Behind the Curtain: Service Selection via Trust in Composite Services

Chung-Wei Hang, Anup K. Kalia, and Munindar P. Singh
Department of Computer Science
North Carolina State University
Raleigh, NC 27695-8206, USA
{chang,akkalia,singh}@ncsu.edu

***Abstract*—Service selection, where some of the services are accessed indirectly as constituents of composite services, is difficult for the following reasons: (1) the interpretation of service qualities is subjective; (2) evidence must be combined from multiple sources; (3) service profiles change dynamically; and (4) constituent services may be only partially observable behind composite services. We propose an approach where we map service qualities to a common probabilistic trust metric. Whereas current trust approaches estimate the trustworthiness of a composite service based on a fully observable and static setting, we propose a statistical approach built on expectation maximized over a finite mixture model. Our experiments show that our approach can dynamically punish or reward the constituents of composite services while making only partial observations.**

***Keywords*-Trust, Reputation, Quality of Service**

I. Introduction

Service selection is a key problem in service-oriented computing. In service-oriented systems, there are a large number of services offering the same functionalities. It is crucial for consumers to select desired services not only by matching functionalities but also by evaluating nonfunctional properties such as quality of service (QoS). Also, services are often accessed indirectly via composite services. In many cases, evaluating the constituents of a composite service helps service selection. For example, if we learn that a constituent service behind a bad composite service is good, then instead of abandoning the whole composite service, we might use the good constituent service in a different composition or directly.

Service selection in the above setting poses significant challenges. *One*, the interpretation of the quality of service (QoS) is subjective in two ways. First, consumers may care about different QoS properties depending on services or consumers' preferences. Second, consumers may have different interpretations of QoS performance depending on their preferences for different needs. For example, one minute can be way too long if you are waiting for a web search result, but not if you are waiting for a bus. Thus, service selection approaches should be able to reflect subjective interpretations from consumers.

Two, the interpretation should be exchangeable and comparable. Consumers may have limited experience with most services—because there are too many services for one to experience or because consumers or services are new to the system. In either case, consumers may want to gather or exchange information with each other. Therefore, the QoS information should be exchangeable and comparable.

Three, QoS should be monitored and tracked dynamically. Another challenge is that the QoS may change over time. For example, a web searching service may yield an unsatisfactory response time during the day because there is higher demand during working hours, but a shopping service may be the opposite because consumers may not do much shopping at work. Also, the frequency of changes in behavior can vary. A service selection approach should be capable of monitoring and tracking the dynamically changing quality provided by the services.

Four, a model of service composition is key. Many existing service selection solutions fail to take the composite service into account [1], or assume the QoS of the constituent services is fully observable [2]. A service composition model is needed to better understand the QoS of the constituent services behind the composition with limited observations. The challenge is that the QoS of the constituent services may not be fully exposed to consumers. For example, when interacting with an itinerary booking service, a service consumer may observe that a flight booking constituent service always responds before the others. But such information may not be observed all the time. Such partial observability makes service selection harder, because consumers may lack information of some of the constituent services. Estimating the QoS of constituent services based on the QoS of the composite is not trivial, and requires a model of service composition.

We define trust as a subjective assessment of QoS expressed probabilistically. Trust modeling in multiagent systems provides a promising solution for the first two challenges: subjectivity and the need for exchangeability. Trust is a subjective interpretation of objective outcomes and reflects personal preferences and requirements. For example, ten minutes can be interpreted as high trust for a bus service but low trust for a web searching service. Besides, trust is comparable and exchangeable. Further, trust can be updated, aggregated, and propagated [3].

Existing trustworthy service selection approaches take advantage of trust modeling to support subjective, exchangeable, and comparable interpretation of QoS, but (a) fail to deal with the dynamic behavior of services, (b) do not take the composite service into account, or assume the constituent services behind the scenes are fully observable.

We overcome these limitations by modeling a composite service as a statistical mixture. Based on the online expectation maximization (EM) algorithm [4], our approach can learn not only the trust to place in the constituent service but also how that trust contributes to the trust in the composite when we do not fully observe the constituent services. We further extend the algorithm by improving the accuracy and the capability of dealing with dynamic services with partial observations of the constituents. Our approach can also adjust the discounting factor to enhance the flexibility of dealing with different levels of dynamism.

The rest of this paper is organized as follows. Section II discusses the most relevant literature. Section III describes our approach. Section IV evaluates proposed approach via simulations. Section V concludes this paper.

II. Literature

Zhang et al. [5] propose an algorithm to detect the accountability of services. They apply entropy-based sensitivity analysis in Bayesian networks, where the nodes are inputs and outputs of the services. Zhang et al. also present an evidence-channel selection algorithm to reduce the number of services that need to be monitored. Their approach needs central agents to monitor accountability, and assumes the environment is fully observable. In contrast, our approach is fully decentralized and assumes partial or even few observations.

Paradesi et al. [6] extend Wang and Singh's trust model [7] to propagate trust through composite services. They discuss four composition types: sequence flow, concurrent flow, conditional flow, and loop, and define a propagation operator. Paradesi et al. show how their operator can be used in those four composition scenarios. However, they employ an ad hoc linear combination for constituent services. This approach ignores the case when the constituent and the composite service perform erratically. By contrast, instead of making distinctions of composition types, we use statistical analysis to explore how trust is composed.

