust estimation of composite service C given good service b (right), for SWITCH.

preceding: specifically, we set a's hyperparameters to $(10, 10)$ and b's to $(6, 8)$. Figure 6 (left) compares the conditional trust in C given a and b with overall trust in C for MAX. We observe that, in 77% of the sampled 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 corresponds to the probability of C performing well given a performs well. We know when a performs well, MAX tends to select a more often. Therefore, the conditional trust in C given a is much higher than the overall trust in C. In contrast, since MAX 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 the overall trust placed in C. As we would expect, MIN selects b 77% of the time. As Figure 6 (right) shows, the conditional trust 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 the overall trust in C because of those 23% observations from a.

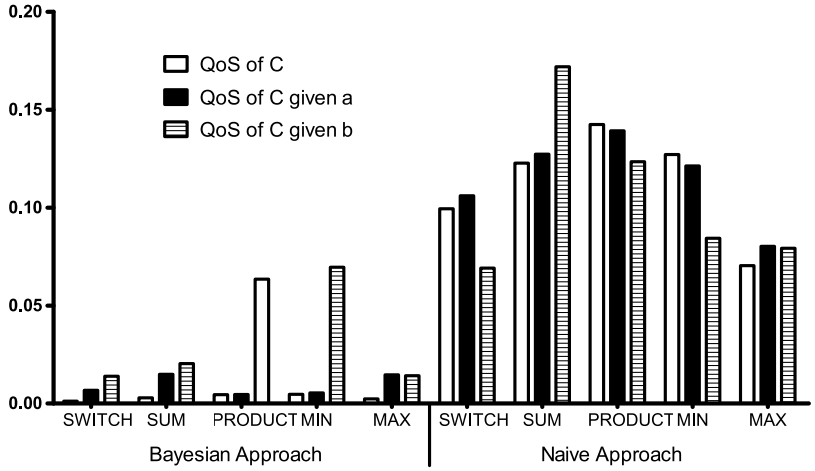


Fig. 5. Prediction errors of Bayesian and naïve approaches with 80% missing data.

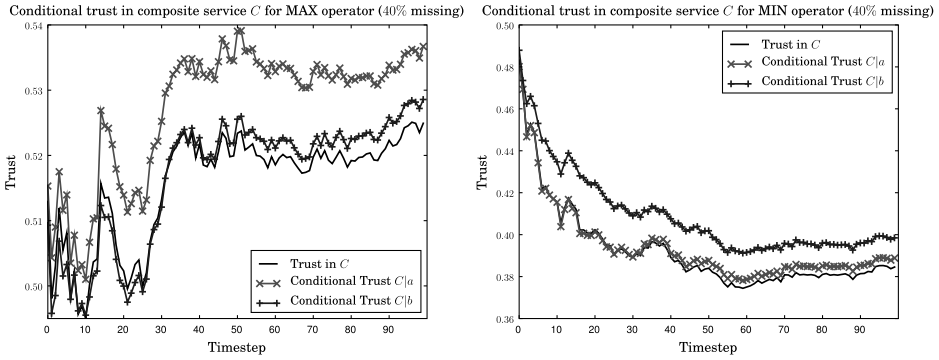


Fig. 6. Conditional trust in composite service C for MAX (left) and MIN (right).

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 more or less 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 up. 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.

Here, we replace the constituent service b with a random walk or a cheating service. Figure 7 shows how our trust values predict the actual behavior of the random walk and the cheating service with 0%, 20%, and 40% missing data. The result shows that

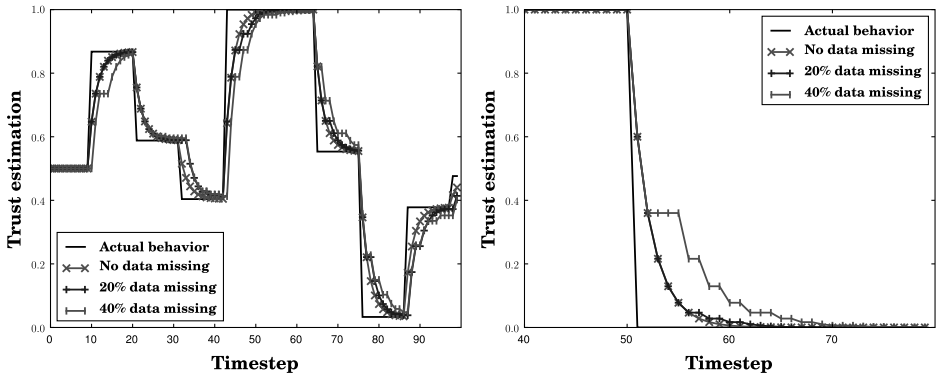


Fig. 7. Trust in a random walk (left) and a cheating (right) constituent service.

our approach captures the dynamism of the constituent services, although the missing data slows down convergence noticeably.

