Online Reliability Time Series Prediction for Service-Oriented System of Systems

Lei Wang[1,2], Hongbing Wang[1,⋆],
Qi Yu[3], Haixia Sun[1], and Athman Bouguettaya[4]

[1] School of Computer Science and Engineering, Southeast University, China
[2] Dept. of Management Science and Engineering, Nanjing Forestry University, China
{leiwang,hbw,haixiasun}@seu.edu.cn
[3] College of Computing and Information Sciences, Rochester Institute of Tech, USA
qi.yu@rit.edu
[4] School of Computer Science and Information Technology, RMIT, Australia
athman.bouguettaya@rmit.edu.au

Abstract. A Service-Oriented System of System (or SoS) considers system as a service and constructs a value-added SoS by outsourcing external systems through service composition. To cope with the dynamic and uncertain running environment and assure the overall Quality of Service (or QoS), online reliability prediction for SoS arises as a grand challenge in SoS research. In this paper, we propose a novel approach for component level online reliability time series prediction based on Probabilistic Graphical Models (or PGMs). We assess the proposed approach via invocation records collected from widely used real web services and experiment results demonstrate the effectiveness of our approach.

1 Introduction

A SoS pools computing resources together to create a new, value-added, and more complex system. As a new computing paradigm that has attracted significant popularity, Service-Oriented Architecture (SOA) provides a principled mechanism to construct a SoS [1, 5] by dynamically integrating its component systems through service composition. It is anticipated that a service-oriented SoS runs under a complicated and highly dynamic environment. Hence, the runtime QoS assurance is of significant importance for a service-oriented SoS.

Proactive Fault Management (or PFM) offers an effective mechanism to enhance the reliability of software systems [6]. Nonetheless, to achieve PFM in service-oriented SoSs, a central challenge lies in automatic and accurate prediction of the reliability of the SoS. In particular, a self-* (*configuration, healing, optimization, or protection*) SoS demands online reliability prediction that accurately predicts the reliability of the SoS in nearly realtime to deal with the

⋆ This work is partially supported by NSFC (No.61232007) and Doctoral Fund of Ministry of Education of China (No.20120092110028) and JSNSF (No.BK2010417) and PSSF of higher education in Jiangsu Prov. (No.2013SJB6300051).

S. Basu et al. (Eds.): ICSOC 2013, LNCS 8274, pp. 421–428, 2013.

highly dynamic running environment. As illustrated in Figure 1, online reliability prediction estimates the system's reliability in the "near future" (i.e., the prediction time period of Δt_p). More specifically, Δt_l is defined as the *leading time*, which starts from t and ends when a user invokes a SoS. Δt_p is defined as the *prediction period*, which corresponds to a future invocation time period. Δt_d is the *data window size* for historic records.

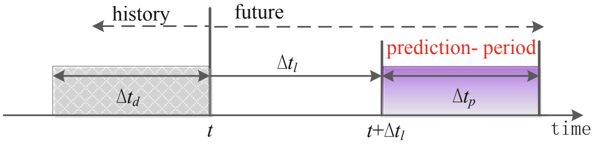


Fig. 1. Schematic View of Online Reliability Prediction

In a service-oriented SoS, the prediction period Δt_p is determined by the execution period of component level systems and the unstable communication links. Hence, the length of invocation is usually uncertain, which makes the length of Δt_p vary from one user's requirement to another [3]. As most existing *online failure prediction* methods are designed for failure probability prediction in a fixed time period, they are not directly applicable for a varied prediction period. In contrast, a viable reliability prediction approach for a service-oriented SoS should capture the changes of reliability during a variable prediction period. Assume that Δt_p is long enough for most user requirements. The key idea of reliability prediction for a service-oriented SoS is to predict the reliability *time series* during the period of Δt_p.

To our best knowledge, this is the first work on *online reliability time series prediction* for service-oriented SoS. It is fundamentally different from relevant existing works, including online failure prediction for traditional computer systems [6] and reliability prediction in service computing [11]. Despite the running environment of a service-oriented SoS is complex and highly dynamic, the reliability of the overall system is mainly affected by several major factors, which include (1) the unstable communication links between the component systems, (2) the internal working status of component systems, and (3) the loading capacity of component systems under the current throughput. It is always difficult and sometimes even impossible to directly collect a outsourced from third-party service providers remote system's performance parameters. Nonetheless, the above factors also significantly affect the throughput and response time of component systems. More importantly, parameters like throughput and response time can be easily obtained via client-side evaluation of the target component systems. This key observation allows us to analyze the component level system's reliability along with its throughput and response time, which provides a holistic view of the system's running state and its surrounding environment.

