International Journal of Science and Engineering Applications
Volume 10-Issue 09, 144 - 147, 2021, ISSN:- 2319 - 7560

A Sampling Approach based on Set Coverage Algorithm

Huiling LI
School of Computer Science and Technology, Shandong University of Technology, Zibo, 255000, China

Xuan SU
School of Computer Science and Technology, Shandong University of Technology, Zibo, 255000, China

Shuaipeng ZHANG
School of Computer Science and Technology, Shandong University of Technology, Zibo, 255000, China

Abstract: Massive amounts of business process event logs are collected and stored by modern information systems. Model discovery aims to discover a process model from such event logs, however, most of the existing approaches still suffer from low efficiency when facing large-scale event logs. Event log sampling techniques provide an effective scheme to improve the efficiency of process discovery, but the existing techniques still cannot guarantee the quality of model mining. Therefore, a sampling approach based on set coverage algorithm named set coverage sampling approach is proposed. The proposed sampling approach has been implemented in the open-source process mining toolkit *ProM*. Furthermore, experiments using a real event log data set from conformance checking and time performance analysis show that the proposed event log sampling approach can greatly improve the efficiency of log sampling on the premise of ensuring the quality of model mining.

Keywords: event logs; log sampling; quality measure; set coverage; conformance checking

1. INTRODUCTION

Process mining[1-3] is a novel discipline that connects data science and business process management. It aims to extract effective information about business processes from event logs and discover, monitor and improve real business processes[4]. Process mining also includes sub-areas such as process prediction [5]-[6] and business process automation [7]. Process discovery is one of the most challenging process mining tasks, which allows the discovery of process models from event logs without any prior information. In recent years, it has received extensive attention. Over the past two decades, domestic and foreign researchers have proposed various process discovery methods, for example, *Alpha Miner*[8], *Heuristics Miner*[9], *Heuristics Miner*[10], *Tsinghua Alpha* [11], *Split Miner*[12], etc. But most discovery methods are no longer suitable for using a single machine to process an entire large data set. With distributed platforms such as the well-known *MapReduce framework* [13]-[14], the process can be very time consuming, so a new approach is urgently needed to address these issues.

The event log sampling approach provides a feasible solution to the above problem. It takes the original event log as input and returns a sample log. At present, many event log sampling approaches have been proposed, such as an event log sampling approach based on graph sorting algorithm named *LogRank*[15]-[16] and an event log sampling approach based on trajectory similarity named *LogRank+*[17]. However, their performance still cannot meet the needs of practical application, for example, the quality of the model is still not ideal, meanwhile, with the increase of the original log size, the difference between the sum of the original log sampling time and the sample log mining time and the original log mining time becomes more and more obvious.

Inspired by the traditional set coverage and other related ideas, we propose set coverage sampling approach. Compared with the existing sampling methods, the set coverage sampling approach proposed in this paper can obtain simpler and higher quality process models. In addition, in order to verify the feasibility and efficiency of the four sampling approaches of event logs, related experiments are done from the aspect of conformance checking and time performance analysis. The quality of sample logs compared with the original logs can be obtained by the experimental results.

The remainder of this paper is organized as follows. Section 2 discusses the related work. Section 3 introduces set coverage sampling approach. Section 4 describes the tool implementation. Section 5 describes the data set used in the experiments, introduces the experiments and shows the results of the evaluation. Finally, Section 6 draws conclusions and points our future research scope.

2. PRELIMINARIES

Let S be a set. We use $|S|$ to denote the number of elements in set S. $B(S)$ is the set of all multisets over set S. $f \in X \rightarrow Y$ is a function, i.e., $dom(f)$ is the domain and $rng(f) = \{f(x) \mid x \in dom(f)\}$ is the range.

Definition 1 (Event, Trace, Event Log). Let A be a set of activities. A trace $\sigma \in A^*$ is a sequence of activities (also referred to as events). For $1 \leq i \leq |\sigma|$, $\sigma(i)$ represents the ith event of σ. $L \in B(A^*)$ is an event log.

An event log records the execution of a potential business process whose business process model is the task target of process mining, so it does not appear explicitly in the definition of the event log. The execution of a business process instance is represented by the corresponding traces. The events in the trace are recorded in the event log.