4.2.1 Summary of the Bayesian Approach. The preceding experiments show our Bayesian approach can: (1) model the relationships of the service composition; (2) distinguish good and bad services in a partially observable setting; and (3) extract the conditional probabilities from the relationships. In the second simulation, our approach tracks the random walk and cheating constituent services adaptively. A limitation of the Bayesian approach is that it fails to estimate the (unconditional) trustworthiness of the constituent services. Also, the Bayesian approach requires at least partial observability.

4.3 Beta-Mixture Approach Evaluation

We now apply our beta-mixture approach to model the composite distribution with different composition operators. The number of mixture components is known to be two because C has two children, a and b. For each experiment, we sample 100 observations. Note that in this experiment, the quality of the constituent 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 and Nepal et al. To enable comparison, we first 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 with respect to the total 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.

We also compare our approach with Nepal et al. [2009]. Their approach propagates reputation from a composite service to its constituents. The propagation is based on heuristics, with predefined and fixed mixing coefficients.

4.3.2 Evaluation Measurement. Here we introduce the Kolmogorov-Smirnov test (or K-S test) for goodness-of-fit measurement. The K-S test is used to compare a sample with a reference probability distribution. A *p-value* from the K-S test is calculated by quantifying the distance between the sample and the reference distributions. A higher p-value means the distribution explains the sample better. In general, by convention, a p-value higher than 0.05 is considered a good fit. That is, a p-value lower than the 5% significant level rejects the null hypothesis; the sample is not consistent

Table II. Actual and Estimated Parameters by FCM-MM and Beta-Mixture, and Their K-STest Goodness-of-Fit Measurements

	Actual			FCM-MM			Beta-Mixture		
	α	β	π	α	β	π	α	β	π
SWITCH	20	20	0.45	17.46	18.31	0.52	19.87	20.37	0.50
	2	9	0.55	3.24	19.36	0.48	2.39	13.46	0.50
(K-S test)			0.90			0.99			0.99
SUM	20	20		16.42	5.59	0.41	17.33	6.22	0.31
	1	6		42.04	32.08	0.59	44.49	29.63	0.69
(K-S test)			N/A			0.31			0.40
PRODUCT	20	20		10.43	65.45	0.64	10.43	65.45	0.43
	5	9		15.44	45.91	0.36	12.50	48.85	0.57
(K-S test)			N/A			0.25			0.65
MIN	3	4		5.82	15.32	0.54	4.42	6.52	0.96
	5	4		17.01	15.28	0.46	5.38	69.52	0.04
(K-S test)			N/A			0.54			0.97
MAX	3	4		13.47	18.29	0.39	107.09	295.96	0.06
	4	4		28.47	15.05	0.61	9.98	7.13	0.94
(K-S test)			N/A			0.61			0.91

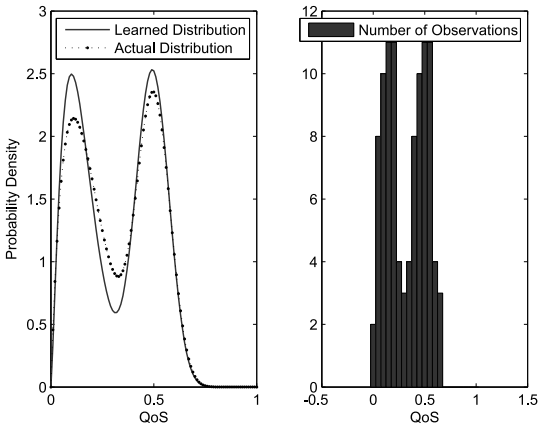


Fig. 8. Beta-mixture approach: estimated beta-mixture and actual distribution and samples of trust in quality for a SWITCH composition. Beta-mixture learns accurate distributions of both component services. One provides good service (left peak); the other provides bad service (right peak).

with the reference distribution. Figure 9 shows the comparison of our beta-mixture approach and the FCM-MM approach.

