Proceedings of the Federated Conference on Computer Science and Information Systems pp. 587–590
DOI: 10.15439/2020F95
ISSN 2300-5963 ACSIS, Vol. 21

Data Mining for Process Modeling: A Clustered Process Discovery Approach

Renato Cirne, Caio Melquiades, Renan Leite, Eronita Leijden, Alexandre Maciel, Fernando Buarque de Lima Neto
University of Pernambuco (UPE)
Recife, Brazil
Email: {rbc3, casml, rfl, emlvl, amam, fbln}@ecomp.poli.br

***Abstract*—Process mining has emerged as a new scientific research topic on the interface between process modeling and event data gathering. In the search for process models that best fit to reality, the process discovery approach of creating referential processes from observed behavior. However, despite these methods showing relevant results, when faced with noisy and divergent tendencies they end up producing limited results. This work proposes the application of process discovery technique, combined to cluster technique k-means, to generate new process models, considering its conformance checking measures. The proposed solution is applied to an *ad hoc* workflow. And as a result, the use of the clustering techniques coupled with process discovery showed significant gains in the generation of process models, unlike the standard approach.**

I. Introduction

DUE to the challenges posed to companies arising from the difficulty of managing complex process flow networks, various types of problems can occur, such as delays, rework, and waste of resources. Business process management methods have been introduced to maximize process that ensure alignment of business strategies with customer and stakeholders aims [1]. Typical examples of improvement include cost savings, runtimes and failure reductions.

Most traditional areas such as Data Mining (DM), Business Intelligence (BI), and Machine Learning (ML) focus on data without considering end-to-end process models. To reduce the gap between these fields of study, process mining techniques have been successfully used [2].

The challenge of process mining is to turn big data into valuable insights related to process performance and compliance. Process mining results can be used to identify and understand bottlenecks, inefficiencies, deviations, and risks [3]. Furthermore, its techniques have been applied in several real-world system, such as [4].

One of the main focuses of the study of process mining and the object of this study is process discovery, where, based on observed behavior, a process model capable of reproducing event logs is inferred [5].

It is worth pointing out that, according to Bose et. al [6], most real-life logs tend to be granular, heterogeneous, voluminous, incomplete, and noisy. Some of the most advanced process discovery techniques try to address these problems.

Therefore, one of the categories of process mining data quality problems is noisy data or outliers. Most process mining techniques are misled by the presence of outliers, which impacts the quality of the mined results [6].

This work proposes the use of the process discovery technique, using the α-Algorithm, a technique strongly impacted by noise, combined with the technique of group selection of instances from data clustering (k-means) to generate new process models, taking into consideration their compliance function in the pursuit of the best process models.

II. State of Art

A. Process Mining

Process mining is a bridge between data mining and business process modeling [3]. To this end, it provides a process analysis method based on models and data-oriented analysis techniques. Through real datasets and algorithms, the approach provides scientific knowledge that can be applied directly to analyze and improve processes in a variety of domains [2].

An event log can be any ordered list of records known as events. Every event has at least a case identifier, an activity identifier, and some additional property such as a timestamp that can be considered to put the events into some deterministic order. This mechanism allows us to point to a specific event or a specific case. A case identifier is used to group events belonging somehow into some common contexts [7]. Therefore, such objects are important for the area of process mining and are defined by van der Aalst [3] as follows.

Definition 1 (*case*) Let C be *case universe*, i.e., a set of all possible cases identifier. Cases have attributes. For any case $c \in C$ and $n \in AN$: *#$_n$*= $\perp$ is the value of attribute *n* for case *c* (*#$_n$(c)*=$\perp$ if case *c* has no attribute named *n*).

Definition 2 (*event log*) Let *L* be a set of cases, i.e., $L \subseteq C$, such that each event appears once in the entire log at most, i.e., for any c$_1$, $c_2 \in L$ such that $c_1 \neq c_2 : \partial set(\widehat{c}_1) \cap \partial set(\widehat{c}_2) = \emptyset$. If an event log contains timestamps, then the ordering in a trace should respect these timestamps, i.e., for any case $c \in L$, *i* and *j* such that $1 \leq i < j \leq |\widehat{c}| : \#_{time}(\widehat{c}(i)) \leq \#_{time}(\widehat{c}(j))$.

