Behavioural Clustering by Extensive Declarative Specifications Measurements(Extended Abstract)

Cecconi, Alessio

Published in:
Proceedings of the ICPM Doctoral Consortium and Tool Demonstration Track 2020 co-located with the 2nd International Conference on Process Mining (ICPM 2020)

Published: 01/01/2020

Behavioural Clustering by Extensive Declarative Specifications Measurements (Extended Abstract)

Alessio Cecconi
Vienna University of Economics and Business, Vienna, Austria
alessio.cecconi@wu.ac.at

I. Introduction

The recognition, classification and grouping of distinct process behaviours in an event log is a key aspect of process analysis. In unstructured and flexible processes contexts this is not straightforward and the literature devises different techniques to tackle the problem. An effective one has been found in trace clustering, namely a set of techniques which automatically group similar traces according to specified criteria, allowing for better understandability and decreased complexity of the analysis. However, all available clustering techniques are designed exclusively with procedural process models. For those techniques the key aspect for trace similarity is the precise sequence of execution of events, as they consider only events that immediately follow or precede one another. Yet, the properties and relations of events in a process may fall outside such a narrow scope.

In our research, we want to explore the opportunity of employing declarative process mining for trace clustering. We believe that the characteristics of declarative specifications can lead to novel results given the focus on different relations of the events in the event log. Indeed, a declarative rule describes a desired property of the process, not a specific execution. Thus, grouping around them suggests clusters centred on flexible, complex, and yet specific behaviours of the process instead of strict events sequence similarity.

Any clustering technique is based on similarity (or distance) concepts describing how close or distant objects are. Nevertheless, the current declarative rules evaluation methods are limited to devise a comprehensive similarity concept for traces based on rules. To fill this gap it is required an extensive measurement system for declarative specifications.

II. Background

Trace Clustering. The goal of trace clustering is to find traces of similar behaviour and group them into clusters. The guiding rule is to maximise the similarity within a cluster while maximising the distance with the other clusters. Three main class of approaches exist: *(i)* Vector-based, where the traces are transformed into feature vectors and distance metrics are used in the vector space (e.g. [1], [2]); *(ii)* Context-aware, where string distance metrics are applied directly on the whole traces (e.g. [4], [9]); *(iii)* Model-based, where traces are clustered around fitting process models (e.g. [5], [6]). Trace clustering has been employed in process mining to assist the discovery of procedural process models. Dividing the event log into different clusters, the discovery techniques can be applied only to discover the models of each cluster, resulting in a set of simpler and more understandable models of particular behaviours of the process. Table I summarizes the current applications of trace clustering in process mining.

Table I: Process mining techniques using trace clustering.

Technique	Clustering approach	Control-flow perspective	Data perspective	Clustering algorithms
Greco et al. [1]	Vector-based	**procedural**	no	K-means, Hierarchical Clustering
Song et al. [2]	Vector-based	**procedural**	yes	K-means, Quality Threshold, Agglomerative Clustering, SelfOrganizing Maps
Jablonski et al. [3]	Vector-based	**procedural**	yes	Hierarchical clustering
Bose and van der Aalst [4]	Vector-based	**procedural**	no	Hierarchical clustering
Ferreira et al. [5]	Model-based	**procedural**	no	1st order Markov chain Expectation-Maximization
De Koninck and De Weerdt [6]	Model-based	**procedural**	no	Active learning
Wang et al. [7]	Model-based	**procedural**	no	Constrained clustering, agglomerative hierarchical clustering, spectral clustering
Bose and van der Aalst [8]	Context-aware	**procedural**	no	edit-distance, agglomerative clustering
Evermann et al. [9]	Context-aware	**procedural**	no	K-means
Nguyen et al. [10]	Mixed	**procedural**	yes	Graph path similarity
De Koninck and De Weerdt [11]	Mixed	**procedural**	yes	K-means, active learning