The major contribution of the paper centers around using Probabilistic Graphical Models (*PGMs*) to analyze historical and current system parameters, including *Reliability*, *Throughput* and *Response Time*. The Markov chain rules are employed to capture the causal relationships between adjacent time series of the system parameters, which represented as Conditional Probability Tables (*CPTs*). These *CPTs* will be used together with the PGM to make online reliability prediction based on the current system parameters. Experiments conducted over real-world web services justify the effectiveness of our approach.

2 Related Work

It has been discovered that the arrival times of atomic web services reliability follow an *Erlangian* distribution because the failures' arrival times are dependent on the operating states (e.g. idle and active states) [4]. Collaborative filtering based approaches have been widely employed in service computing to predict the QoS (including reliability) of previously unknown services [8–11]. As an example, a matrix factorization based approach is used to predict the missing values in service component's user-item failure probability matrix. With the component level service reliability prediction results, the system level reliability is aggregated by the composition structures [12].

Online failure prediction in traditional computer systems aims to identify during runtime whether a failure will occur in the *near future*. A taxonomy for existing *online failure prediction* approaches mainly include three categories [6], which are *Failure tracking*, *Symptom monitoring* and *Detected error reporting*. All of these methods depend on a directly server side observation on system working status or system log files.

3 Online Reliability Time Series Prediction

In this section, a Probabilistic Graphical Models based Reliability Time Series Online Prediction for component level service-oriented System of Systems (*PGMs-RTSOP*) is proposed.

The typical *PGMs* model supporting dynamic changes of the time, which is Dynamic Bayesian Networks (*DBNs*) model [2]. The changes from adjacent time points of historic system parameters studied in this paper exist no obviously causality, due to the dynamics of the SoS runtime environment. While the regulation of changes from continuous system parameters time series (time series is composed of a plurality of time points) can often reflect a specific event (such as software version upgrade). The continuous change of the SoS system parameters time series satisfy the Markov chain rules.

In this paper, we propose *m_DBNs* (see Figure 2), an augmentation of the traditional DBNs that uses the Markov chain rule to model the causal relationship between adjacent system parameters (e.g., the current time series t and the near future time series $t+1$). The nodes in the *m_DBNs* correspond to the system parameters, including *Response Time* (RT), *Throughput* (T) and *Reliability* (R). The parameters are represented by the time series patterns, which are referred to as *motifs* [7], discovered from historical data to facilitate the prediction of future time series.

Since both *Response Time* and *Throughput* are easy to measure, they can be regarded as observed variables for the current time series t. On the other hand, directly measuring *Reliability* is difficult so its state is usually derived based on *Response Time* and *Throughput*. Hence, *Reliability* can be regarded as a hidden variable whose value is affected by both *RT(t)* and *T(t)*. The dependencies between different nodes are denoted by the arcs in the *m_DBNs*.

Using the proposed *PGMs-RTSOP* for online reliability time series prediction consists of the following four major steps.

1. **Motifs Discovery:** Motifs discovery is to identify the featured patterns of time series from parameters of RT, T and R of a component system. We divide the historic parameter into two different time series. Hence, motifs for each parameter can be divided as two categories: motifs for current time t and motifs for near future *t+1*. To achieve motifs discovery, we group together similar time series in historic parameters (RT, T, R) through a clustering algorithm (e.g., K-means). Consequently, motifs are defined as the centroids of the resultant clusters. We will use RT as an example.
   Assume that the length of each time series is Δt_p so the n time series of RT is given by $RT_{t-n\Delta t_p}, \cdots, RT_{t-2\Delta t_p}, RT_{t-\Delta t_p}, RT_t$. Specifically, the i-th time series of RT is defined as: $RT(i) = \overrightarrow{RT_{((t-(n-i-1))\Delta t_p,(t-(n-i))\Delta t_p)}}$, where $i = 1, \cdots, n$. We further assume that s is the time interval that RT is collected. Hence, each time series can be represented by a vector with size $\frac{\Delta t_p}{s}$ and the distance between two time series $RT(i)$ and $RT(j)$ can be calculated as

$$dis(RT(i), RT(j)) = \sqrt{\sum_{\mathrm{k}=1}^{\frac{\Delta t_p}{s}} (RT(i)_k - RT(j)_k)^2} \tag{1}$$

