Discovering Patterns using Process Mining

Ishak Meddah, LAMOSI Laboratory, USTO-MB University, Oran, Algeria
Belkadi Khaled, LAMOSI Laboratory, USTO-MB University, Oran, Algeria

ABSTRACT

Process mining provides an important bridge between data mining and business process analysis, his techniques allow for extracting information from event logs. In general, there are two steps in process mining, correlation definition or discovery and then process inference or composition. Firstly, the authors’ work consists to mine small patterns from a log traces of two applications; SKYPE, and VIBER, those patterns are the representation of the execution traces of a business process. In this step, the authors use existing techniques; The patterns are represented by finite state automaton or their regular expression; The final model is the combination of only two types of small patterns whom are represented by the regular expressions (ab)* and (ab*c)*. Secondly, the authors compute these patterns in parallel, and then combine those small patterns using the composition rules, they have two parties the first is the mine, they discover patterns from execution traces and the second is the combination of these small patterns. The patterns mining and the composition is illustrated by the automaton existing techniques. The Execution traces are the different actions effected by users in the SKYPE and VIBER. The results are general and precise. It minimizes the execution time and the loss of information.

KEYWORDS

Finite State Automata, Log File, Process Mining, Traces

1. INTRODUCTION

Many techniques have been proposed that mine patterns from execution traces. However; existing techniques mine only simple patterns, or a single complex pattern that is limited to a particular set of manually selected events.

Authors [Ammons et al., 2002] have recognized that patterns can be specified as regular languages, that allows the compact representation of patterns as regular expressions or finite state automata, and it allows the characterization of the pattern mining as a language learning problem.

Process mining approaches are fundamentally similar; each takes as input a static program or a dynamic traces or profile and produces one or more compact regular languages that specify the pattern representation or the workflow. However; the individual solutions differ in key ways.

In this paper, we present a new general approach to patterns mining that addresses several of the limitations of current techniques. Our insight is twofold. First, we recognize that instances of smaller patterns can be composed in parallel into larger patterns. Second, we observed also that the composition of small pattern can be in parallel.

We then leverage this insight to divide our work into two parties; The first one, we use a technique how we can mine two types of small patterns and we compose them by using standard algorithms for finite state automaton manipulation, and some special rules using by Gabel and Su (Gabel, & Su, 2021).
2008a; Ammons et al., 2002), the mining is also performed by symbolic mining algorithm (Gabel & Su, 2008b; Gao, 2013; Guyet, & Quiniou, 2011).

Our approach is an amelioration of existing work in pattern mining, it has been implemented in the java programming language with more log files of two applications; the SKYPE and VIBER applications. The size of those applications log files is 10 GB, 15 GB, 18 GB, and 20 GB which are generated by log file generator.

The executions traces in our applications are the user’s actions like call, the answer, and the messages.

2. RELATED WORK

Many techniques are suggested in the domain of process mining, we quote:

Gabel et al. (Gabel & Su, 2008a) present a new general technique for mining temporal specification, they realized their work in two steps; firstly they discovered the simple patterns using existing techniques, then combine these patterns using the composition and some rules like Branching and Sequencing rules.

Temporal specification expresses formal correctness requirement of an application’s ordering of specific actions and events during execution, they discovered patterns from traces of execution or program source code; The simples patterns are represented using regular expression (ab)* or (ab*c)* and their representation using finite state automaton, after they combine simple patterns to construct a temporal specification using a finite state automaton.

Greco et al (Greco et al., 2006) discovered several clusters by using a clustering technique, and then they calculate the pattern from each cluster, they combine these patterns to construct a final model, they discovered a workflow scheme from, and then they mine a workflow using a MineWorkflow Algorithm, after they define many clusters from a log traces by using clustering technique and Process Discover Algorithm and some rules cluster.

Then they use a Find Features Algorithm to find a patterns of each cluster, finally they combine these patterns to construct a completely hierarchical workflow model.

In their clustering algorithm, clusters reflect only structural similarities among traces; they say that in future works extending their techniques to take care of the environment so that clusters may reflect not only structural similarities among traces, but also information about, e.g., users and data values.

Motahari-Nezhed et al. (Motahari-Nezhad, et al., 2008) use a service conversation log; first they split a log into several partitions, 2nd they discovered a model from each partition, and 3rd, they annotate the discover protocol model with various metadata to construct a protocol model from real-word service conversation logs.

The protocol is the specification of all possible conversations that a service can have with its partners and the conversation consists of a sequence of messages exchanged between two or more services.

During the split they discovered a simple precise protocol models by analyzing messages sequences in the log, they eliminate conversations considered noisy or not presented in the log; they augmently discovered a protocol with various metadata including state and transition supports to get a final protocol model of the log a most generalized model based splitting.

“Mining frequent patterns and finding associations among them require handling large and distributed databases. As FP-tree considered being the best compact data structure to hold the data patterns in memory there has been efforts to make it parallel and distributed to handle large databases” (Itkar, & Kulkarni, 2013). However, it incurs lot of communication over head during the mining. Parallel and distributed frequent pattern mining algorithm using Hadoop Map Reduce framework is
Hybrid Clustering using Elitist Teaching Learning-Based Optimization: An Improved Hybrid Approach of TLBO
*International Journal of Rough Sets and Data Analysis* (pp. 1-19).
[www.igi-global.com/article/hybrid-clustering-using-elitist-teaching-learning-based-optimization/144703?camid=4v1a](www.igi-global.com/article/hybrid-clustering-using-elitist-teaching-learning-based-optimization/144703?camid=4v1a)

Security in Border Gateway Protocol (BGP)
[www.igi-global.com/chapter/security-in-border-gateway-protocol-bgp/113161?camid=4v1a](www.igi-global.com/chapter/security-in-border-gateway-protocol-bgp/113161?camid=4v1a)