It can be noticed that different approaches, perspectives, and algorithms have been tried, yet all the current trace clustering techniques in process mining share, not really a limit, but rather a common trait: only procedural models are considered. Accordingly the control-flow perspective is inspected only for its continuous subsequences, i.e., only directly following relations, thus local proximity of activities is preferred in the clustering composition. This is not a limit of the clustering techniques per se, but in the object used to devise the characteristics upon which basing the clustering. For example, consider two traces $\langle a, b, c, d, e, f \rangle$ and $\langle b, a, d, c, f, e \rangle$ where the events are couple-wise swapped, but a transitivity property between tasks a, b, and c is preserved (i.e., $a \rightarrow c \rightarrow e$). If this transitivity property is of interest, both the traces should be grouped in the same cluster, but the directly-follow relations between the two traces is messed, thus they may result too different to appear in the same cluster. As a result, similar traces may be disjoined or different ones may be grouped.

Evaluation of declarative specifications. Declarative process mining mostly resorts to quality measures from *association rule mining* [12] to qualify single rules with respect to event logs. Support and confidence are the most adopted measures on that regard, yet they are reportedly not sufficient to avoid a great amount of spurious results [13], which threatens the statistical soundness of the results. Also, there are different definitions for support [14], [15], [16] and confidence [14], [15], [16]. For example, the support measure of [16] cannot be compared to the support of [14] because of the different definitions. Furthermore these techniques defined the measures only for a limited set of rules (i.e., the standard DECLARE rules-set). Thus, the comparison of techniques is hampered by their customized definitions of the same measures and the transferability of measures themselves between techniques is limited. The result is a scattered adoption of a small set of measures dependent either to a specific language or set of rules. Different other measures have been studied to go beyond this limit [17], yet they have been not fully exploited in process mining area. Thus a more advanced and extensive evaluation system for declarative specifications is required to base efficiently trace clustering on them.

III. CONTRIBUTION

With this research we aim to explore the integration of declarative process mining and trace clustering. The expressiveness of declarative rules can allow for a new clustering based on clear desired properties of the process, and not strict events sequences. In order to do so, an extension of the current evaluation techniques for declarative specifications is required.

A declarative specification allows for complex relations among activities regardless of their distance in the execution flow. That is because each specification models a desired properties of the process, not a specific executions. At the best of our knowledge, the combination of declarative process mining with trace clustering is still unexplored. We believe that this novel intuition can lead to distinct and interesting results, beyond the reach of procedural processes. Also, clustering around rules makes the clustering semantic explicit, easing supervised techniques and the injection of experts knowledge.

To make this clustering possible, it is mandatory to devise a similarity concept between traces and rules. Indeed a declarative injection can be used for both model-based and vector-based techniques. For both is paramount to devise an informative evaluation of the rules on the trace. The validity or violation of a rule in a trace can be a possible direction, but the boolean evaluation may be too limited to clearly differentiate the clusters. Furthermore it would be a single perspective, not enough to build a feature vector. A more flexible and broad mean of rules evaluation would be desirable, but the current declarative techniques are limited on that regard. For this reason we will devise an extensive measurement framework for declarative specifications going beyond these limits.

The goal of our measurement framework is to provide a sound ground where to define, compute, and verify measures for generic temporal logic formulae. On top of it will be based the similarity function for clustering of traces. In order to validate these results, we are going to implement the measurement framework first and the overall behavioural clustering afterwards into a proof-of-concept software with which experimental evaluations will be conducted. The empirical evaluation of the techniques will be carried out both on simulated artificial data and publicly available real-life data like BPI Challenge datasets, e.g. [18]. The controlled environment of a simulation is required to check the validity of the results in absence of a ground truth, while real-life data allows to asses the feasibility of the technique in realistic settings.

IV. CONCLUSION

Trace clustering is a relevant topic and the employment of declarative process mining in that regard is promising and especially still unexplored. Yet, the current evaluation systems for declarative specification are not enough for a truly effective trace clustering based on them. Given these open points, there is a call for: *(i)* an extended evaluation system for declarative specifications. *(ii)* a novel application of declarative process mining for trace clustering. Markedly, we recently achieved the first point in [19], based our previous work [20].

