Knowledge-Based Systems 92 (2016) 151–168

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Effective service composition using multi-agent reinforcement learning

Hongbing Wang [a,*], Xiaojun Wang [a], Xingzhi Zhang [a], Qi Yu [b], Xingguo Hu [a]

[a] School of Computer Science and Engineering, Southeast University, Nanjing, China
[b] College of Computing and Information Sciences, Rochester Institute of Technology, USA

ARTICLE INFO

Article history:
Received 29 October 2014
Revised 19 October 2015
Accepted 22 October 2015
Available online 3 November 2015

Keywords:
Web service composition
Multi-agent
Reinforcement learning
Q-learning

ABSTRACT

As online services may keep evolving, service composition should maintain certain adaptivity especially for a dynamic composition environment. Meanwhile, the large number of potential candidate services poses scalability concerns, which demand efficient composition solutions. This paper presents a multi-agent reinforcement learning model for Web service composition that effectively addresses the above challenges. In particular, we model a service composition as a Markov Decision Process. Based on the model, agents in a team would benefit from one another. In contrast to single-agent reinforcement-learning, our method can speed up the convergence to an optimal policy. We develop two multi-agent reinforcement learning algorithms. The first one introduces the concept of articulate state and distributed Q-learning to speed up the convergence time. The second one proposes the experience sharing strategy to improve the efficiency of learning. As the learning process continues throughout the life-cycle of a service composition, our algorithms can automatically adapt to the change of environment and the evolving component services. We conduct a simulation study to compare our algorithm with other similar reinforcement learning algorithms, including the traditional Q-learning algorithm, a multi-agent Sarsa algorithm, a Q-learning algorithm based on gaussian process, and a multi-agent Q-learning algorithm, to justify the effectiveness of our model and algorithm.

1. Introduction

Service Oriented Computing (SOC) provides a powerful computing paradigm, where data, software and hardware can all be encapsulated as services that are accessed over the Internet. In an SOC environment, application builders can focus on the deployment of business logics, saving time and efforts on the establishment of the computing infrastructure. It has been widely recognized that Web services are self-describing and building blocks for rapid, low-cost composition of distributed applications [18]. In practice, a single Web service may not be sufficient to perform complex tasks. We usually need to combine multiple existing services together to meet customers' complex requests. With today's SOC technology, such composition is usually performed by service engineers. However, the Web environment is highly dynamic, and most Web services are evolving over the time. A service engineer cannot always foresee all the changes that could happen in the future. A manual service composition may also be too rigid to adapt to a dynamic environment. As a result, a number of methods have been investigated for dynamic service composition.

Existing dynamic Web service composition approaches primarily rely on workflow and AI programming based techniques [7,8,12,20,25]. Workflow based approaches implement the business logic by creating a business process, where the business information is represented by nodes in a process graph and the function to be performed will be achieved in the form of Web services. Typically, a workflow needs to be manually crafted, making it hard to automate [12,20]. AI planning is commonly used to automate Web service composition [2,24]. Doshi et al. [5] and Gao et al. [6] have studied the application of Markov Decision Processes (MDPs) in Web service composition. MDPs assume a fully observable world and require explicit reward functions and state transition functions. Such requirements are too strict to a real world scenario. Reinforcement learning (RL) is one type of machine learning methods, concerned with how an agent ought to take actions in an environment so as to maximize some notion of cumulative reward [26]. It emulates the learning behaviors of human and normally follows a trial and error process. Wang et al. [34] proposed to use (RL) for service composition, so as to avoid complex modeling of the real world. Although it has been proved to be effective for small-scale service compositions, it can be computationally expensive when working on a large number of services.

In human learning, people do not learn everything through their own discovery, but rather through exchanging information and learning from the others. This motivates the multi-agent approach, which can be applied to the Web service composition. For example, we can

* Corresponding author. Tel.: +86 2552090861; fax: +86 2552090880.
E-mail addresses: hbw@seu.edu.cn (H. Wang), Xiaojun.Wang@seu.edu.cn (X. Wang), Xingzhi.Zhang@seu.edu.cn (X. Zhang), qi.yu@rit.edu (Q. Yu), Xingguo.Hu@seu.edu.cn (X. Hu).

http://dx.doi.org/10.1016/j.knosys.2015.10.022
0950-7051/

allow two agents to search for an optimal service composition simultaneously. They can share the information about the best services and policies. The resulting parallelism can help boost the efficiency of reinforcement learning. In fact, some studies have already applied the technology to service composition to improve the scalability and flexibility requested by the composition. Based on multi-agent and context awareness, Maamar et al. [15] proposed a novel Web service composition method. It guides all agents to communicate with each other to make the services gain an agreement in a composition process. Although applying multi-agent to service composition demonstrates high efficiency, a method as such cannot satisfy the self-adaptivity needed in the dynamic environment.

In fact, reinforcement learning does not necessarily rely on a single agent to search the complete state–action space to obtain the optimal policy. Considering the multiple requirements in the dynamic environment, such as self-adaptivity, efficiency, coordination and interoperability, combining multi-agent and reinforcement learning technologies can provide better performance in the composition process. By combining the two technologies, we can effectively integrate both of their strengths, which ensure the efficiency and self-adaptivity at the same time.