Definition 2 (Process Discovery). Let UM be the set of all process models, a process discovery method is a function γ mapped from an event log $L \in B(A^*)$ to a process model $pm \in UM$, i.e., $\gamma(L) = PM$. In general, the process discovery method can transform the event log into a process model represented by marked *Petri nets*, *BPMN*, *EPC*, etc. Regardless of the representation used by the process model, each trace in the input event log corresponds to a possible execution sequence in the discovered process model.

Definition 3 (Directly Follows Relation). Let a and $b \in A$ be two activities and $\sigma = \langle \sigma_1, .., \sigma_n \rangle$ is a trace in the event log. A directly follows relation from a to b exists in trace σ, if there is $i \in \{1, ..,$

n−1} such that $\sigma_i = a$ and $\sigma_{i+1} = b$ and we denote it by $a >_\sigma b$. For example, in $\sigma = \langle a, b, c, e, g \rangle$, we have $c >_\sigma e$, but $d \not>_\sigma a$.

Definition 4 (Start point set). The start event of each trace in the event log constitutes the start point set.

Definition 5 (End point set). The end event of each trace in the event log constitutes the end point set.

3. SAMPLING APPROACH

In theory, we can choose an arbitrary subset of the trace from the event log as its sample log, while the real challenge is to find sample logs that are representative enough so that a reliable process model can be found compared to the original event log. In response to this challenge, this paper propose a sampling approach based on set coverage algorithm named set coverage sampling approach. This sampling approach can get the sample log that is a representative subset of the original log. It can reduce the computational cost. At the same time, compared with the existing event log sampling approaches, set coverage sampling approach can not only ensure the quality of the process model mined from the sample logs, but also greatly shorten the sampling time and mining time and improve the efficiency of process discovery.

The set coverage sampling approach is mainly based on the greedy algorithm to solve the set coverage problem, so the idea of the Set coverage sampling approach is as follows: Input the original event log in the platform, firstly, the directly follows relation of all traces in event log are traversed. If the trace's directly follows relation has the biggest intersection with the log's directly follows relation, meanwhile this intersection is not empty, or the trace's start point has an intersection with the log's start point set, or the trace's end point has an intersection with the log's end point set, then put this trace into the sample log. Finally, delete the following three parts: (1) The intersection of the log's directly follows relation set and the trace's directly follows relation set in the trace's directly follows relation set; (2)The intersection of the start point set and the trace's start point; (3) The intersection of the end point set and the trace's end point. Trace traversal is stopped until the log's directly follows relation set, start point set and end point set are all empty. In the end, the platform outputs a sample log.

4. TOOL IMPLEMENTATION

In this experiment, we use a laptop with a 2.70 *GHz CPU*, *Windows* 10 Professional, *Java SE* 1.8.0_281 (64-bit) , Python 3.7.6 (64-bit) and allocate 12 *GB* of *RAM*. In addition, the drawing software *Origin* 2021 *Pro* version is used to show the experimental results.

The open source process mining tool platform *ProM* provides a fully pluggable experimental environment for process mining. It can be extended by adding plugins and currently contains more than 1600 plugins. The tool and all plugins are open source. Set coverage sampling approach proposed in this paper has been implemented in *ProM* platform as plugin, which called *Business Process Event Log Sampling Plugin*. The snapshot of this tool is shown in Figures 1. It takes an original event log as input and outputs a sample log when the sampling approach is selected.

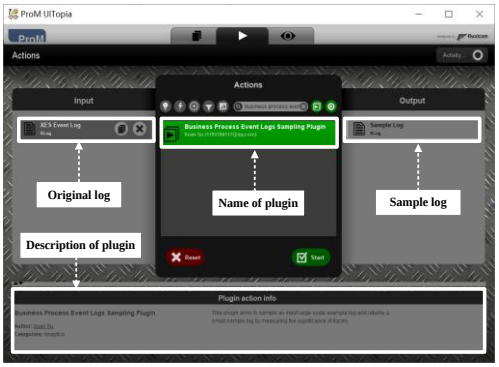

Figure. 1 The instance of *ProM* plugin

In the conformance checking experiment to verify the effectiveness of the set coverage sampling approach, the plugin called *Replay a Log on Petri Net for Conformance Analysis* implemented in *ProM* as shown in Figure 2 is used. It takes original event log and the process model mining from the sample log as input.

