International Journal of Science and Engineering Applications
Volume 10-Issue 09, 116 - 121, 2021, ISSN:- 2319 - 7560

Trace Clustering: A Preprocessing Method to Improve the Performance of Process Discovery

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

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

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

Abstract: The information system collects a large number of business process event logs, and process discovery aims to discover process models from the event logs. Many process discovery methods have been proposed, but most of them still have problems when processing event logs, such as low mining efficiency and poor process model quality. The trace clustering method allows to decompose original log to effectively solve these problems. There are many existing trace clustering methods, such as clustering based on *vector space approaches*, *context-aware trace clustering*, *model-based sequence clustering*, etc. The clustering effects obtained by different trace clustering methods are often different. Therefore, this paper proposes a preprocessing method to improve the performance of process discovery, called as trace clustering. Firstly, the event log is decomposed into a set of sub-logs by trace clustering method, Secondly, the sub-logs generate process models respectively by the process mining method. The experimental analysis on the datasets shows that the method proposed not only effectively improves the time performance of process discovery, but also improves the quality of the process model.

Keywords: process discovery; trace clustering; process model; log similarity; quality measure

1. INTRODUCTION

Process mining [1-3] aims to extract effective information about business processes from event logs to discover, monitor and improve actual processes. Process mining mainly includes: 1) Process discovery takes the event log as input to the automatic production process model; 2) Conformance checking can be used to check if reality, as recorded in the log, conforms to the model and vice versa; 3) Enhancement is to extend or improve an existing process model using information about the actual process recorded in some event log. In addition, process mining also includes process prediction [4-5] and business process automation [6]. Process discovery is one of the most challenging process mining tasks, aims to discover a business process model form an event log. In the past two decades, researchers have proposed various process discovery approaches, e.g, Alpha Miner [7], Heuristic Miner [8], Inductive Miner [9], etc.

However, with the advent of the era of big data, business processes produce larger and more complex event logs. For these event logs, most existing process discovery approaches unable to mine the information correctly, and usually lead to process discovery low efficiency. In the process mining manifesto [10], Professor Van der Aalst and others take that existing process mining methods are difficult to handle the massive amounts of data is generated ASML's wafer scanner. as an example, therefore, dealing with large-scale and complex event log problems is one of the important challenges of process mining.

When dealing with complex and large-scale event log, the event log is reasonably decomposed into several sub-logs, and then the sub-logs are discovered by the existing process discovery approaches, thereby improving the efficiency of process discovery and the quality of process models. An effective way to decompose the event log is to cluster the trace in the event log, so that the process model combination corresponding to the clustered sub-logs can clearly and completely express the behavior in the original event log. On the one hand, the preprocessing operation of trace clustering can effectively improve the time performance of the process discovery method, and on the other hand, it also reduces the probability of complex process models (similar to the spaghetti process model), and then more intuitively understand the process model. To this end, we propose a preprocessing method to improve the performance of process discovery, called as trace clustering. The sub-logs by the trace clustering methods are mined by the existing process discovery approaches to generate the sub-process models. Finally, Checking the conformance of the above sub-logs with the original log by measuring fitness, precision, F-Measure to verify the feasibility and efficiency of the trace clustering preprocessing operation.

2. RELATED WORK

2.1 History of Process Mining Algorithms

In 2002, Wil van der Aalst proposed the *Alpha Miner* in [7]. From the perspective of workflow, it is based on the direct follow activity relationship between logs to mine the activity associations in event logs. The disadvantage of Alpha Miner is that it unable to flexibly handle noise, incomplete event logs, and cannot identify short loops, map non-local dependencies, and handle non-free choice structures. Many researchers have devoted themselves to improving and extending the *Alpha Miner*, and different algorithms have been proposed to solve these limitations.

For this reason, Weijters & van der Aalst et al. (2003) extended the Alpha Miner in [8] and considered the frequency of directly follow activity relationship, and calculated the dependency/frequency parameter to obtain the heuristic network. The algorithm is called *Heuristic Miner*. It can handle noise and allows comparison between manually designed models and execution processes. This algorithm is the most commonly used and customized because it

guarantees good adaptability, but it cannot provide complete reliability because uncommon paths are not incorporated into the model.

Jansen-Vullers et al. (2006) created a new algorithm based on integer programming technology. It shows that it is possible to search for the best settings using objective functions and applying integer programming techniques. This method finds all the solutions of a system of equations, and implements a minimization function through integer programming techniques.