REFERENCES

[1] G. Greco, A. Guzzo, L. Pontieri, and D. Saccà, “Discovering expressive process models by clustering log traces,” *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 8, pp. 1010–1027, 2006.

[2] M. Song, C. W. Günther, and W. M. P. van der Aalst, “Trace clustering in process mining,” in *BPM Workshops*, 2008, pp. 109–120.

[3] S. Jablonski, M. Röglinger, S. Schönig, and K. M. Wyrtki, “Multi-perspective clustering of process execution traces,” *Enterp. Model. Inf. Syst. Archit. Int. J. Concept. Model.*, vol. 14, pp. 2:1–2:22, 2018.

[4] R. P. J. C. Bose and W. M. P. van der Aalst, “Trace clustering based on conserved patterns: Towards achieving better process models,” in *BPM Workshops*, 2009, pp. 170–181.

[5] D. R. Ferreira, M. Zacarias, M. Malheiros, and P. Ferreira, “Approaching process mining with sequence clustering: Experiments and findings,” in *BPM*, 2007, pp. 360–374.

[6] P. De Koninck and J. De Weerdt, “Multi-objective trace clustering: Finding more balanced solutions,” in *BPM Workshops*, 2016, pp. 49–60.

[7] P. Wang, W. Tan, A. Tang, and K. Hu, “A novel trace clustering technique based on constrained trace alignment,” in *HCC*, 2017, pp. 53–63.

[8] R. P. J. C. Bose and W. M. P. van der Aalst, “Context aware trace clustering: Towards improving process mining results,” in *SIAM International Conference on Data Mining*, 2009, pp. 401–412.

[9] J. Evermann, T. Thaler, and P. Fettke, “Clustering traces using sequence alignment,” in *BPM Workshops*, 2015, pp. 179–190.

[10] P. Nguyen, A. Slominski, V. Muthusamy, V. Ishakian, and K. Nahrstedt, “Process trace clustering: A heterogeneous information network approach,” in *SIAM International Conference on Data Mining*, 2016, pp. 279–287.

[11] P. De Koninck and J. De Weerdt, “Scalable mixed-paradigm trace clustering using super-instances,” in *ICPM*, 2019, pp. 17–24.

[12] L. Geng and H. J. Hamilton, “Interestingness measures for data mining: A survey,” *ACM Comput. Surv.*, vol. 38, no. 3, p. 9, 2006.

[13] W. Hämäläinen and G. I. Webb, “A tutorial on statistically sound pattern discovery,” *Data Min. Knowl. Discov.*, vol. 33, no. 2, pp. 325–377, 2019.

[14] F. M. Maggi, R. P. J. C. Bose, and W. M. P. van der Aalst, “Efficient discovery of understandable declarative process models from event logs,” in *CAiSE*, 2012, pp. 270–285.

[15] S. Schönig, A. Rogge-Solti, C. Cabanillas, S. Jablonski, and J. Mendling, “Efficient and customisable declarative process mining with SQL,” in *CAiSE*, 2016, pp. 290–305.

[16] C. Di Ciccio and M. Mecella, “On the discovery of declarative control flows for artful processes,” *ACM Trans. Management Inf. Syst.*, vol. 5, no. 4, pp. 24:1–24:37, 2015.

[17] T. B. Le and D. Lo, “Beyond support and confidence: Exploring interestingness measures for rule-based specification mining,” in *SANER*, 2015, pp. 331–340.

[18] B. F. van Dongen, “BPI challenge 2012,” Eindhoven University of Technology, 2012.

[19] A. Cecconi, G. De Giacomo, C. Di Ciccio, F. M. Maggi, and J. Mendling, “A temporal logic-based measurement framework for process mining,” in *ICPM*, 2020.

[20] A. Cecconi, C. D. Ciccio, G. De Giacomo, and J. Mendling, “Interestingness of traces in declarative process mining: The janus LTLpf approach,” in *BPM*, 2018, pp. 121–138.