Because Nepal et al.'s approach is not based on probabilities, we cannot apply the K-S test to compare our approach to theirs. Instead, we compute prediction errors by measuring the absolute difference between the actual and predicted qualities.

4.3.3 Experimental Results. Table II summarizes the results for all composition operators using the beta-mixture and FCM-MM approaches. Figure 8 and the online appendix accessible in the ACM Digital Library provide detailed results for all operators.

Since SWITCH 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.

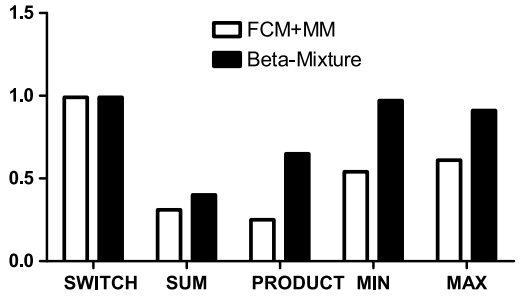


Fig. 9. Kolmogorov-Smirnov test comparison for FCM-MM and beta-mixture.

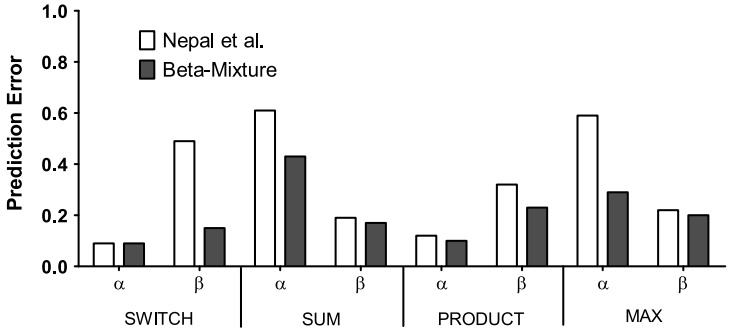


Fig. 10. Prediction error comparison of Nepal et al. and beta-mixture.

For SUM and PRODUCT composition, the beta-mixture approach approximates the composite distribution fairly well with solid p-values, but yields inaccurate parameter estimations of the constituents. The observation histograms from SUM and PRODUCT (shown in the appendix) tend to follow a unimodal distribution. In this case, it is harder to estimate the parameters from each component. However, if the quality from the constituent services can be partially observed, the accuracy of the component parameters can be improved. We defer this to future work.

MIN and MAX are similar to SWITCH in the sense that the composite quality inherits from one of the components, except that the mixing coefficients are unknown. The p-values show that the beta-mixture approach is still highly promising in estimating the composite quality distribution. Note that, unlike SWITCH, MIN and MAX tend to yield a dominant component. 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 II, the second component in the MIN case and the first component in the MAX case are dominated. Their corresponding α and β are not accurate. However, the beta-mixture approach can still distinguish the strong from the weak components via the estimated mixing coefficients which tell us which of the constituent services are better than others.

Figure 10 compares our beta-mixture approach with Nepal et al.'s approach. Our approach yields better trust predictions than Nepal et al. in all cases. The predefined parameters (mixing coefficients and reputations) of Nepal et al. are initialized by FCM-MM. Consequently, once initialized, their mixing coefficients cannot be updated, that is, remain fixed. Although the reputations of constituent services are propagated dynamically, the accuracy is limited by inaccurate mixing coefficients.

4.3.4 Summary of the Beta-Mixture Approach. Our experiments show the beta-mixture approach provides a powerful way of estimating the quality distribution of a composite service without knowing the quality of its constituents. It accurately estimates the *responsibilities* of each constituent service in the overall composite quality. However, the beta-mixture approach has two drawbacks when learning the parameters of the constituent 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 qualities of the constituent services are partially observable. Second, the constituent services that rarely contribute are difficult to learn because of the lack of evidence, although beta-mixture can correctly identify those services. Besides, beta-mixture cannot track dynamic behavior.