Leemans et al. (2013) proposed an extensible framework called *Inductive Miner*[9]. The purpose of the algorithm is to discover block-structured process model that is reasonable and suitable for the behavior observed on the event log. This algorithm represents the minimum information of the discovery process model. Inductive Miner provided polynomial time complexity and feasible computational cost.

In 2017, vanden Broucke and Weerdt extended the most popular Heuristic algorithm and proposed the *Fodina Miner*[11]. This method is robust to noise and can identify repetitive activities. In addition, the algorithm is flexible, allowing users to choose to adjust the discovery process.

2.2 Quality Evaluation Index

This article uses the following three indicators to evaluate the quality of the event log, where L represents the event log and M represents the process model.

Index 1-Fitness

Fitness[12] quantifies the degree to which the process model can accurately reproduce the trace recorded in the event log, and it quantifies the ability of the process model to regenerate the trace recorded in the event log. A degree of fitness of 1 means that the process model can regenerate all trace in the event log, and a low degree of fitness indicates that most of the behaviors in the event log cannot be reproduced by the process model;

Index 2-Precision

Precision[13] quantification of some behaviors that can be repeated in the process model but not seen in the event log. It measures the ability of the process model to only generate traces in the event log. A precision of 1 means that all traces generated by the process model are included in the event log, and low precision means that the process model allows more behavior than the event log.

Index 3-F-measure

The F-measure value[14] is defined as the harmonic mean value of fitness and precision, calculated as follows:

$$F\text{-}measure(L,M) = \frac{2 \times fitness(L,M) \times precision(L,M)}{fitness(L,M) + precision(L,M)}$$

Where *fitness* (*L*, *M*) is the degree of fitness of the process model found in the event log relative to the original log, and *precision* (*L*, *M*) is the precision of the process model found in the sample log relative to the original log.

3. Framework

This paper proposes a process mining algorithm based on trace clustering. On the basis of the existing process mining algorithm, the log is preprocessed for trace clustering operation, and then the clustered sub-logs are respectively applied to the existing process mining algorithm performs. Finally, evaluates the obtained process model. Fig.1 shows an overview of our approach.

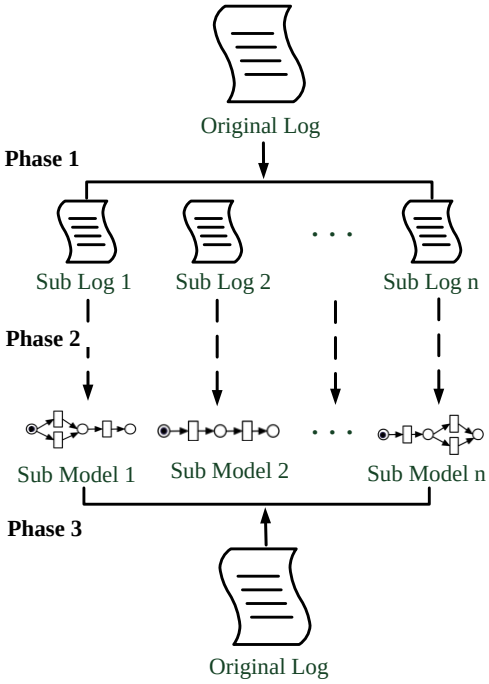

Figure 1. An Approach Overview

Phase 1: Preprocessing based on trace clustering

There are many existing trace clustering approaches, such as K-means trace clustering, active learning clustering, etc. After the original log is processed by the trace clustering method, a set of sub-logs are obtained, so that they belong to the same sub-log The traces the same sub-log of are similar, and the traces belonging to different sub-logs are different. The trace clustering method used in this process requires preset parameters, such as the number of clusters, and setting different parameters may affect the final log quality.

Phase 2: Event log process discovery

There are many existing process discovery methods, such as Alpha Miner, Heuristic Miner, and Inductive Miner. According to the event log input by the user, these mining algorithms are used to obtain the corresponding process model. It is worth noting that the parameter settings of the process mining algorithm may result in different process models, and the default parameter settings are used in this article.

Phase 3: Process model quality assessment

The feasibility and efficiency of the method proposed in this article can be evaluated from the following two perspectives.