   The motifs are calculated as the centroids of the clusters.
   We re-divide the time period for parameter RT and get the near future system parameters. We choose the time period after Δt_l for each $RT(i)$, which is defined as $RT(i)_{t+\Delta t_l}$. We will use the motifs discovery method presented for *RT(i)* to discover motifs in $RT(i)_{t+\Delta t_l}$.
2. **Motifs based Time Series Representation:** In this step, we label system parameters by the discovered motifs. In particular, each time series of a system parameter will labeled by the nearest motifs discovered in the previous step. Again, we will use RT as an example and the same process applies to other parameters. Given the k motifs for RT, $RT\_motifs(j), j = 1, \cdots, k$. we label $RT(i)$ by $RT\_motifs(m)$. i.e.,

$$RT(i) \leftarrow Label\_of(RT\_motifs(m)), \text{ where} \tag{2}$$
$$dis(RT(i), RT\_motifs(m)) \leq dis(RT(i), RT\_motifs(j)), \ j \neq m \tag{3}$$

3. **Conditional Probability Table Construction:** In the proposed *m_DBNs*, we define its original state B_0 by the motifs of system parameter time series at time t. The *m_DBNs* captures the transition model of

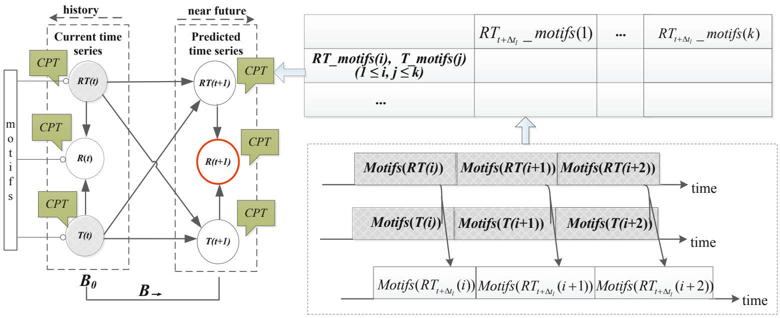


Fig. 2. The Construction Process of the *CPTs*

causal relations for the system parameter time series from time t to time $t+1$, i.e., the state transition $B_{\rightarrow}$ from current state to the near future. Each node in the *m_DBNs* model is associated with a conditional probability table (or CPT). As an example, the CPT for node $RT(t+1)$ is shown in Figure 2. Each row in the CPT corresponds to one possible combinations of values (represented by motifs) taken by its dependent nodes (or conditional nodes) (i.e., $RT(t)$ and $T(t)$). Each column denotes one of the k motifs of $RT(t+1)$, .i.e., $RT_{t+\Delta t_l}\_motifs(1), \cdots, RT_{t+\Delta t_l}\_motifs(k)$. We analyze the labeled historical parameters to gather statistics on historic parameters to construct the $CPTs$ for each node. Let $motifs(RT(i)) = RT\_motifs(\alpha)$, $motifs(T(i)) = T\_motifs(\beta)$ and $motifs(RT_{t+\Delta t_l}(i)) = RT_{t+\Delta t_l}\_motifs(\delta)$. The probability from all of the causal relation satisfies $(RT\_motifs(\alpha), T\_motifs(\beta)) \rightarrow RT_{t+\Delta t_l}\_motifs(\delta)$ will be the value of the cell at the intersection of the row $(RT\_motifs(\alpha), T\_motifs(\beta))$ and column $RT_{t+\Delta t_l}\_motifs(\delta)$.

4. **Online Reliability Time Series Prediction:** The prediction is carried out through the *m_DBNs* model using the following three steps:
   (a) The real-time system parameters $RT(t), T(t)$ will be labeled by their nearest motifs, which results in $RT(t)\_motifs(\alpha)$ and $T(t)\_motifs(\beta)$.
   (b) The motifs of $RT(t)$ and $T(t)$ will be the conditional item in the CPT of $RT(t+1)$ and $T(t+1)$. Hence, the prediction results for parameter $RT(t+1)$ and $T(t+1)$ will be the motifs holding the maximal probability by the conditional item of $RT(t)\_motifs(\alpha)$ and $T(t)\_motifs(\beta)$.
   (c) The predicted motifs of $RT(t+1)$ and $T(t+1)$ will be substituted into the CPT of $R(t+1)$ as the conditional items to get the cell holding the maximal probability value. The prediction result for reliability time series of $R(t+1)$ will be the motifs of the cell's column name.