In this paper, using the concept of reinforcement learning and introducing multi-agent into an adaptive system, we propose a novel mechanism that combines the multi-agent technology and reinforcement learning to achieve adaptive service composition. We model a service composition task as a dynamic environment so that multiple alternative services and workflows can be incorporated into a single service composition for each agent. The mechanism makes adaptive agents have the learning ability, which enables them to learn the environment knowledge and program an efficient adaptive policy to fit the changes of the environment. This learning significantly enhances the adaptivity of a system in a dynamic and uncertain environment.

In sum, our contribution is twofold. Firstly, in order to adapt to the dynamic internet environment and diverse user requirements on QoS performance, we design a Web service composition optimization model based on Markov Decision Process (MDP), which takes semantic web services and workflow technology into consideration. Secondly, we present a distributed Q-learning algorithm for coordinating multi-agent reinforcement learning, which can accelerate the learning rate. The model proposed in this paper extends the reinforcement learning model that has been introduced in [34]. In order to reduce the time of convergence, we introduce a sharing strategy to share the policies among the agents, through which one agent can use the policies explored by the others. Because our algorithm is extended from the traditional Q-learning algorithm, we conduct a simulation study to compare it with the traditional Q-learning algorithm to justify the efficiency of our algorithm. Besides, to further evaluate our method, we carry out a thorough comparison study with other approximate reinforcement learning algorithms, including a multi-agent Sarsa algorithm [30], a Q-learning algorithm based on gaussian process [32], and a multi-agent Q-learning algorithm [33]. The results show the effectiveness of our algorithm as well.

2. Related work

In this section, we give an overview of existing approaches for dynamic Web service composition.

Wang et al. [35] proposed a service composition method in the design stage and the execution stage based on dynamic workflows. It tackles the changes during execution, and also designs a workflow ontology for model reuse in the design stage. However, it requires the definition of the flow structure in advance, thus cannot effectively handle dynamic service composition. Doshi et al. [5] and Gao et al. [6] applied MDPs (Markov Decision Processes) to Web service composition. Doshi et al. developed a policy-based approach for dynamic Web service composition, leading to workflows. The focus is on how to create workflow logic instead of the concrete execution details. In contrast, the work done by Gao et al., is a framework for representing Web service composition using MDP. These methods mainly focus on the structure characteristics of composite services process and using MDP to model different structures. However, as the decision times in each state were ignored, which have also important impact on the success rate of service composition, it may reduce the superiority of these methods in certain situations.

Based on the MDP model, Wang et al. [32] integrated gaussian process with reinforcement learning to address the scalability issue and adaptive requirements in service composition. They utilized kernel function approximation to predict the distribution of the objective function value with strong communication skills and generalization ability based on an off-policy Q-learning algorithm. Through the gaussian process, the off-policy Q-learning algorithm can speed up the convergence rate. This combined algorithm is not only effective for the problem of large scale discrete service composition problem, but also applicable to the service composition problem of continuous state space. Nevertheless, the efficiency in this algorithm may not meet the requirements in service composition with huge state spaces and candidate services.

Reinforcement learning is a formal framework in which an agent interacts with its environment through a series of actions, and receives some rewards as feedback to its actions, but is not told what the correct actions would have been. In standard reinforcement learning, an agent is in a blind exploration condition, which makes target reward spread slowly. Therefore, it is challenging to use reinforcement learning to address complex problems. In [22], through integrating planning and learning into an online algorithm based on the well known Dyna architecture, Santos et al. presented a novel reinforcement learning planning algorithm, Dyna-H. This method incorporates the ability of A^* to focus on specific search subtrees in order to make the search more efficient by taking advantage of the heuristics. This inspired planning approach allows an agent to get rid of blind exploration, thus improve the efficiency.

The reinforcement learning technique in XCS has been based on Q-learning, which optimizes the discounted total reward received by an agent but tends to limit the length of action chains. Therefore, in [40], the authors introduced an undiscounted reinforcement learning methods, R-learning, into XCS to optimize the average reward. The modification results in a classifier system that is rapid and able to solve large-scale problems. Thus, XCS with R-learning is able to maximize the average reward per time step, not the cumulative discounted rewards. It is a promising classifier system to solve big size multi-step problems. However, it is necessary to search more effective estimation methods for average reward to help the system reach an optimal performance.

Sirin et al. [23] presents a semi-automatic method for Web service composition. Each time when a user has to select a Web service, all possible services that match the selected service, will be presented to the user. The choice of the possible services is based on both the functional and non-functional attributes. Although the proposed method is simple, it indicates the trend that an automatic planner and human user can work together to generate the composite service for a user request.

Parejo et al. [19] solved the service composition problem through combining GRASP with Path Relinking. They considered five properties and four composition structures. Since it mainly focuses on service rebinding, this work does not approach the redesign of the workflow itself. In [44] a negative selection immune algorithm (NSA) is applied to solve the web service selection problem, which comes from the negative selection mechanism in biological immune recognition. It was the first effort to apply NSA to web service selection to eliminate improper solutions. However, these methods tend to transform multiple objectives into a single objective through Simple