Luo et al. [8] propose an algorithm to select a composite service by choosing the path with the best QoS and the lowest cost. Luo et al.'s proposed algorithm, based on Dijkstra's path search, assumes that QoS attributes are additive. Luo et al. claim that qualities such as duration and throughput are additive. However, it is easy to see that additivity depends on the nature of the composite service. For example, if a composite service invokes its constituent services in parallel, the overall duration is not the addition of the durations of the constituents. Our approach makes no assumption about QoS attributes. It can handle both additive and nonadditive QoS.

Nepal et al. [9] present a contribution-based method for propagating reputation from a composite service to its constituent services. Their approach assigns reputation based on (a) the contribution of constituent services to the composite, and (b) the difference between new and past reputation. Compared to our approach, Nepal et al. assign trust based on heuristics rather than statistics. Their approach requires predefined weights to be assigned to constituent services at the beginning. Our approach requires no predefined parameters except the number of constituent services.

Vu et al. [10] present a trust-based service ranking method as part of a service discovery approach. The input of this method is a list of service candidates obtained by semantic similarity matching based on providers' advertisements and consumers' requirements. It outputs a list of services ordered by their predicted QoS properties. Vu et al. use algorithms such as K-means clustering and trust-distrust propagation. However, they assume there exist several trusted third parties monitoring services. Also, they rely heavily on the P-Grid peer-to-peer architecture. In contrast, our approach is fully decentralized and makes no assumptions about the architecture.

Yue et al. [11] study a Bayesian network-based approach of generating a guidance of web service composition. Their approach has two parts. The first one is to construct a Bayesian network based on the past direct invocations among web services. Yue et al. iteratively find all direct and indirect invocations, calculate the conditional mutual information to test the conditional independencies. The second part of their approach is (given the Bayesian network) to use a Markov blanket to guide the composite service. Instead of using Bayesian networks, our approach applies a statistical model to determine trust in the constituents of a composite services.

Vu and Aberer [12] present a framework to estimate the quality of a service. Their framework involves three steps. In the first step, they build a Bayesian network model to represent QoS capabilities of the service. In the second step, they train the model with feedback from different sources to learn the unknown parameters for the service. In the third step, they estimate the quality of the service by making probabilistic inference on the basis of certain contextual information associated with the service. By contrast, we estimate the quality of multiple constituent services of a composite service. We estimate the unknown parameters from a Beta-Mixture model.

Mancioppi et al. [13] study the process fragmentation problem—identifying process fragments, a subset of process elements (i.e., constituent services), from a process model (i.e., a composition) for optimizing quality of service, simplifying and analyzing process models, and enabling the reuse of process fragments. In our approach, we make no assumption on how to fragment the composition. The constituent service can be a process element or a process fragment.

Lécué and Mehandjiev [14] consider the combination of both functional and nonfunctional properties while designing a service composition. The functional properties refer to the semantic quality of a service whereas the nonfunctional properties refer to QoS. They calculate the semantic quality of a composite service based on the degree of semantic similarity between an input and an output of the connected constituent services. To measure such degree, Lécué and Mehandjiev adopt the concept of a semantic link, a connection between the corresponding pairs of web service parameters. Thus they rank composite services based on both QoS and semantic QoS. Given a composed service, our approach considers how to analyze the constituent services based on nonfunctional properties only. The analytics learned from our approach can be incorporated into the above approach.

We propose an approach that assigns trust to constituents of composite services in such a way as to reward or punish constituents based on their contribution to the overall performance.

III. Approach: Beta-Mixture

Our approach has the following characteristics. First, it statistically learns the responsibilities and trust of the constituent services from the overall QoS. Second, it can use partial observations (if any) from the constituent services to improve the accuracy of its predictions. Third, it captures dynamism in service quality by incorporating recent observations with a discounting window.

Consider a general scenario where a consumer has been interacting with a service C, composed of K constituent services c_k ($k = 1, \ldots, K$). The consumer tracks the composite service iteratively as follows.

1) Collect and interpret a number of initial QoS observations as trust expectation $\mathbf{x}$ (Section III-A).
2) Bootstrap the initial trust θ of all constituent services and their corresponding responsibility π based on the initial evidence, where $\theta = \{\theta_1, \ldots, \theta_K\}$, and $\pi = \{\pi_1, \ldots, \pi_K\}$ (Section III-B).
3) Collect and interpret the new QoS observation as a trust expectation x_n (Section III-A).
4) Update the current responsibility π_k and constituent trust θ_k for each constituent service c_k based on the new observation x_n (Section III-B to Section III-E).