Thus, for every event, an unambiguous case can be identified, which represents a collection of events belonging to the same process. The events for a case are represented in the form of a trace, i.e., a sequence of unique events.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001

Definition 3 (*trace*) Let σ be a finite sequence of events and $\sigma \in E^*$, such that each event appears only once, i.e., for $1 \leq i < j \leq |\widehat{\sigma}| : \sigma(i) \neq \sigma(j)$. Each case has a special mandatory attribute trace, $\#_{trace}(c) \in E^*.\widehat{c} = \#_{trace}(c)$ is a shorthand for referring to the trace of a case.

It is worth mentioning, in addition to the properties listed above, every event can also include any number of additional event attributes. Among the many existing ones, the process mining purpose to this project is 'Discovery'. This technique uses an event log (Definition 2) and produces a process model without using any prior information.

Process discovery output is the process model, describing events and flows. This model serves to check if events are occurring according to the proposed description [5], which is useful for compliance assurance. Therefore, compliance addresses events that should happen and are not occurring, and events that happen and are not described in the model.

In this context, the α-Algorithm is widely accepted and used [3]. It aims to extract an event log and produce a process model explaining the behavior recorded in the log [5]. So, the α-Algorithm is a process discovery algorithm that aims to build a process model through the mutual occurrences of a set of scenarios, using log-based ordering relations (Definition 4). This algorithm has as input a set of events and results in a Petri Net, defined by van der Aalst [3] and conforms to the example shown in Figure 1. In this case, events form sequences that relate to various scenarios, and the path of each scenario reports on the network.

Definition 4 (*Log-based ordering relations*) Let L be an event log over A, i.e., $L \in B(A^*)$. Let $x, y \in A$:

- $x >_L y$ if and only if there is a trace $\sigma = \langle t_1, t_2, ..., t_n \rangle$ and $i \in 1, 2, ..., n-1$ such that $\sigma \in L$ and $t_i = x$ and $t_{i+1} = y$;
- $x \rightarrow_L y$ if and only if $x >_L y$ and $x \not>_L y$;
- $x \#_L y$ if and only if $x \not>_L y$ and $y \not>_L x$;
- $x \|_L y$ if and only if $x \not>_L y$ and $y \not>_L x$.

Through a conformance checking purpose, it is possible to evaluate the existence of divergences between the model and the base, and assign a value to it [2]

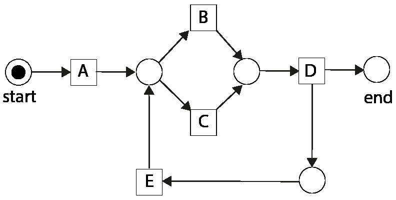

Fig. 1. Example of model in representation of Petri Net

Conformance checking [3] relates events in the event log to activities in the process model and compares both. The objective is to find similarities and discrepancies between the modeled behavior and the observed behavior.

The *token-based replay* is a refined conformance checking method that assigns a fitness value to each scenario. It allows somehow to discover the fraction of the scenario that conforms to the model and the fraction that does not [2]. The fitness calculation is done by counting the missing network tokens (m), and the remaining (r), the produced (p) and the consumed (c) ones, according to Definition 5 [8].

Definition 5 (*Fitness-token-based replay*). Let E be an event log and *PN* process model represented by a Petri Net. For each trace $\sigma \in S = \alpha(E)$ (simplified log, i.e., every event in *E* is replaced for activity attribute), consider m_σ the number of missing tokens, r_σ the number of remaining tokens, c_σ the number of consumed tokens, and p_σ the number of produced tokens during *E* reproduction in *PN*, Fitness-token-based replay (*Tbr*) is defined by:

$$Tbr = \frac{1}{2}\left(1 - \frac{\sum_{\sigma \in S} S(\sigma).m_\sigma}{\sum_{\sigma \in S} S(\sigma).c_\sigma}\right) + \frac{1}{2}\left(1 - \frac{\sum_{\sigma \in S} S(\sigma).r_\sigma}{\sum_{\sigma \in S} S(\sigma).p_\sigma}\right)$$

Thus, according Rozinat and van der Aalst [9], the number of tokens that had to be created artificially (that is, the transition belonging to the registered event was not activated and therefore could not be successfully executed) is counted and the number of tokens that were left in the model, which indicates that the process was not completed correctly. From Definition 5 it can be concluded that the closer to 1, the higher the model conforms to the reference event logs.