5. CONCLUSIONS AND FUTURE WORK

This article 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 its constituents. The trust information is learned sequentially from directed observations and further combined with indirect evidence in terms of service qualities. Our approaches can deal with incomplete observations which arise when the constituent services are not observable. Each consumer maintains its local knowledge of the environment and monitors the quality metrics of the parties with whom it is interacting. We show how to model the relationship between service qualities and important service composition operators. Our approach is evaluated via simulations. We are seeking datasets from a real service-oriented deployment.

Two possible limitations of the Bayesian approach in practical settings are: (1) the lack of unconditional trust in the constituent services, and (2) the assumption of at least partial observability. A possible limitation of beta-mixture is that it cannot deal with dynamic behavior. How these two approaches can be used together to compensate for each other is our future work.

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 approach is 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. An important direction involves situations where the composition operators inherently have the effect of hiding or diminishing the information about the constituent services. Section *4.3.3* discusses this situation. We can 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 preceding methods to deal with situations where the consumers participate in a social network wherein they may exchange referrals and ratings about services. Such indirect evidence can be aggregated with the trust information, thus helping consumers discover strangers and identify desired services more quickly than otherwise.

ELECTRONIC APPENDIX

The electronic appendix for this article can be accessed in the ACM Digital Library.

ACKNOWLEDGMENTS

We thank Carla Savage for helpful comments on an earlier version.

REFERENCES

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

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

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

BOUGUILA, N., ZIOU, D., AND MONGA, E. 2006. Practical Bayesian estimation of a finite beta mixture through Gibbs sampling and its applications. *Statist. Comput. 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. *J. Artif. Intell. Res. 2*, 159–225.

DEMPSTER, A. P., LAIRD, N. M., AND RUBIN, D. B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statist. Soc. Series B 39*, 1, 1–38.

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

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

FRIEDMAN, N. 1998. The Bayesian structural EM algorithm. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI)*. Morgan Kaufmann, San Francisco, CA, 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 Conference on Autonomous Agents and Multiagent Systems*. 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 Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 1025–1032.

LAURITZEN, S. L. 1995. The EM algorithm for graphical association models with missing data. *Comput. Statist. Data Anal. 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. *J. Comput. Syst. Sci. 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, 104–109.

MAXIMILIEN, E. M. AND SINGH, M. P. 2004. A framework and ontology for dynamic Web services selection. *IEEE Internet Comput. 8*, 5, 84–93.

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

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

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

NEPAL, S., MALIK, Z., AND BOUGUETTAYA, A. 2009. Reputation propagation in composite services. In *Proceedings of the 7th IEEE International Conference on Web Services (ICWS)*. IEEE Computer Society, 295–302.

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, CA.

PEARL, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Francisco, CA.

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, 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. 2010. Evidence-Based trust: A mathematical model geared for multiagent systems. *ACM Trans. Auton. Adaptive Syst. 5*, 3.

WU, G., WEI, J., QIAO, X., AND LI, L. 2007. A Bayesian network based QoS assessment model for Web services. In *Proceedings of the IEEE International Conference on Services Computing*. IEEE Computer Society, Los Alamitos, CA, 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, vol. 4505. Springer, Berlin, 777–784.

ZHANG, N. L. AND POOLE, D. 1996. Exploiting causal independence in Bayesian network inference. *J. Artif. Intell. Res. 5*, 301–328.

Received June 2009; revised March 2010; accepted July 2010

Online Appendix to: Trustworthy Service Selection and Composition

CHUNG-WEI HANG and MUNINDAR P. SINGH, North Carolina State University

A. BAYESIAN NETWORKS

We provide some additional technical background in this section.

A.1 Parameter 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

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

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

$$P(D|\theta) = \prod_{t=1}^{n} P(x_1^t, \ldots, x_d^t|\theta), \tag{6}$$

$$= \prod_{t=1}^{n} \prod_{i=1}^{d} \theta_i, \tag{7}$$

$$= \prod_{i=1}^{d} \prod_{x_i, x_{pa_i}} \theta_i^{n(x_i, x_{pa_i})}, \tag{8}$$

$$= \prod_{i=1}^{d} \theta_i^{m_i} (1 - \theta_i)^{l_i}, \tag{9}$$

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, suppose a consumer obtains 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 $l_i = n(x_{pa_i} = 1) - m_i = 15 - 10 = 5$. From these observations, the consumer can calculate that the estimated trustworthiness $\hat{\theta}_i$ is $\frac{10}{15}$. By using MLE, a

DOI 10.1145/1921641.1921646 http://doi.acm.org/10.1145/1921641.1921646

consumer can estimate the trustworthiness of a service from the consumer's observations of it.