A. Background: Trust

We adopt Wang and Singh's [7] trust representation based on the beta probability distribution. They consider trust as binary evidence $\langle r, s\rangle$, where $r \geq 0$ and $s \geq 0$ are the numbers of positive and negative interactions, respectively. Based on this representation, a prediction can be made by calculating the expectation of the beta distribution $\alpha = \frac{r}{r+s}$. When a consumer observes a QoS from an interaction with a service, it translates the QoS into trust $\langle x, 1-x\rangle$, where $0 \leq x \leq 1$. For example, a consumer searches a keyword by a web searching service and receives the result in 0.5 seconds. The consumer may interpret the response time as $\langle 0.8, 0.2\rangle$, whose expectation is 0.8 indicating that it is a good experience. Suppose the consumer collects a series of observations, say $\mathbf{x} = \{x_1, \ldots, x_N\}$, which stand for $\{\langle x_1, 1-x_1\rangle, \ldots, \langle x_N, 1-x_N\rangle\}$. We apply Bayesian inference to update trust by simply adding the evidence together as $\langle \sum_{i=1}^{N} x_N, \sum_{i=1}^{N}(1-x_N)\rangle$ [15]. Note that we treat each QoS attribute separately because service consumers may have different interpretations and preferences of each QoS. In the following sections, the observations refer to the QoS observations written in the form of trust expectation, which is a single number between 0 and 1.

B. Background: Finite Mixture Models

The purpose of assigning trust to constituent services is to correctly reward or punish them based on their contribution to the overall performance. For example, consider the response time of a flight ticket booking service like Bing Travel that books tickets from either airline A or airline B. Suppose A provides a good response time but B does not, and the composite service ends up with a bad response time, thereby receiving a low trust value. Then, the greater part of the low trust value should be assigned to B. Conversely, if the composite service ends up with a good response time, then the greater part of the reward should be assigned to A. In other words, if the observations of a constituent service agree more with the overall observations, the constituent service makes a higher contribution. Such analysis can help consumers evaluate the constituent services and can be incorporated into monitoring tools like Amazon CloudWatch whereby consumers may attempt to optimize their compositions. For example, if airline A causes the poor response time but B is good, then the consumer should go to B directly, or the service composer should improve the response time of airline A.

However, observing the QoS of constituent services is not trivial, because they may be hidden from consumers. This makes the underlying QoS difficult to collect. For example, consumers may observe that a composite service is not accessible without knowing the availability of its underlying services. Our solution is as follows: Given the observations of the composite service, use a statistical approach to learn the trust of the constituents and their corresponding responsibilities.

We adopt the *finite mixture model* [16] to learn about the constituent services from the composed observations. The idea of finite mixture models is to use the superposition of multiple probability density distributions. Following our trust representation (Section III-A), we use the *Beta-Mixture* model [15], where the probability density distributions of the constituents are beta distributions with parameters $\theta_k = \langle r_k, s_k\rangle$, i.e., trust of the constituent c_k. This superposition can be written as

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k Beta(\mathbf{x}|\theta_k), \tag{1}$$

where $\mathbf{x} = \{x_1, \ldots, x_N\}$ is a series of observations from the composed probability distribution, K is the number of constituents, and π_k are the *mixing coefficients* of the constituent probability distributions. The mixing coefficients represent how many percentages of the constituent observations contribute to the overall observations. We can use the learned mixing coefficients as a measure of responsibility. Note that, since the finite mixture model and the EM algorithm can learn the constituent distributions (i.e., assign trust to each) directly, the motivation for learning the responsibility is that, when a constituent service contributes few observations to the overall observations, the traditional approach has difficulty in learning the trust for lack of evidence. In this case, responsibility can provide the information that the observations of the composite are not affected by this constituent. Therefore, trust in the constituent can be left unchanged. Besides, responsibility can guide service composers to optimize the service composition. For example, if the overall quality is dominated by one constituent (i.e., with high responsibility), instead of all constituents, the service composer should focus on optimizing the dominant constituent.

Now the problem becomes the following: Given a series of composed observations $\mathbf{x}$, what are the responsibilities π_k and the trust θ_k of the K constituents? We apply the *Expectation Maximization* (or EM) algorithm [17] to learn the parameters by maximizing the log likelihood, $\ln p(\mathbf{x}|\theta)$. The EM algorithm is a two-step iterative process. In our case, the *E-Step* holds θ_k and computes π_k. The *M-Step* holds π_k and computes θ_k. The two steps iterate until π_k and θ_k converge.

Here, the EM algorithm introduces a latent binary random variable $\mathbf{z} = \{z_1, \ldots, z_K\}$, each of whose components indicates whether a composed observation x_n is from constituent k, where $z_k \in [0, 1]$ and $\sum_k z_k = 1$. The distribution of $\mathbf{z}$ is specified in terms of π_k by $p(z_k = 1) = \pi_k$. Also, $p(\mathbf{z}) = \prod_{k=1}^{K} \pi_k^{z_k}$. Thus, we can rewrite $p(\mathbf{x})$ as

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{z})p(\mathbf{x}|\mathbf{z}) = \sum_{k=1}^{K} \pi_k Beta(\mathbf{x}|\theta_k), \quad (2)$$

where $p(\mathbf{x}|\mathbf{z}) = \prod_{k=1}^{K} Beta(\mathbf{x}|\theta_k)^{z_k}$.