Figure. 2 Plugin for fitness index

5. EXPERIMENTAL EVALUATION

5.1 Experimental data sets

In this experiment, a real event log data set is used to evaluate the proposed set coverage sampling approach. Table 1 details some major statistics of this event log, including the trace number, event number and activity number and so on.

Table 1. Major statistics of event logs

Event log	Trace number	Variant number	Event number	Activity number	Trace length		
					Minimum value	Average value	Maximum value
BPIC_2012_A	13087	32	146044	20	6	11	20

·**BPIC_2012_A data set:** This data set is a real-life log, taken from a Dutch Financial Institute. Apart from some anonymization, the log contains all data as it came from the financial institute. The process represented in the event log is an application process for a personal loan or overdraft within a global financing organization. The amount requested by the customer is indicated in the case attribute AMOUNT_REQ, which is global, i.e. every case contains this attribute. The event log is a merger of three intertwined sub processes. The first letter of each task name identifies from which sub process (source) it originated from. Feel free to run analyses on the process as a whole, on selections of the whole process and/or the individual sub processes.

5.2 Conformance checking

To verify the availability of the set coverage sampling approach, we measure it in terms of conformance checking. The conformance checking experiment associates events in the event log with activities in the process model and compares them. The goal of it is to find commonalities and differences between the modeled behavior and the inspected behavior. In this experiment, we use fitness degree as quality standard, which is the measure related to conformance. Firstly, the process model of sample log is mined by using the version of *IM* algorithm with noise threshold of 0.9, then the process model and the original event log are measured for fitness degree. The experimental results are shown in Figure 3.

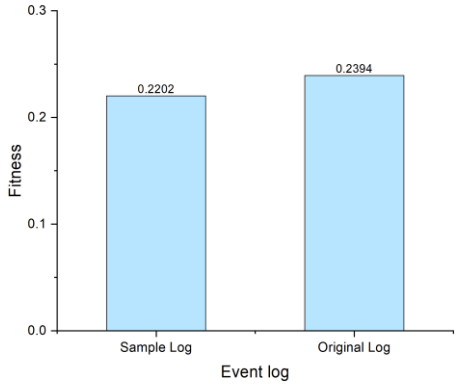

Figure. 3 The result of conformance checking

The experimental results show that the fitness of sample log obtained by set coverage sampling approaches is very closely to the fitness of original event log. It is proved that this set coverage sampling approach can extract sufficiently representative sample logs to a large extent and further prove its availability.

5.3 Time performance analysis

Time performance analysis experiment measures and records three types of time: (1) the original event logs' mining time; (2) the sampling time by using four sampling methods; (3) sample logs' mining time. Due to the computer internal environment every time may be different, so in order to guarantee the accuracy of experimental results, we measure each data for 5 times to get the average. Finally, the sampling time of each of the set coverage sampling approach is summed up with the sample log mining time, then compare with the original logs' mining time. The experimental result is shown in Figure 4.

The experimental results show that compared with the original event log, the sample log obtained by set coverage sampling approach can use less time to mining process models when the log's scale is large. Meanwhile, with the increase of the event log's scale, the difference between them becomes more and more obvious. It can be proved that the operation efficiency can be greatly improved by using sample log instead of original log, meanwhile the set coverage sampling approach of event log have high efficiency.

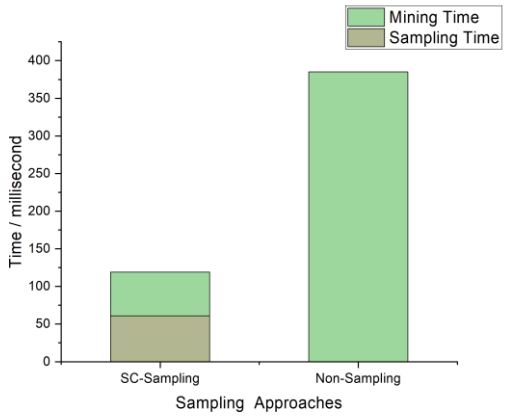

Figure. 4 The result of time performance analysis

6. CONCLUSIONS

In this paper, we propose set coverage sampling approach to effectively obtain sample logs with sufficient representability in large-scale event logs. Meanwhile we implemented set coverage sampling approach in the *ProM* platform. In addition, we assess the quality of the sample logs relative to the original logs in terms of conformance checking and time performance analysis. At the end, the experimental results on a real event log data set shows that compared with the existing sampling approaches, the proposed set coverage approach can not only greatly improve the efficiency of log sampling, but also ensure the integrity of the model.