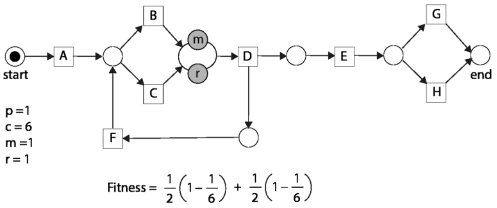

Fig. 2. Conformance checking example [3]

By means of example, Fig. 2 presents the calculation of conformance checking over a given Petri Net and a trace equal to $\langle A, D, C, E, H \rangle$.

B. K-means

K-means is a clustering algorithm based on Euclidean distance in vector spaces. This algorithm searches for center points (or centroids) that group the input vectors into sets. Each cluster has a centroid, and the k parameter determines the number of centroids (and thus, the number of clusters).

One of the techniques used to determine k, the number of clusters, is called the elbow method. It is a visual method. The idea is that it starts with k = 2, and increases the k step by step by 1, calculating the clusters and training-related cost. At some k value, the cost drops dramatically, and so it reaches a plateau when k is raised again, and hence the desired k value is obtained [10].

It is noteworthy that this type of combination of techniques has already been the object of research in the area of process mining, but from a different approach [11], which iteratively divided the traces into clusters until the log was partitioned into clusters that allow the generation of more accurate process models.

Other example of this preprocessing approach, Greco et al [12] have devised a novel framework that substantially differs from previous approaches for it performs a hierarchical clustering in which each trace is seen as a point of a properly identified space of features.

Hinkka et al. [7] concluded that the most consistent feature selection algorithm was the cluster algorithm developed in their paper, which first used the k-means algorithm to group the characteristic in the desired number of clusters.

Recently, Fani Sani et al. [13] analyzed several methods of selecting subsets and demonstrated that it is possible to considerably accelerate the discovery using strategies of subset selection of features. In addition, the results show that selection with some bias of process instances compared to random selection results in higher quality process models.

III. Materials and methods

A. Experiment

The database used here has been provided by the Government of the State of Pernambuco (Brazil) and consists of all public data recorded in the Electronic Information System (SEI) that occurred until 27/03/2019.

As an *ad hoc* workflow, it runs business processes with no predetermined pattern of information movement between users. Moreover, the use of techniques such as process discovery allows application on new challenges of process management.

For this, a CSV file has been created using the relevant attributes for the research and data referring to the following processes of a public agency of Pernambuco Government: (1) passive transparency; (2) Equity Movement; (3) Holiday Alteration; (4) Electoral License; (5) Contract Monitoring.

For α-Algorithm execution, by default, the attributes that represent the process event log have been renamed and the data were converted to the eXtensible Event Stream (XES) format developed by Verbeek et al. [14] to meet the α-Algorithm assumption.

Finally, the experiment has been confirmed using a sample of dataset made available by BPI Challenge 2019 collected [15] from a large multinational company operating from the Netherlands in the area of coatings and paints. Specifically, one type of cases in the data “3-way matching, invoice before goods receipt” has been used.

B. Modeling

In order to reduce the complexity of the model produced by the α-Algorithm, it has been realized an unsupervised search for scenario blocks that produced similar conformance variables and produced models of these blocks, as shown in Fig. 3.

Initially, the α-Algorithm is executed on the entire event log to generate an initial model, to be used at the end of the experiment to compare performance gains.

In summary, the experiment performed the following steps:

1) the event log is randomly divided into *n blocks* with the same number of cases;
2) α-Algorithm is applied to generate the model on *n-1* blocks *remaining* for each block;
3) *token-based replay* (TBR) is performed on the selected block in comparison with the generated model. The procedure is repeated with all the blocks, so that each one of them is used once, guaranteeing the complete approach of all data;
4) before using k-means for clustering traces, the elbow method was applied to determine the number of clusters created (k) (Tbr measures);
5) k-means algorithm is used to create groups of traces in each scenario using the variables produced (p), consumed (c), remaining (r) and missing (m);
6) the best group of traces is chosen by calculating the fitness average of the most representative token-based replay method.