The log likelihood of $p(\mathbf{x}|\theta)$ decomposes into two parts [18]:

$$\begin{aligned} \ln p(\mathbf{x}|\theta) &= \ln p(\mathbf{x}, \mathbf{z}|\theta) - \ln p(\mathbf{z}|\mathbf{x}, \theta) & (3) \\ &= \sum_{\mathbf{z}} q(\mathbf{z})(\ln p(\mathbf{x}, \mathbf{z}|\theta) - \ln q(\mathbf{z})) & (4) \\ &\quad - \sum_{\mathbf{z}} q(\mathbf{z})(\ln q(\mathbf{z}) - \ln p(\mathbf{x}, \mathbf{z}|\theta)) & (5) \\ &= L(q, \theta) - KL(q \parallel p), & (6) \end{aligned}$$

where $q(\mathbf{z})$ is a distribution of the latent variable $\mathbf{z}$, and $KL(q \parallel p) \geq 0$. Note that $L(q, \theta)$ contains complete-data log likelihood, $\ln p(\mathbf{x}, \mathbf{z}|\theta)$. From here, maximizing $\ln p(\mathbf{x}|\theta)$ is equivalent to maximizing $L(q, \theta)$ [4].

The EM algorithm departs from an initial guess of parameters $\theta^{\mathbf{0}}$ (i.e., $\mathbf{r}^0$ and $\mathbf{s}^0$). Then it iterates two steps to generate successive θ^1, θ^2, and so on, until the parameters converge. The *E-Step* holds the current parameters θ^{old} fixed to maximize $L(q, \theta^{old})$ with respect to $q(\mathbf{z})$ given by $p(\mathbf{z}|\mathbf{x}, \theta^{old})$. the *M-Step* holds the $q(\mathbf{z}) = p(\mathbf{z}|\mathbf{x}, \theta^{old})$ obtained from the *E-Step* fixed to find new parameters $\theta^{\mathbf{new}}$ that maximize the expectation of the complete-data log likelihood

$$\begin{aligned} Q(\theta, \theta^{old}) &= \sum_{\mathbf{z}} p(\mathbf{z}|\mathbf{x}, \theta^{old}) \ln p(\mathbf{x}, \mathbf{z}|\theta) & (7) \\ \theta^{new} &= \arg\max_{\theta} Q(\theta, \theta^{old}) & (8) \end{aligned}$$

We initialize the parameters by using *Fuzzy C-Means Clustering* (or FCM) [19] and *Method of Moments* (or MM) [20]. Bouguila et al. [21] show that combining FCM and MM provides good initial parameters for the EM algorithm to converge.

C. Online EM Algorithm

In real systems, however, consumers require an online algorithm to track the behavior of services with whom they interact, rather than collecting observations and learning all at once. For example, in the airlines example, a consumer books multiple flight tickets through Bing Travel over time. The consumer would maintain trust in Bing Travel and each airline regarding its response time, and update its trust in them on an ongoing basis upon making each booking. There are two reasons for using online algorithms. First, in most cases, it is not easy for consumers to collect enough information at once. Instead, a more feasible way is that consumers interact with services and evaluate their trustworthiness continually. An online trust model should track the behavior as the interaction proceeds. Second, online EM algorithms are shown to speedup the convergence and may lead to better solutions [22]. Therefore, we construct an online Beta-Mixture that learns the behavior of the constituent services by adopting an online variant of the EM algorithm [4].

To motivate the online EM algorithm, recall that, in the EM algorithm described in Section III-B, the observed variable $\mathbf{x}$ and latent variable $\mathbf{z}$ can be decomposed into $\{x_1, \ldots, x_N\}$ and $\{z_1, \ldots, z_n\}$, respectively. Note that, $L(q, \theta)$ is the summation of $q(z_n)(\ln p(x_n, z_n|\theta) - \ln q(z_n))$. Thus, we can deal with only one observation at each step. Neal and Hinton [4] show that updating only one observation at each *E-Step* can increase $L(q, \theta)$ (i.e., increase $p(\mathbf{x}|\theta)$) until convergence. This result shows that the online EM algorithm has the same capability as the EM algorithm.

Another variant of an online algorithm is the *stepwise* algorithm [22], which uses η_k steps to update the parameters in the *M-step*. Here, k keeps the count of the number of updates made to the parameters. We update the parameters as follows:

$$\theta^{new} = (1 - \eta_{\mathbf{k}})\theta^{old} + \eta_{\mathbf{k}} \arg\max_{\theta} Q(\theta, \theta^{old})$$

The stepwise algorithm performs similarly to online EM. In our evaluation, we use the online EM algorithm.

D. Partial Observation

The online EM algorithm suffers from some difficulties in dealing with dynamic behavior. For example, consider Bing Travel composed of two constituent airline services: A (good) and B (bad). They have the same responsibility π. When A turns bad and B turns good, the online EM algorithm may not detect the change accurately because the overall performance is not affected. Or, there are situations wherein different sets of constituents yield similar composed quality. In these situations, EM's accuracy may be poor but can be improved by considering partial observations from the constituents. For example, when searching flight tickets through Bing Travel, the consumer sometimes can partially observe the response time of each airline by looking at when its result shows up. These partial observations help EM calculate the responsibility more accurately.

In the *E-Step* of the online EM algorithm, we hold the old constituent parameters fixed to maximize the $p(z|x)$ with respect to the responsibility π. When there are no partial observations, we calculate the new π_k of constituent c_k as the posterior percentage of the probability density:

$$p(z_k|x) = \frac{\pi_k Beta(x|\theta_k)}{\sum_{j=1}^{K} \pi_j Beta(x|\theta_j)}, \quad (9)$$

where x is the composed observation, and $Beta$ is the beta probability density function. Consider that the performance of the k^{th} constituent is observed as x_k. Instead of Equation 9, we calculate the new π_k by comparing the probability density of x and x_k, i.e.,

$$p(z_k|x, x_k) = \frac{\min(Beta(x|\theta_k), Beta(x_k|\theta_k))}{\max(Beta(x|\theta_k), Beta(x_k|\theta_k))}. \quad (10)$$

Equation 10 follows from the idea of $p(z_k|x)$ expressing how probable it is that the constituent k contributes the composed observation x. Given that trust in the constituent is θ_k, the closer the probability densities of x and x_k are the more responsibility the constituent c_k should take. Section IV-A shows how partial observations can help improve prediction accuracy.