As future work, we aim to apply the set coverage sampling approaches proposed in this paper to the event log for specific fields, such as education, medical care, finance, manufacturing, etc. It is also valuable to study the deployment of set coverage sampling approach in the distributed system because it is convenient to process the super-large event logs collected by other information systems in real life.

7. ACKNOWLEDGMENTS

This work was supported in part by National Natural Science Foundation of China under Grant 61902222, the Taishan Scholars Program of Shandong Province under Grant ts20190936 and Grant tsqn201909109, and Engineering and Technology R&D Center of IIOT in Colleges of Shandong Province (QingDao Technical College, Grant KF2019002).

REFERENCES

[1] VAN DER AALST W. Data science in action[M]//Process mining. Berlin, Germany: Springer-Verlag, 2016: 3-23.

[2] ZENG Q, SUN S X, DUAN H, et al. Cross-organizational collaborative workflow mining from a multi-source log[J]. Decision support systems, 2013, 54(3): 1280-1301.

[3] LIU C, DUAN H, ZENG Q, et al. Towards comprehensive support for privacy preservation cross-organization business process mining[J]. IEEE Transactions on Services Computing, 2019,12(4):639-653.

[4] VAN DER AALST W. Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer-Verlag, Berlin, 2011.

[5] POURBAFRANI M, VAN ZELST S J, VAN DER AALST W M P. Scenario-based prediction of business

processes using system dynamics[C]//OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". Berlin, Germany: Springer-Verlag, 2019: 422-439.

[6] QAFARI M, VAN DER AALST W. Fairness-aware process mining[C]//OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". Berlin: Springer, 2019: 182-192

[7] GAO J, VAN ZELST S J, LU X, et al. Automated robotic process automation: A self-learning approach[C]//OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". Berlin, Germany: Springer-Verlag, 2019: 95-112.

[8] VAN DER AALST W, WEIJTERS T, MARUSTER L. Workflow mining: Discovering process models from event logs[J]. IEEE transactions on knowledge and data engineering, 2004, 16(9): 1128-1142.

[9] WEIJTERS A, RIBEIRO J T S. Flexible heuristics miner (FHM)[C]//2011 IEEE symposium on computational intelligence and data mining (CIDM). Washington, D. C., USA: IEEE, 2011: 310-317.

[10] LEEMANS S J J, FAHLAND D, VAN DER AALST W M P. Discovering block-structured process models from event logs-a constructive approach[C]//International conference on applications and theory of Petri nets and concurrency. Berlin, Germany: Springer-Verlag, 2013: 311-329.

[11] WEN L, WANG J, W.M.P. VAN DER AALST W, et al. A novel approach for process mining based on event types. Journal of Intelligent Information Systems, 32(2): 163-190, 2009.

[12] AUGUSTO A, CONFORTI R, DUMAS M, and ROSA M. Split miner: automated discovery of accurate and simple business process models from event logs. Knowledge and Information Systems, 1-34, 2018.

[13] CHENG L, LI T. Efficient data redistribution to speed up big data analytics in large systems[C]//2016 IEEE 23rd International Conference on High Performance Computing (HiPC). Washington, D. C., USA: IEEE, 2016: 91-100.

[14] Evermann J. Scalable process discovery using map-reduce[J]. IEEE Transactions on Services Computing, 2014, 9(3): 469-481.

[15] LIU C, PEI Y, ZENG Q, et al. LogRank: An approach to sample business process event log for efficient discovery[C]//International Conference on Knowledge Science, Engineering and Management. Berlin, Germany: Springer-Verlag, 2018: 415-425.

[16] LIU C, PEI Y, CHENG L , et al. Sampling business process event logs using graph-based ranking model[J]. Concurrency and Computation: Practice and Experience, 33(5):1-14, 2021.

[17] LIU C, PEI Y, ZENG Q, et al. LogRank+: A Novel Approach to Support Business Process Event Log Sampling[C]//International Conference on Web Information Systems Engineering. Berlin: Springer, 2020: 417-430.