Learning Workflow Models from Event Logs Using Co-clustering

Xumin Liu, Department of Computer Science, Rochester Institute of Technology, Rochester, NY, USA
Chen Ding, Department of Computer Science, Ryerson University, Toronto, Canada

ABSTRACT

The authors propose a co-clustering approach to extract workflow models by analyzing event logs. They consider two major issues that are overlooked by most of the existing process mining approaches. First, a complex system typically runs multiple workflow models, all of which share the same log system. However, current approaches mainly focus on learning a single workflow model from event logs. Second, most systems support multi-users and each user is typically associated with (or use) certain number of operation sequences, which may follow one or more than one workflow models. Users can thus be leveraged as an important context when learning workflow models. Therefore, the authors propose to learn User Behavior Pattern (UBP) that reflects the usage pattern of a user when accessing a business process system and exploit it to discover multiple workflow models from the event log of a complex system. The authors model a UBP as a probabilistic distribution on sequences, which allows computing the similarity between UBPs and sequences. The authors then co-cluster users and sequences to generate two types of clusters: user clusters that group users sharing similar UBP, and sequence clusters that group sequences that are the instances of the same workflow models. The workflow model can then be learned by analyzing its instances. The authors conducted a comprehensive experimental study to evaluate the effectiveness and efficiency of the proposed approach.

Keywords: Co-Clustering, Probabilistic Suffix Tree, Process Mining, User Behavior Pattern (UBP), Workflow Model Discovery

1. INTRODUCTION

Process mining has attracted great attention in recent years as it is considered as an important component of business intelligence (van der Aalst, 2012). The idea of process mining is to automatically learn workflow models through the analysis of event logs. This is motivated by the observation that, although users are assumed to follow the design of workflow models when using a business process system, a workflow instance may deviate from the model in various scales, depending on the degree of flexibility allowed by the system. The deviation is caused by many factors, such as exceptional situations, errors, unexpected events, personalized requirements, and randomness of user behaviors. Meanwhile, leveraging service-oriented...
architecture for the development of business processes significantly increases the flexibility and personalization of business processes (Bouguettaya, 2010, Liu, 2011). To better study user requirement and improve the design of a business process system, process mining approaches analyze event logs and reconstruct the workflow models from the logs, which can be used to adjust or evolve workflow models, improve user interfaces, improve resource allocations, and so on (van der Aalst, 2012).

Driven by the benefits brought by business intelligence, there is a considerable amount of work conducted in the field of process mining, a.k.a. workflow mining (de Medeiros, 2004, Greco, 2006, Kim, 2007, Tang, 2010). These approaches usually take event logs as their input. Despite the various formats of event logs, process mining approaches assume that the following information can be found in the logs: session id, timestamp, workflow id, user id, and operation id. The logs are first processed to generate a trace of the user’s operations after entering and before leaving the system, i.e., in a session. Machine learning techniques are then used to mine the temporal relationship between each pair of operations in the same workflow model. These process mining approaches differ mostly in the way of modeling processes (e.g., Petri-Net workflows, finite state machine, Hidden Markov Model), the targeting problems (e.g., iterations of operations, hierarchical structure of workflow processes, noises), and mining approaches (e.g., sequence mining, EM algorithms, neural networks). A comprehensive survey of the existing process mining techniques can be found in (van der Aalst, 2004). However, these techniques are limited in two aspects, which are described as follows.

First, the existing approaches assume that the instances of a workflow are well organized and the link between a sequence, i.e., a workflow instance, and the workflow model is available. Specifically, some approaches analyze the entire event log and try to learn a single workflow model out of them. That is, all the sequences are treated as the instances of the same workflow model. Others allow multiple workflow models but simply assume that workflow model ids are recorded, which can be used to group log records that are related to the same workflow models. Assuming a single workflow model is obviously not suitable for a complex system, which runs multiple workflow models (Greco, 2006, Ferreira, 2007). Furthermore, modern software systems have been increasingly designed to provide great convenience and flexibility for their users, which makes it difficult or even impossible to connect the log records of diverse users to a small set of predefined workflow models. As usability becomes the major concern when deciding the layout of the user interface, it becomes common that users access the same system to perform different types of business processes by using different workflow models. For example, the Amazon website supports multiple types of business processes, such as purchasing products, requesting for returning items, managing subscriptions, and creating baby registry lists. Users can invoke any operation whenever it is accessible regardless of the underlying workflow models as long as the invocation does not conflict with any predefined business constraint. Such a constraint may be that a user should input the payment method before placing an order. Therefore, before analyzing an event log and learning the temporal orders between operations, log records should be organized in a way, so that the ones associated with the same workflow models should be grouped together.

Second, the existing approaches ignore the user factor during the workflow model learning process. Specifically, the user factor may include business requirement of the user along with his/her personal preference and/or constraint. The user factor affects the usage pattern of the system by the user, which we coin as User Behavior Pattern (or UBP). The UBP of a user can be reflected by the types of business processes, i.e., workflow models, the user has performed, and the corresponding frequencies. For example, a frequent buyer’s UBP would be very different from that of a frequent seller as the former usually purchases products while the latter frequently updates product inventories.
Analytics-as-a-Service (AaaS): An Elucidation to SOA
www.igi-global.com/chapter/analytics-as-a-service-aaas/217886?camid=4v1a

Aspect-Oriented Framework for Web Services (AoF4WS): Introduction and Two Example Case Studies
www.igi-global.com/chapter/aspect-oriented-framework-web-services/26082?camid=4v1a