E. Discounting Window

Another difficulty of dealing with dynamic behavior is that, depending on the dynamism of the service, their next behavior may be similar to the most recent behavior or to the overall behavior. Suppose, airline A improves its response time by upgrading its servers. This improvement may or may not affect the response time of Bing Travel. The trust value of Bing Travel and A should be able to adapt to this change.

Many modern trust models introduce an adjustable *discounting factor* [23] to determine how much should past experience be weighted. If the services tend to follow their recent behavior, we should discard the old experience faster, i.e., employ a higher discounting factor. In contrast, if the services behave similarly to their overall performance, we should use a lower discounting factor, which yields more accurate predictions.

Our approach incorporates the discounting factor indirectly as a *discounting window*. The size of the discounting window indicates how many observations the model should consider. For example, let the window size be 50. Our approach collects as many observations as possible until the window is full. Upon the 51^{st} observation, our approach replaces the oldest observation with the new one. Section IV-C discusses how different sizes of discounting windows deal with dynamism.

The detailed procedure of our Beta-Mixture approach is shown in Algorithm 1.

IV. Evaluation

To show the generality of our approach, we consider different types of composition by adopting four composition operators f from Hang and Singh [15]: SWITCH, MAX (MIN), SUM, and PRODUCT. Depending on the characteristics of the QoS and the types of interactions (e.g., sequence, flow, and

Algorithm 1 Beta-Mixture with the online EM

Require: $\mathbf{x}$, x^{new}, π^{old}, θ^{old}

if size of $\mathbf{x}$ less than the discounting window size **then**
  append x^{new} to $\mathbf{x}$
  $x \leftarrow x^{new}$
else
  replace the oldest $x \in \mathbf{x}$ with x^{new}
end if
// E-Step
for $k = 1$ **to** K **do**
  if there exists partial observation x_k from c_k **then**
    $\pi_k^{new} \leftarrow \frac{\min(Beta(x|\theta_k), Beta(x_k|\theta_k))}{\max(Beta(x|\theta_k), Beta(x_k|\theta_k))}$
  else
    $\pi_k^{new} \leftarrow \frac{\pi_k^{old} Beta(x|\theta_k)}{\sum_{j=1}^{K} \pi_j^{old} Beta(x|\theta_j)}$
  end if
end for
//M-Step
for $k = 1$ **to** K **do**
  $\theta^{new} \leftarrow \arg\max_\theta Q(\theta, \theta^{old})$
end for
return $\mathbf{x}$, π^{new}, θ^{new}

TABLE I
Composition operator examples of different qualities and their interaction types.

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

case, as defined in BPEL [24]), different composition operators can be used. Table I shows some examples.

The SWITCH operator chooses the observations of one of the constituents as the composed observations based on a predefined multinomial distribution (i.e., π). The MAX (MIN) operator composes the observations by choosing the constituent with the highest (lowest) observation value. The SUM operator adds the observations from all constituents together (but no greater than one) as the overall set of observations. The PRODUCT operator calculates the composed observations by multiplying together all the constituent observations. Note that our approach is not limited to these operators, though we restrict our attention to these operators for the evaluation.

Due to the lack of real datasets, in our experiments, we assume that the constituent observation x_k is governed by a beta distribution. We construct a simple scenario where there is one composite service C with two constituent services c_1 and c_2. The behavior of each constituent is defined by the *probability*, *damping*, or *random* profiles. The constituent observations x_k generated by the *probability* profile is sampled from a fixed beta distribution with parameter $\theta = \langle r, s \rangle$. The *damping* profile starts with a parameter $\theta = \langle r, s \rangle$ and changes the parameter to $\theta' = \langle r', s' \rangle$ in the middle of the simulation. The *random* profile uniformly samples observations of constituent services as $x_k \in [0, 1]$.

The QoS of C is determined by the composition operator f. Here f can be any operator. We consider only SUM, MAX, PRODUCT, and SWITCH because they are the most natural ones. We calculate the root mean square error by

$$e = \sqrt{\frac{\sum_{k=1}^{K}(\alpha_k - x_k)^2}{K}}, \tag{11}$$

where α_k is the expected behavior based on trust estimation θ_k, and x_k is the observed behavior of the constituent c_k.

Table II shows the three approaches we compare in our experiments. BM and BMPO are both Beta-Mixture models. BM uses the traditional EM algorithm and admits no constituent observations, whereas BMPO adopts the online EM algorithm and considers partial observations. Nepal implements Nepal et al.'s approach [9]. Nepal represents trust as a single value from 0 to 1, which corresponds to our $\mathbf{x}$ and assigns trust heuristically. Nepal et al. predefine a fixed weight w_k to reflect the responsibility (i.e., our π_k) of each constituent c_k, whereas BM and BMPO learn the responsibility π_k dynamically. Nepal takes the previous averaged observation of both the composed and the constituent services and the current composed observation as inputs. We initialize each of the three approaches using FCM-MM, as described in Section III-B.