A.2 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, Chapter 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 Eq. (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 Eqs. (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.

$$P(x_i|D) \;=\; \int_0^1 P(x_i|\theta_i)P(\theta_i|D)d\theta_i \tag{14}$$

$$\;=\; \int_0^1 \theta_i P(\theta_i|D)d\theta_i \tag{15}$$

$$\;=\; E(\theta_i|D) \tag{16}$$

$$\;=\; \frac{m_i + \alpha_i}{m_i + \alpha_i + l_i + \beta_i} \tag{17}$$

Bayesian inference provides an intuitive way to update the trust (a beta distribution) placed in 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 $\frac{7}{13}$.

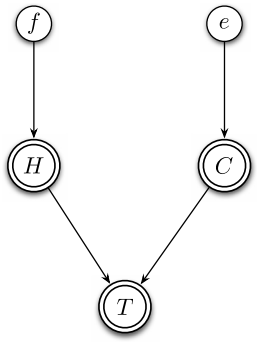


Fig. 11. Service composition example.

Table III. 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

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 Eq. (17) we have

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

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.

B. EXTENDED EXAMPLE FOR BAYESIAN APPROACH

We can implement a sequential approach to construct and learn the service composition model from observations. Taking the scenario of Figure 11 as an example, Table III 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 constituent 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 constituent 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 constituent 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 constituent Enterprise service e.

Table IV shows the parameters estimated using Bayesian inference. The parameters are represented as pairs of hyperparameters α_i, β_i of the corresponding beta distribution. The numbers in the parentheses in Table III are the inferred counts to

Table IV. Parameter Estimation Over Time Based on the Observations of Table III

t	θ^t_f	θ^t_e	$\theta^t_{H\|f=0}$	$\theta^t_{H\|f=1}$	$\theta^t_{C\|e=0}$	$\theta^t_{C\|e=1}$
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)

complete the missing data in the E step. For example, $n(x^2_f = 1) = E(\theta^1_f) = \frac{\alpha^1_f}{\alpha^1_f+\beta^1_f} = 0.67$. Then we can infer $n(x^2_H = 1)$ as follows.

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

Subsequently, we use the completed data to update the parameter estimation. For example, the new estimation θ^2_H (including $\theta^2_{H|f=0}$ and $\theta^2_{H|f=1}$) is given by

$$\begin{aligned} &(\alpha^2_{H|f=1}, \beta^2_{H|f=1}) \\ &= (\alpha^1_{H|f=1} + n(x^2_H = 1, x^2_f = 1), \beta^1_{H|f=1} + n(x^2_H = 0, x^2_f = 1)) \\ &= (2 + P(x^2_H = 1|x^2_f = 1) \times x^2_f, 1 + P(x^2_H = 0|x^2_f = 1) \times x^2_f) \\ &= (2.44, 1.22) \\ &(\alpha^2_{H|f=0}, \beta^2_{H|f=0}) \\ &= (\alpha^1_{H|f=0} + n(x^2_H = 1, x^2_f = 0), \beta^1_{H|f=0} + n(x^2_H = 0, x^2_f = 0)) \\ &= (1 + P(x^2_H = 1|x^2_f = 0) \times (1 - x^2_f), 1 + P(x^2_H = 0|x^2_f = 0) \times (1 - x^2_f)) \\ &= (1.17, 1.17). \end{aligned}$$

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 constituent services may be observed later. For example, $\theta^1_{C|e=0}$ actually means θ^1_C in the first observation because service e is not reported. 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 the new discovery.

C. ADDITIONAL EXPERIMENTAL RESULTS

Here we present some additional experimental results. The explanations for these graphs (Figures 12, 13, 14, 15) follow those given in the main article.