(1) Process model quality: In order to quantify the quality of the process model, we first process the original log into a set of sub-logs from the trace clustering method by the existing process discovery methods for each sub-log to obtain the corresponding sub-process models, and separate the sub-process model from the original log checking conformance to measure fitness, precision, and F-Measure to quantify the quality of the new process model. By comparing the quality of the process model

with the original log, the feasibility of the method proposed in this article is demonstrated;

(2) Process discovery efficiency: The efficiency of process discovery can be quantified by the time it takes to obtain the process model. The less time it takes to obtain the process model, the higher the efficiency of process discovery. The efficiency of the method proposed in this paper is demonstrated by comparing the time it takes to obtain the process model.

4. TRACE CLUSTERING METHOD

4.1 Vector Space Method

Song et al. [15] proposed a method to construct a vector space model for the trace in the event log. This method is based on a set of configuration files, each of which measures multiple characteristics of each trace from a specific angle, such as activities, directly follow relation, etc., these features can form a corresponding feature matrix. Based on the feature matrix, multiple distance metrics (Euclidean distance, etc.) are used to calculate the distance between any two traces in the event log. Finally, the traditional clustering algorithms such as K-means clustering is applied in data mining to group the traces in event logs into sub-logs.

4.2 Context-aware Trace Clustering

Bose and van der Aalst described this trace clustering technique in [16,17], which extends the previous trace clustering method by improving the context awareness of control flow. The context awareness here refers to the control flow attributes of the trace in the event log, rather than context information such as organizer, case data, etc. In [16], Bose and van der Aalst proposed a general edit distance technique based on Levenshtein[18], in which editing operations include insertion, deletion or replacement. In [17], the idea of context-aware trace clustering was further developed, and the idea of generating a vector space model for the traces in the event log was reconsidered, using conservative patterns or subsequences to replace the previous activities as the basis of the vector space model. In this way, the concepts of maximum, supermax, and near-supermax repetition are defined to create a feature set that determines a certain trace vector. The corresponding trace clustering method in this article is Guide Miner Tree trace clustering.

4.3 Model-based Sequence Clustering

Ferreira et al. [19] proposed a trace clustering that is different from previous methods. Inspired by the work of Cadez et al. [20] in the field of Web usage mining, they proposed to cluster traces by learning a hybrid first-order Markov model using an expectation maximization (EM) algorithm. In [21], this model-based trace clustering technique was applied to server logs, proving its availability in real life.

De Weerdt et al. proposed in [22] the problem of finding the optimal distribution of traces on a given number of clusters, so as to maximize the combined accuracy of the associated process model. This method changes the goal of traditional trace clustering, which is based on grouping similar traces to find the optimal distribution and solves the problem of finding the optimal trace distribution. It proposes a top-down greedy algorithm and a standard for trace selection. Not because they exhibit similar behavior, but because they fit a particular process model well. The corresponding trace clustering method in this article is *ActiTrac* trace clustering.

5. EXPERIMENT ANALYSIS

5.1 Experimental Environment Settings

The open source process mining tool platform ProM (see http://www.promtools.org/) provides a fully pluggable experimental environment for process mining. It can be extended by adding plug-ins, currently contains more than 1,600 plug-ins, the tool and all plug-ins are open source.

The experiments in this article are all based on PC Intel Core i5-4210M 2.60GHz CPU, 12GB RAM environment, using Java language.

5.2 Simulation Data Structure

This article uses WoPed simulation tool (see as https://woped.dhbw-karlsruhe.de/) to construct a Petri net model. The model is constructed as follows, and then a jar package is generated from the log to generate a simulation Log. It contains 206 traces, 3228 events and 20 activities. The process model is shown in Figure 2.

The Petri net in Figure 2 is designed to generate different event logs with three types of behavior: a trace with activity X, a trace with activity Y, and a trace with neither activity X nor activity Y. Please note that it has chosen to include a large number of parallel and circular behaviors to approximate the complexity of a real event log.

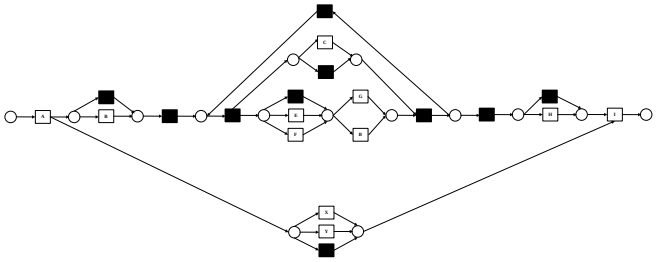

Figure 2. Example Petri net used for generating different classes of behavior according to the presence/absence of activities X and Y.