A. General Evaluation

The first simulation considers how our approach deals with static behavior. Both c_1 and c_2 use the *probability* profile. The parameters are set to $\theta_1 = \langle 2, 18\rangle$ and $\theta_2 = \langle 10, 10\rangle$. The multinomial distribution of the SWITCH operator is given by $\{0.65, 0.35\}$. The size of the discounting window is set to 50. The experiment is conducted for 500 timesteps. We use the first 50 observations to initialize all three approaches with FCM-MM, and dynamically track the remaining 450 observations. Here we let BMPO observe 50% partial observations.

Figure 1 shows the prediction error of all composition operators. In MAX, the actual responsibility π is obtained by the actual constituents the MAX operator chooses during the simulation. The actual responsibility is not available in the SUM and the PRODUCT cases. In general, because of the additional 50% partial observations, BMPO performs better than BM. Nepal produces worse results, except for θ_1 and π in SWITCH.

BM performs better than BMPO in SWITCH without partial observations. This is because SWITCH follows the intrinsic nature of the finite mixture model, i.e., the superposition of multiple probability densities, which means all the observations can be perfectly divided into two groups based on which constituent service the observations are chosen from. Thus, BM performs well without any partial observations. Considering partial observations based on heuristics obscures the accuracy insignificantly. However, when one constituent is barely contributing (e.g., the first constituent in MAX), BM cannot estimate trust (e.g., θ_1) of it because of the lack of evidence. It is because the overall quality is totally dominated by the second constituent. As mentioned in Section III-B, for those constituents with low responsibility, although the assigned trust may not be accurate, the consumer can ignore such constituents because their performance does not affect the composite service too much. For the SUM and PRODUCT operators, since all the observations involve both constituents, BM yields less accurate predictions. Note that the parameter of the constituent that contributes more can be learned more accurately than the other constituent. For example, for the SUM operator, constituent c_2 contributes more because it has higher expectation $\alpha = \frac{r}{r+s} = 0.5$ than constituent c_1 ($\alpha = 0.1$). In general, the constituent with higher responsibility provides more evidence for BM to learn from. Fortunately, the accuracy in all the cases is significantly improved with partial observations. With 50% partial observations, BMPO estimates trustworthiness of the constituent services with prediction error less than 15%.

To summarize, this result shows BM and BMPO provide generally better predictions than Nepal. Beta-Mixture is flexible in terms of partial observations. BMPO yields more accurate predictions with partial observations than BM and Nepal.

B. Partial Observations

The second simulation evaluates how BMPO improves the accuracy by considering the partial observations from constituent services. We follow the same configuration as Section IV-A except that the percentage of the partial observations is varied from 0% to 100%. Figure 2 shows the prediction error of BMPO for all composition operators. In SUM, the prediction accuracy improves as the availability of partial observations increases. In PRODUCT, only the accuracy of the good constituent ($\alpha = 0.5$) significantly improves, because the overall observation is dominated by the bad constituent ($\alpha = 0.1$). Similarly, in SUM and MAX, the dominating constituent is the good one. Partial observations significantly help predict those less dominating bad constituents. This experiment shows that partial observations improve the prediction accuracy of the constituent services, especially the less dominating ones.

C. Discounting Windows

The last simulation shows how BMPO deals with the dynamic behavior of the constituent services by adjusting the size of the discounting window. Section III-E claims incorporating discounting windows can help deal with services with varying levels of dynamism. Here, constituent c_1 uses two dynamic behavior profiles. The *damping* profile starts with $\theta_1 = \langle 18, 2\rangle$ ($\alpha_1 = 0.9$) and turns to $\theta'_1 = \langle 2, 18\rangle$ ($\alpha'_1 = 0.1$) in the middle of the experiment. The *random* profile uniformly samples $x_1 \in [0, 1]$. Constituent c_2 adopts the *probability* profile, where $\theta_2 = \langle 10, 10\rangle$ ($\alpha_2 = 0.5$). We consider ten sizes of discounting windows: $10, 20, \ldots, 100$. The composition operator is SWITCH with multinomial probability $\{0.65, 0.35\}$. There are a total of 500 observations. The first $10, 20, \ldots, 100$ observations are used for initializing BMPO with corresponding sizes of discounting windows and the rest are used for

TABLE II
APPROACHES COMPARED IN OUR EXPERIMENTS.