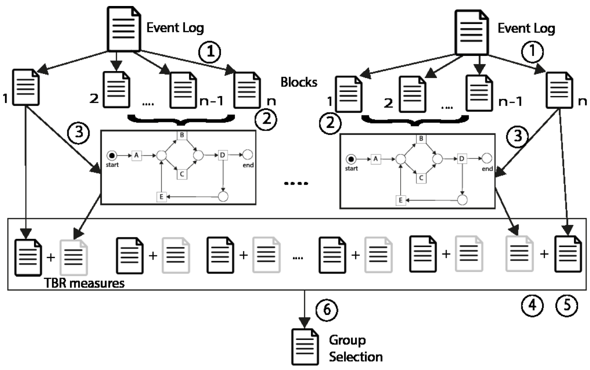

Fig. 3. Method proposed

IV. Analysis

Considering the previously established criteria and after data pre-processing, the α-Algorithm has been used for the generation of the initial Petri Net of these process types.

In order to verify the quality of the initially generated network, a conformance checking has been calculated using the token-based replay method of the generated model in comparison with all the traces related to the process in question (column “Initial” of Table 1).

Subsequently, the dataset has been divided into five blocks, and the α-Algorithm has been used to generate new Petri Nets. Then, the statistical analysis of the generated processes considering all the blocks has been performed. In this context, the graph shown in Fig. 4 demonstrates that the number of tokens produced per trace has greater dispersion, as well as the number of remaining tokens. This feature may imply that the traces are diverse and have relevant cases of unfinished flows, demonstrating their heterogeneity. Therefore, this suggests that the use of clustering techniques improves the model.

Before using k-means for clustering traces, the elbow method had been applied to determine the number of clusters created (k). In the five types of process in question, the mode was k equal 3, thus being the most suitable for clustering.

Finally, after selecting the traces group that has a higher average fitness value from each block, a new process model has been generated, and conformance checking calculated, considering all traces, as presented in the “Final” column of Table 1.

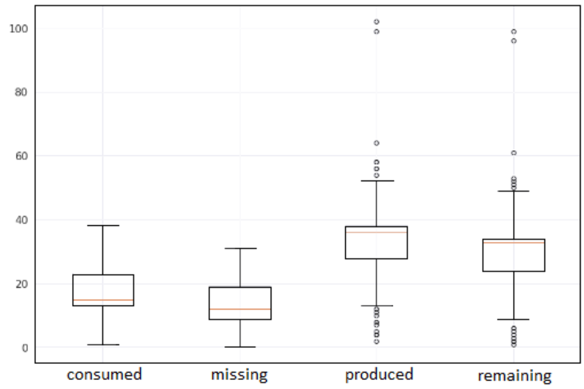

Fig. 4. Boxplot - Conformance Checking Chart of All Blocks - Process Type: Passive Transparency

From the results presented in Table 1, the fitness function of the initial model presented gains in the conformance checking calculation, using the *token-based replay* method, ranging from 6% to 30%, depending on the type of process.

TABLE I
FITNESS GAIN AFTER APPLICATION OF THE CLUSTER APPROACH

Process Type	Initial	Final	Gain
Passive Transparency	0.1476912	0.1966167	33.13%
Equity Movement	0.4651573	0.5134469	10.38%
Holiday Alteration	0.4111576	0.4502492	9.51%
Electoral License	0.2129815	0.2678744	25.77%
Contract Monitoring	0.2114739	0.2262380	6.98%

V. Conclusion

Given the results obtained, the use of clustering techniques (k-means) coupled with process discovery (via the α-Algorithm) showed substantial gains in the generation of new process models. In this sense, less complicated process models can be generated from the considered events, with more adequacy. This is different from the standard approach, where it is only possible to evaluate scenario by scenario, separately.

As there might be a clear distinction of performance between the different categories of processes, it is important to evaluate the process features in order to use the technique best suited to process discovery problems. This is due to the fact that nonconforming flows and divergent trends end up producing insufficient results for this type of approach. Thus, the condition of process log heterogeneity may recommend the use of clustering techniques for effective gains in process model generation.

It is noticeable that the application of the proposed model in all processes tested had positive results. However, depending on the complexity of its configuration, there will be an increase in computational effort. The experiments reveal that there were significant impacts on the solutions even though there was not substantial loss in performance.