5.3 Simulation Log Analysis

We use four classic process discovery methods (Alpha Miner, Heuristic Miner, Inductive Miner,Fodina Miner) and three representative trace clustering methods(ArciTrac Trace Clustering,Guide Tree Miner,K-Means) to analyze the feasibility and accuracy of the method proposed in this article from the following two aspects.

5.3.1 Time Performance Analysis

By comparing the time taken by the original log to obtain the process model using each process mining method and the total time used by the process model obtained by the clustered sub-logs through each process mining method, it shows that the trajectory clustering method improves the process mining to a certain extent. The time of the method, the results obtained are shown in Table 1.

It can be seen from Table 1 that for most mining algorithms, the sum of the sub-logs after trajectory clustering is mostly less than the time to mine the original log directly through the process discovery method, which shows that the trace clustering process Later, the time performance of the process

Table 1. Process discovery algorithm time performance comparison(ms)

Trace Clustering Method	Number of clusters	Process discovery algorithm			
		Alpha Miner	Heuristic Miner	Inductive Miner	Fodina Miner
OriginalLog		31	90	153	35
Arctrac	3	32	24	55	30
	4	19	41	31	18
	5	26	49	39	10
Guide Tree Miner	3	19	24	133	23
	4	26	49	39	28
	5	29	63	26	19
K-means	3	19	44	29	30
	4	25	55	57	12
	5	29	43	42	20

discovery method has been further improved. In fact, if the processed sub-logs are processed on a distributed platform, the time performance will be further improved.

It is worth noting that this article does not compare the processing time of trace clustering. The reason is that this article only discusses whether the effect of clustering has further improved the process discovery method. In addition, this article also compares the trace clustering time statistics are performed, as shown in Table 2. From Table 2, it can be seen that the processing time of trace clustering is to a certain extent far longer than the time used for process discovery. The time of different trace clustering is different, which is due to the different operations in different trace clustering method caused.

Table 2. Trace clustering preprocessing time of log (ms)

Trace Clustering Method	Number of clusters	Trace clustering time
ArciTrac	3	6215
	4	5411
	5	4849
Guide Tree Miner	3	9024
	4	4415
	5	7451
K-Means	3	2189
	4	2163
	5	2160

5.3.2 Process Model Quality

By comparing the quality of the process model generated by the above process discovery method with the quality of the process model generated by the newly proposed method, the quality of the traditional process model is to compare the fitness, precision, and F-Measure of the process model generated by the original log and the original log. The Measure value is quantified; the new method proposed in this paper is to obtain the corresponding process model through the existing process discovery method through the several sub-logs generated by trace clustering, and then respectively do the fitness degree and the original log of the respective process model and the original log. The accuracy and F-Measure index values are quantified, and then the weighted average is used to obtain the final evaluation value. The results obtained are shown in Table 3, Among them, *F* represents fitness, *P* represents precision, *F1* represents *F-Measure*.

It can be seen from Table 3 that, except for the Alpha algorithm, which cannot obtain the relevant results, the final evaluation quality values obtained by the other process mining algorithms are all greater than the quality of the logs directly evaluated. This shows that the new method proposed in this paper has improved the process to a certain extent. Accuracy of discovery. Take the clustering method *ArciTrac*, when the number of clusters is 4 as an example, it is found that the fitness value of the method is reduced, but the accuracy value is increased, and the harmonic average value of the two is F-Measure value is increased, which shows that the quality of the process model has been improved.

6. CONCLUSIONS

This paper proposes a preprocessing method, called as trace clustering to improve the performance of the process discovery methods. The analysis on the simulation experiment data set shows that the method proposed in this paper can not only effectively improve the time performance of the process discovery method, but also improve the quality of the process model.