Approach	Description	Online	Observation	Dynamism
BM	Beta-Mixture	No	None	Discounting Window
BMPO	Beta-Mixture with Partial Observations	Yes	Partial	Discounting Window
Nepal	Nepal et al. [9]	Yes	None	No

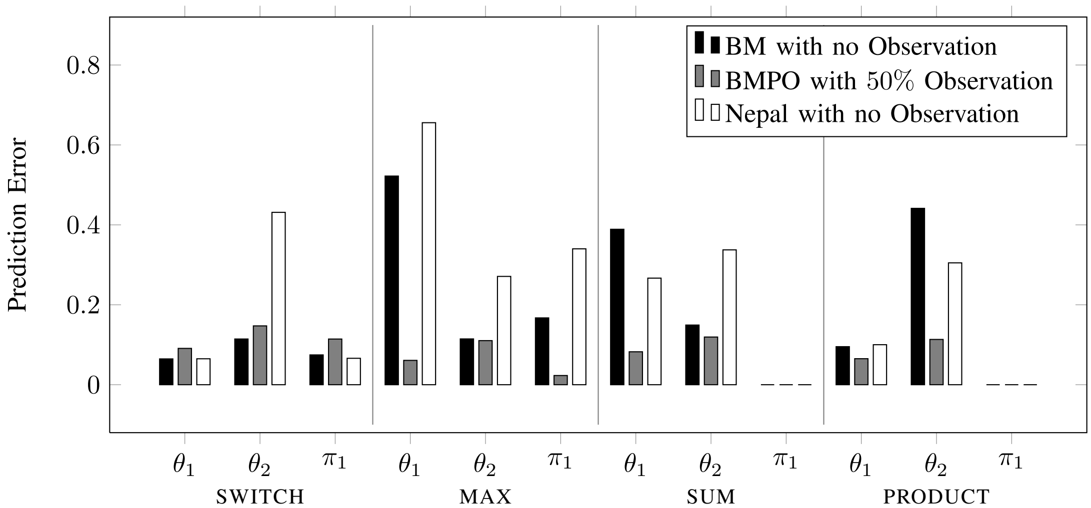
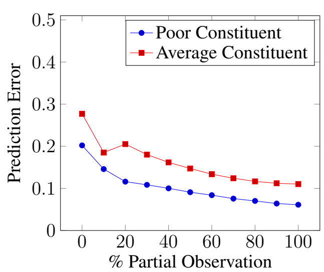
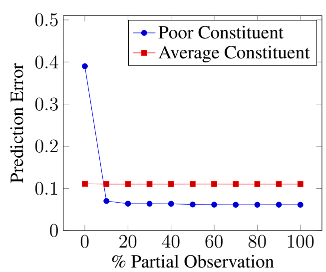


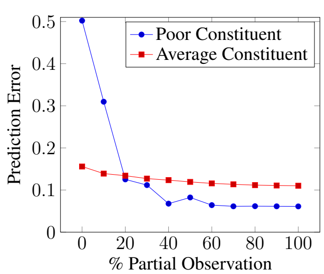
Fig. 1. The average prediction error e of the parameters θ and the responsibility π with all composition operators (responsibility π for the SUM and PRODUCT operators is not available because there are no actual values).



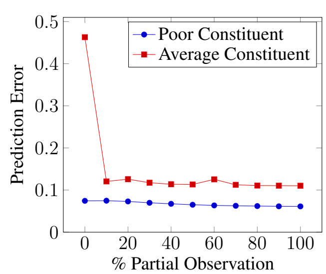
(a) SWITCH



(b) MAX

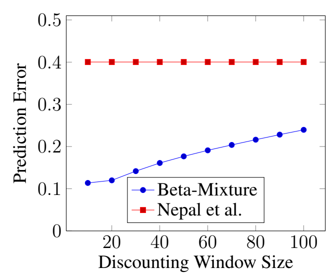


(c) SUM

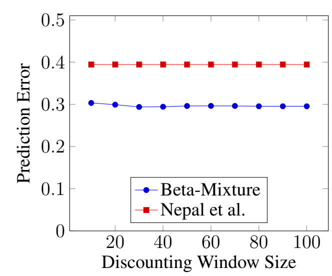


(d) PRODUCT

Fig. 2. The error of predicting trust of constituent services using BMPO with varying percentages of partial observations.



(a) *Damping*



(b) *Random*

Fig. 3. The error of predicting *damping* and *random* constituent service using Nepal and BMPO with varying sizes of discounting windows.

online learning. We compare the last 400 predicted and actual observations. We use the setting of 50% partial observations.

Figure 3 shows the prediction error of the dynamic constituent using BMPO with varying sizes of discounting windows, compared with Nepal. Nepal produces the same error across the x-axis because it does not use discounting windows. In Figure 3(a), the *damping* constituent can be better predicted with a smaller discounting window. However, in Figure 3(b), the *random* constituent is less predictable regardless of the sizes of discounting windows, because its current observation is independent to the previous observation. Using the size of 30 yields the best prediction, although the difference is not significant. Nepal lacks the flexibility of dealing with both dynamic profiles.

D. Discussion

Our evaluations show that (a) Beta-Mixture can provide good predictions of both responsibility and constituent parameters with four composition operators: SWITCH, MAX, SUM, and PRODUCT; (b) Beta-Mixture can improve prediction accuracy by considering partial observations; and (c) Beta-Mixture deals flexibly with dynamism by adjusting the size of the discounting window.

V. CONCLUSIONS

This paper proposes an approach for assigning trust to the constituents of composite services. It models a composite service as a beta-mixture and supports the following features:

- assigns trust to the constituents behind the composite service based on their contribution to the overall performance, and provides a responsibility measurement for each constituent;
- improves prediction accuracy by considering partial observations (if any) from the constituents; and
- accommodates dynamic behavior by introducing a discounting window.