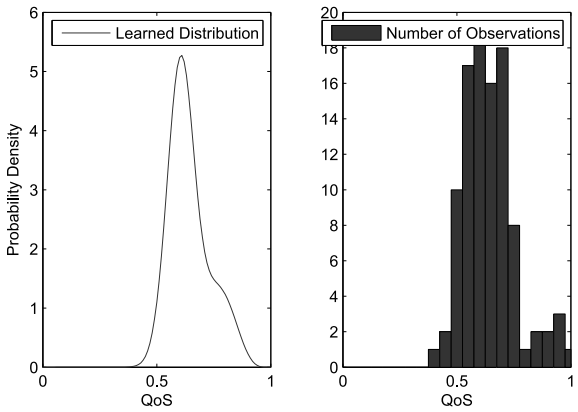


Fig. 12. Beta-mixture approach: estimated beta-mixture and actual distribution of trust in quality for a SUM composition. The composite distribution is learned accurately. However, the beta-mixture approach fails to learn the constituent components well, because the composite histogram tends to follow a unimodal distribution (i.e., only one peak).

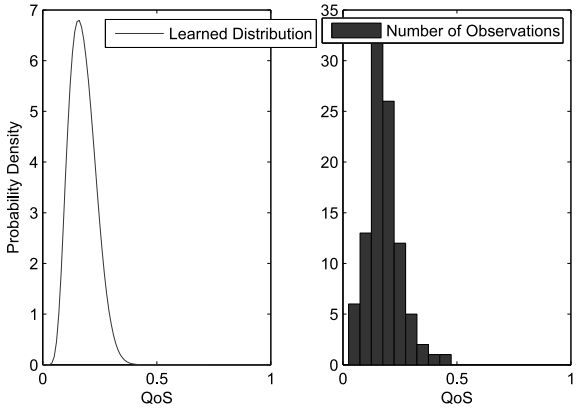


Fig. 13. Beta-mixture approach: estimated beta-mixture and actual distribution of trust in quality for a PRODUCT composition. Similar to Figure 12, the histogram is accurately fit by the composite distribution, but the accuracy of the constituent distributions is hard to learn because of the unimodal observations.

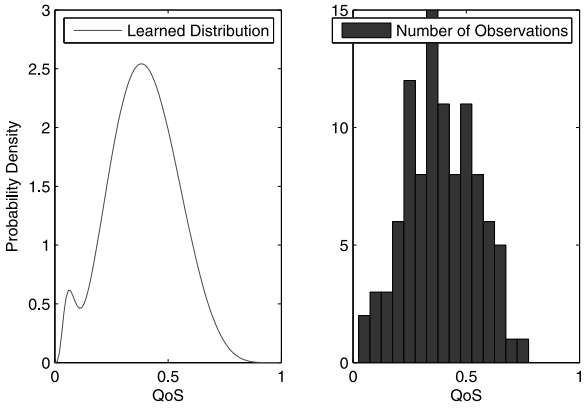


Fig. 14. Beta-mixture approach: estimated beta-mixture and actual distribution of trust in quality for a MIN composition. The histogram is dominated by one constituent component (i.e., one provides good service), which is accurately learned by the beta-mixture approach. Also, beta-mixture accurately estimates the responsibility (dominance) of each component. However, the component other than the dominating one is not accurately learned because of the lack of evidence.

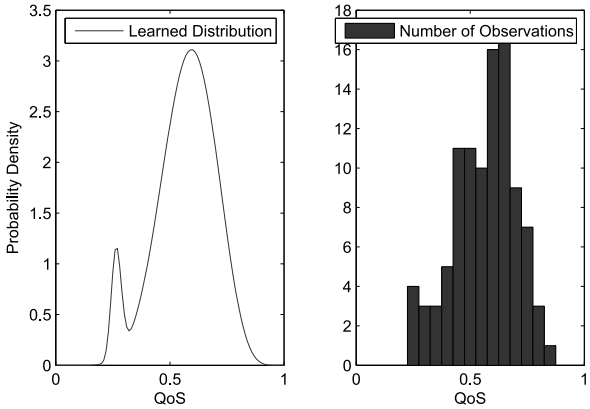


Fig. 15. Beta-mixture approach: estimated beta-mixture and actual distribution of trust in quality for a MAX composition. Similar to Figure 14, the composite distribution, and the dominating constituent component is predicted well, but the lack of evidence affects the accuracy of the minor component. However, beta-mixture learns the responsibility (dominance) well.