4 Experiments

We conduct a set of experiments to assess the effectiveness of the proposed *PGMs-RTSOP*. Since there is no sizable service dataset that provides continuous observation on system parameters RT, T and R, we build our own dataset by invoking a selected set of web services and recording the *Response Time*, *Throughput* and *Reliability* of the service invocations.

4.1 Data Set Description

To build our dataset, we download the WSDL files of web services, including: (1) the well-known popular web services, such as *bing, SalesForce, PayPal, ebay, Google Search, Amazon*; (2) web services from WebserviceX service repository; and (3) three popular web services published in China: *Weather, QQ Online*, and *DomesticAirline*. We convert the WSDL files into java classes and generate java test files using Axis2. Finally, service invocation requests for a selected API of each web services are sent out every $200ms$ from our PC client and the response time, the size of the returned data (bit), and return type of the HTTP message are collected. Let the data size for the returned message from a remote web service be *res_size*. We represent the *Throughput* as the data size successfully transmitted within a unit time from the web service, i.e., $\frac{res\_size}{RT*1000}$ ($kbps$). We set an upper limit for the response time of a service invocation as $1000ms$. If the response time goes beyond the limit, it will be considered as a *timeout* error. We collect the system parameters continuously for 24 hours.

We preprocess to the collected system parameters as follows. We define the time interval for continuous 10 returned messages as a time series point. Since a service request is sent every $200ms$, a time slide is set to $2s$, so each time series contain 10 time points. Then, for a given time point, $RT = (\sum_{i=1}^{10} RT_i)/10$, $T = (\sum_{i=1}^{10} T_i)/10$, $R = e^{-\gamma \cdot t}$. where, RT_i is the *Response Time* and T_i is *Throughput* parameter for each invocation during the time point. γ is the proportion of failure invocations and $t = 2s$. We set $\Delta t_p = 20s$ and $\Delta t_l = 4s$. The historic time series parameters for time t and *t+1* is built separately. More specifically, the time series for time t: the collected 24-hour *Response Time*, *Throughput* and *Reliability* system parameters are divided into 4320 ($= 24 * 60 * 3$) continuous time series, which are represented as $RT^j(i)$, $T^j(i)$, $T^j(i)$, where j indexes web services and i indexes time series. To generate the time series for time *t+1*, we move right for two time points (i.e. the time span of Δt_l), also each 10 continuous time points as a time series, represented as $RT^j_{t+\Delta t_l}(i)$, $T^j_{t+\Delta t_l}(i)$, $R^j_{t+\Delta t_l}(i)$.

4.2 Approaches to Compare

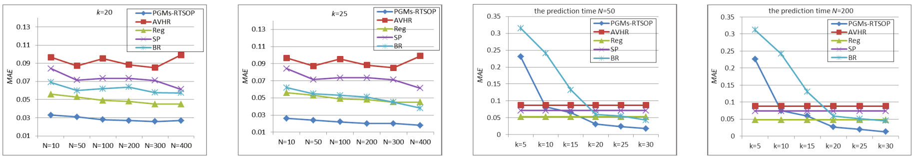
We implement four different reliability time series online prediction methods to compare with our approach. Specifically, the four comparison approaches include:

- Average Value of Historic Reliability ($AVHR$): The 10 points of the predicted time series result all equals the historic average reliability value.
- Regression (*Reg*): A least square fitting function is calculated according to historical reliability time series parameter. The fitting function is used to predict the near future reliability time series.

- Similarity based Prediction (*SP*): Let the real-time observed reliability time series be R_t. $R_{t+\Delta t_l}(i)$ will be the reliability time series prediction result, when $R(i)$ is the nearest historic (in time t) *Reliability* time series to R_t.
- Bayes' Rules (*BR*): We collect the statistics on the conditional probability of motifs for the historic parameter R, i.e., $P(R(t)_{t+\Delta t_l}\_motifs|R(t)\_motifs)$. With the conditional item of real-time system parameter of R (labeled by its motifs), the motifs will be the prediction result, which makes $R(t)_{t+\Delta t_l}\_motifs$ obtain a maximal probability.

4.3 Performance Comparison

We set the number of motifs as *k=20* and *k=25* in *PGMs-RTSOP* and *BR* and compare the averaged *MAE* (Mean Absolute Error) [11] of different prediction methods for 10, 50, 100, 200, 300 and 400 number of predictions. The experimental results are shown in Figure 3 (a-b).