Our future work includes studying more complicated settings involving more constituents and deeply nested compositions with real datasets such as for scientific workflows. Another possible extension is to learn the number of constituents behind the composite service because in some cases, consumers may lack this information. Besides, our approach should be able to be combined with beta-distribution based trust models. By doing this, we can incorporate indirect evidence such as referrals into our approach. Thus, additional trust information can be aggregated to improve the accuracy or expedite the initialization process.

ACKNOWLEDGMENTS

This research was supported by the U.S. Army Research Office (ARO) under grant W911NF-08-1-0105 managed by NCSU Secure Open Systems Initiative (SOSI). We thank the reviewers for helpful comments.

REFERENCES

[1] E. M. Maximilien and M. P. Singh, "A framework and ontology for dynamic web services selection," *IEEE Internet Computing*, vol. 8, no. 5, pp. 84–93, 2004.

[2] Y. Liu, A. H. Ngu, and L. Z. Zeng, "QoS computation and policing in dynamic web service selection," in *Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers & Posters*. New York: ACM Press, 2004, pp. 66–73.

[3] C.-W. Hang, Y. Wang, and M. P. Singh, "Operators for propagating trust and their evaluation in social networks," in *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, vol. 2. Budapest: IFAAMAS, 2009, pp. 1025–1032.

[4] R. M. Neal and G. E. Hinton, "A view of the EM algorithm that justifies incremental, sparse, and other variants," in *Learning in Graphical Models*. Cambridge, MA: MIT Press, 1999, pp. 355–368.

[5] Y. Zhang, K.-J. Lin, and J. Y.-J. Hsu, "Accountability monitoring and reasoning in service-oriented architectures," *Service Oriented Computing and Applications*, vol. 1, no. 1, pp. 35–50, Apr. 2007.

[6] S. Paradesi, P. Doshi, and S. Swaika, "Integrating behavioral trust in web service compositions," in *Proceedings of the 7th IEEE International Conference on Web Services (ICWS)*. Los Angeles: IEEE Computer Society, Jul. 2009, pp. 453–460.

[7] Y. Wang and M. P. Singh, "Formal trust model for multiagent systems," in *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*. Hyderabad, India: IJCAI, Jan. 2007, pp. 1551–1556.

[8] J.-Z. Luo, J.-Y. Zhou, and Z.-A. Wu, "An adaptive algorithm for QoS-aware service composition in grid environments," *Service Oriented Computing and Applications*, vol. 3, no. 3, pp. 217–226, Sep. 2009.

[9] S. Nepal, Z. Malik, and A. Bouguettaya, "Reputation propagation in composite services," in *Proceedings of the 7th IEEE International Conference on Web Services (ICWS)*. Los Angeles: IEEE Computer Society, Jul. 2009, pp. 295–302.

[10] L.-H. Vu, M. Hauswirth, and K. Aberer, "QoS-based service selection and ranking with trust and reputation management," in *On the Move to Meaningful Internet Systems: CoopIS, DOA, and ODBASE*, ser. Lecture Notes in Computer Science, R. Meersman and Z. Tari, Eds. Springer Berlin / Heidelberg, 2005, vol. 3760, pp. 466–483.

[11] K. Yue, W. Liu, and W. Li, "Towards web services composition based on the mining and reasoning of their causal relationships," in *Advances in Data and Web Management*, ser. Lecture Notes in Computer Science. Berlin: Springer Berlin, 2007, pp. 777–784.

[12] L.-H. Vu and K. Aberer, "Towards probabilistic estimation of quality of online services," in *Proceedings of the 7th IEEE International Conference on Web Services (ICWS)*. Los Angeles: IEEE Computer Society, Jul. 2009, pp. 99–106.

[13] M. Mancioppi, O. Danylevych, D. Karastoyanova, and F. Leymann, "Towards classification criteria for process fragmentation techniques," in *Business Process Management Workshops (1)*, ser. Lecture Notes in Business Information Processing. Clermont-Ferrand, France: Springer, Aug. 2011, pp. 1–12.

[14] F. Lécué and N. Mehandjiev, "Seeking quality of web service composition in a semantic dimension," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 6, pp. 942–959, Jun. 2011.

[15] C.-W. Hang and M. P. Singh, "Trustworthy service selection and composition," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 6, no. 1, pp. 5:1–5:17, Feb. 2011.

[16] G. McLachlan and D. Peel, *Finite Mixture Models*. New York, NY, USA: Wiley-Interscience, 2000.

[17] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[18] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, Aug. 2006.

[19] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. Norwell, MA: Kluwer Academic Publishers, 1981.

[20] B. D. Fielitz and B. L. Myers, "Estimation of parameters in the beta distribution," *Decision Sciences*, vol. 6, no. 1, pp. 1–13, Jan. 1975.

[21] N. Bouguila, D. Ziou, and E. Monga, "Practical Bayesian estimation of a finite beta mixture through Gibbs sampling and its applications," *Statistics and Computing*, vol. 16, no. 2, pp. 215–225, Jun. 2006.

[22] P. Liang and D. Klein, "Online EM for unsupervised models," in *Proceedings of Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*. Boulder, Colorado: The Association for Computational Linguistics, May 2009, pp. 611–619.

[23] Y. Wang, C.-W. Hang, and M. P. Singh, "A probabilistic approach for maintaining trust based on evidence," *Journal of Artificial Intelligence Research*, vol. 40, pp. 221–267, 2011.

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