Table 3. Comparison of evaluation indicators

Trace Clustering Method	Number of clusters	Process Mining Algorithms											
		Alpha Miner			Heuristic Miner			Inductive Miner			Fodina Miner		
		F	P	F1	F	P	F1	F	P	F1	F	P	F1
Original Log		-	-	-	0.8557	0.795	0.8242	0.9527	0.516	0.67	0.847	0.768	0.8057
ArciTrac	3	-	-	-	0.7492	0.987	0.852	0.7909	0.9326	0.8546	0.7501	0.993	0.8547
	4	-	-	-	0.77	0.9379	0.8409	0.8207	0.8045	0.7988	0.7756	0.9378	0.8423
	5	-	-	-	0.78	0.9338	0.8427	0.8206	0.7637	0.7778	0.7667	0.9358	0.8373
Guide Tree Miner	3				0.8436	0.8353	0.839	0.925	0.5824	0.703	-	-	-
	4				0.836	0.841	0.8379	0.9185	0.5403	0.763	-	-	-
	5				0.8379	0.8511	0.8438	0.8791	0.616	0.7056	-	-	-
K-Means	3				0.8446	0.8546	0.8496	0.9263	0.5161	0.6587	0.8329	0.7294	0.777
	4				0.8333	0.8667	0.85	0.9165	0.3907	0.5471	0.8475	0.76	0.8008
	5				0.8441	0.8704	0.8567	0.9309	0.5606	0.6977	0.827	0.777	0.8007

7. REFERENCES

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

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

[3] 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. Tavel, P. 2007 Modeling and Simulation Design. AK Peters Ltd.

[4] 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.

[5] QAFARI M S, VAN DER AALST W. Fairness-Aware Process Mining[C]//OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". Berlin, Germany: Springer-Verlag, 2019: 182-192.

[6] 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.

[7] 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.

[8] 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.

[9] 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.Forman, G. 2003. An extensive empirical study of feature selection metrics for text classification. J. Mach. Learn. Res. 3 (Mar. 2003), 1289-1305.

[10] VAN DER AALST W, et al. Process mining manifesto[C]//Business Process Management Workshops: BPM 2011 International Workshops, Clermont-Ferrand, France, August 29, 2011, Revised Selected Papers, Part I. Berlin, Germany: Springer-Verlag, 2011, 99: 169-194.

[11] VAN DER AALST W, et al. Process mining manifesto[C]//Business Process Management Workshops: BPM 2011 International Workshops, Clermont-Ferrand, France, August 29, 2011, Revised Selected Papers, Part I. Berlin, Germany: Springer-Verlag, 2011, 99: 169-194.

[12] VANDEN BROUCKE S K L M, De Weerdt J. Fodina: a robust and flexible heuristic process discovery technique[J]. decision support systems, 2017, 100: 109-118.

[13] ADRIANSYAH A, MUNOZ-GAMA J, CARMONA J, et al. Alignment based precision checking[C]//International conference on business process management. Berlin, Germany: Springer-Verlag, 2012: 137-149.

[14] SONG M, GÜNTHER C W, VAN DER AALST W M P. Trace clustering in process mining[C]//International conference on business process management. Berlin, Germany: Springer-Heidelberg, 2008: 109-120.

[15] BOSE R P J C, VAN DER AALST W M P. Context aware trace clustering: Towards improving process mining results[C]//proceedings of the 2009 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, 2009: 401-412.

[16] BOSE R P J C, VAN DER AALST W M P. Trace clustering based on conserved patterns: Towards achieving better process models[C]//International Conference on Business Process Management. Berlin, Germany: Springer-Heidelberg, 2009: 170-181.

[17] LEVENSHTEIN V I. Binary codes capable of correcting deletions, insertions, and reversals[C]//Soviet physics doklady. 1966, 10(8): 707-710.

[18] FERREIRA D, ZACARIAS M, MALHEIROS M, et al. Approaching process mining with sequence clustering: Experiments and findings[C]//International conference on business process management. Berlin, Germany: Springer-Heidelberg, 2007: 360-374.

[19] CADEZ I, HECKERMAN D, Meek C, et al. Model-based clustering and visualization of navigation patterns on a web site[J]. Data mining and knowledge discovery, 2003, 7(4): 399-424.

[20] VEIGA G M, FERREIRA D R. Understanding spaghetti models with sequence clustering for ProM[C]//International conference on business process management. Berlin, Germany: Springer-Heidelberg, 2009: 92-103.

[21] DE WEERDT J, VANDEN BROUCKE S, VANTHIENEN J, et al. Active trace clustering for improved process discovery[J]. IEEE Transactions on Knowledge and Data Engineering, 2013, 25(12): 2708-2720.

[22] PAGE L, BRIN S, MOTWANI R, et al. The PageRank citation ranking: Bringing order to the web[R]. Stanford InfoLab, 1999.