(a) **(b)** **(c)** **(d)**

Fig. 3. The Prediction Performance Comparison. (a) *MAE*, k=20; (b) *MAE*, k=25; (c) *MAE*, N=50; (d) *MAE*, N=200.

As can be seen from the results, for *BR* and the proposed *PGMs-RTSOP*, the prediction accuracy increases slightly with the increasing of prediction times, while the approaches of *AVHR*, *Reg* and *SP* change differently as the number of predictions increases. In addition, the curves of the *PGMs-RTSOP* and *BR* are closer to a straight line whereas those of *AVHR* and *SP* show obvious fluctuations. This observation demonstrates the robustness of prediction performance of *PGMs-RTSOP* and *BR*. This is mainly due to that the dependency between adjacent motifs have certain patterns as the system parameters change, which makes the proposed *m_DBNs* model more suitable to carry out online reliability time series prediction.

In the second set of experiments, we vary the number of motifs for from 5 to 30. Each method is executed 50 and 200 times, respectively. Also, in the *BR* method, the number of motifs is the same with our *PGMs-RTSOP* method. As can be seen from Figure 3 (c-d), the motifs number exhibits a significant impact on both *PGMs-RTSOP* and *BR*. The larger value of k results in a smaller *MAE*. When $k \geq 20$, the improvement of prediction accuracy slows down. Since the *AVHR*, *Reg* and *SP* do not exploit motifs, their *MAE* values remain constant over different k values. When $k \geq 20$, the prediction accuracy of *PGMs-RTSOP* significantly outperforms all other four approaches.

5 Conclusion

In this paper, we present an online reliability time series prediction approach, referred to as *PGMs-RTSOP* for service-oriented SoS. The proposed approach integrates motifs into the traditional dynamic Bayesian Networks, resulting in an *m_DBNs* model, to deal with the uncertain SoS runtime environment. We conduct experiments on real-world web services to evaluate the effectiveness of the proposed approach. Four other reliability prediction approaches are implemented for comparison purpose. The experimental results demonstrate the high prediction accuracy and the robust prediction performance of *PGMs-RTSOP*. The proposed online reliability time series prediction approach is instrumental to achieve *online fault removal* and *fault tolerance recovery* mechanisms under a complicated and changing environment.

References

1. Cook, T.S., Drusinksy, D., Shing, M.T.: Specification, validation and run-time monitoring of soa based system-of-systems temporal behaviors. In: IEEE International Conference on System of Systems Engineering, SoSE 2007, pp. 1–6. IEEE (2007)
2. Koller, D., Friedman, N.: Probabilistic graphical models: principles and techniques. MIT press (2009)
3. Lelli, F., Maron, G., Orlando, S.: Client side estimation of a remote service execution. In: 15th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, MASCOTS 2007, pp. 295–302. IEEE (2007)
4. Mansour, H.E., Dillon, T.: Dependability and rollback recovery for composite web services. IEEE Transactions on Services Computing 4(4), 328–339 (2011)
5. Rothenhaus, K.J., Michael, J.B., Shing, M.T.: Architectural patterns and autofusion process for automated multisensor fusion in soa system-of-systems. IEEE Systems Journal 3(3), 304–316 (2009)
6. Salfner, F., Lenk, M., Malek, M.: A survey of online failure prediction methods. ACM Computing Surveys 42(3), 10:1–10:42 (2010)
7. Shellman, E.R., Burant, C.F., Schnell, S.: Network motifs provide signatures that characterize metabolism. Molecular BioSystems 9(3), 352–360 (2013)
8. Yu, Q.: Decision tree learning from incomplete qos to bootstrap service recommendation. In: Proc. 19th IEEE International Conference on Web Services, ICWS 2012, pp. 194–201 (2012)
9. Yu, Q.: Qos-aware service selection via collaborative qos evaluation. World Wide Web Journal (accepted to appear, 2013)
10. Yu, Q., Zheng, Z., Wang, H.: Trace norm regularized matrix factorization for service recommendation. In: Proc. 20th IEEE International Conference on Web Services, ICWS 2013 (2013)
11. Zheng, Z., Lyu, M.R.: Collaborative reliability prediction of service-oriented systems. In: Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering, ICSE 2010, pp. 35–44. ACM (2010)
12. Zheng, Z., Ma, H., Lyu, M., King, I.: Collaborative web service qos prediction via neighborhood integrated matrix factorization (early access articles). IEEE Transactions on Services Computing (2012)