Differently from Medeiros et al. [11] and according to Fani Sani et al., this method evidence, which was clustered iteratively such that each of the resulting clusters corresponds to a coherent set of cases, can consider some bias, that in the case of this research are compliance measures, to allow the generation of a more accurate process.

Finally, the application of techniques for feature selection has been evaluated as an opportunity for future work, especially approaches that use artificial intelligence techniques.

References

[1] ABPMP, BPM CBOK VERSION 4.0 - A Guide to Business Process Management - Common Body of Knowledge. 2019.

[2] W. M. P. van der Aalst, ‘Process mining in the large: A tutorial’, Lect. Notes Bus. Inf. Process., vol. 172 LNBIP, pp. 33–76, 2014,https://doi.org/10.1007/978-3-319-05461-2_2.

[3] W. van der Aalst, Process Mining: Data Science in Action, 2nd ed. Berlin Heidelberg: Springer-Verlag, 2016.

[4] P. Markowski and M. R. Przybyłek, ‘Process mining methods for post-delivery validation’, in 2017 Federated Conference on Computer Science and Information Systems (FedCSIS), Sep. 2017, pp. 1199–1202,https://doi.org/10.15439/2017F372.

[5] W. van der Aalst, T. Weijters, and L. Maruster, ‘Workflow Mining: Discovering Process Models from Event Logs’, IEEE Trans. Knowl. Data Eng., vol. 16, no. 9, pp. 1128–1142, Sep. 2004,https://doi.org/10.1109/TKDE.2004.47.

[6] R. P. J. C. Bose, R. S. Mans, and V. D. W. M.P. Aalst, Wanna improve process mining results?: it’s high time we consider data quality issues seriously’, 2013 IEEE Symp. Comput. Intell. Data Min. CIDM13 Singap. April 16-19 2013, pp. 127–134, 2013,https://doi.org/10.1109/CIDM.2013.6597227.

[7] M. Hinkka, T. Lehto, K. Heljanko, and A. Jung, ‘Structural Feature Selection for Event Logs’, ArXiv171002823 Cs Stat, vol. 308, pp. 20–35, 2018,https://doi.org/10.1007/978-3-319-74030-0_2.

[8] A. Rozinat, ‘Process mining: conformance and extension’, 2010,https://doi.org/10.6100/IR690060.

[9] A. Rozinat and W. M. P. van der Aalst, ‘Conformance Testing: Measuring the Fit and Appropriateness of Event Logs and Process Models’, in Business Process Management Workshops, Berlin, Heidelberg, 2006, pp. 163–176,https://doi.org/10.1007/11678564_15.

[10] T. M. Kodinariya and P. R. Makwana, ‘Review on determining number of Cluster in K-Means Clustering’, 2013.

[11] A. K. A. de Medeiros et al., ‘Process Mining Based on Clustering: A Quest for Precision’, in Business Process Management Workshops, Berlin, Heidelberg, 2008, pp. 17–29,https://doi.org/10.1007/978-3-540-78238-4_4.

[12] G. Greco, A. Guzzo, L. Pontieri, and D. Sacca, ‘Discovering expressive process models by clustering log traces’, IEEE Trans. Knowl. Data Eng., vol. 18, no. 8, pp. 1010–1027, Aug. 2006,https://doi.org/10.1109/TKDE.2006.123.

[13] M. Fani Sani, S. J. van Zelst, and W. M. P. van der Aalst, ‘The Impact of Event Log Subset Selection on the Performance of Process Discovery Algorithms’, in New Trends in Databases and Information Systems, Cham, 2019, pp. 391–404,https://doi.org/10.1007/978-3-030-30278-8_39.

[14] H. M. W. Verbeek, J. C. A. M. Buijs, B. F. van Dongen, and W. M. P. van der Aalst, ‘XES, XESame, and ProM 6’, in Information Systems Evolution, Berlin, Heidelberg, 2011, pp. 60–75, https://doi.org/10.1007/978-3-642-17722-4_5.

[15] van Dongen, B.F., ‘Dataset BPI Challenge 2019’. 4TU.Centre for Research Data., 2019, https://doi.org/10.4121/uuid:d06aff4b-79f0-45e6-8ec8-e19730c248f1.