Label-independent feature engineering-based clustering in Public Administration Event Logs

Flavio Corradini[1], Caterina Luciani[1], Andrea Morichetta[1], Marco Piangerelli[1] and Andrea Polini[1]

[1]University of Camerino, Via Andrea D'Accorso, 16, 62032 Camerino, Italy

Abstract

Process mining algorithms infer business models by analyzing Log files derived from the execution of business activities in organizations. In this paper, a label-independent clustering methodology is proposed. It allows an analysis completely agnostic with respect the nature and domain knowledge of the process Logs. The methodology is totally data-driven and it is based on features that do not depend on activity labels and do not need model extraction at all, thus not requiring the four quality dimensions of a mining discovery algorithm to be satisfied. Due to its independence from asset labels, the methodology is very flexible and applicable in different scenarios. The methodology was tested on the process logs of a municipality of twenty thousand inhabitants showing good performances when evaluated using a mining discovering algorithm.

Keywords

Label-independent clustering, Process Mining, Complexity

1. Introduction

All organisations, be they governmental, non-profit or corporate, are characterised by processes [1]. As pointed out by [2], process analysis can be used to achieve standardisation: for example, in the case of merging and acquisitions of companies, it may be important to recognise similar processes and give a common description of them in terms of models. However, the processes are not always modelled or the model does not necessarily correspond to the actual process performed. Process mining [3] is a technique that allows the process models to be derived from the logs produced by information systems. In [4], process mining techniques are used to visually compare two logs. This methodology has been applied by [5] to compare public administration logs. The authors showed that an activity is only present in logs acquired after 2017, perhaps as a result of a new law coming into force. However, the approach has two limitations: only two logs can be compared at a time and the methodology requires homogeneity between the labels of the two logs to be effective. Particularly the latter, is a stringent requirement in reality, where the labels may be different either because of recording errors, or because they are modelled with different names [6]. In the context of model comparison, the problem of labels was addressed

EGOV-CeDEM-ePart 2022, September 06–08, 2022, Linköping University, Sweden (Hybrid)

flaviocorradini@unicam.it (F. Corradini); caterinaluciani@unicam.it (C. Luciani); andreamorichetta@unicam.it (A. Morichetta); marcopiangerelli@unicam.it (M. Piangerelli); andreapolini@unicam.it (A. Polini)

0000-0001-6767-2184 (F. Corradini); - (C. Luciani); 0000-0003-1738-9043 (A. Morichetta); 0000-0002-8545-3740 (M. Piangerelli); 0000-0002-2840-7561 (A. Polini)

and several solutions were found, based on measures of syntactic and semantic similarity [2]. In [7], the authors highlight that there is still a lack of methods for comparing logs underlying models with activities from different universes. The authors propose a methodology of log comparison that does not take into account labels, comparing the temporal distribution of activities in the two logs. However, the timestamp is not always available in logs and the authors do not develop a clustering methodology. In [8] the authors develop a log clustering methodology that strictly depends on labels limiting its applicability to logs containing similar labels. Here we present a label-independent clustering methodology, which allows an analysis agnostic to the nature and domain knowledge of the process log collection. The methodology is based on feature engineering that does not depend on activity labels and does not require model extraction in any way. This results in a fully data-driven approach, which does not need to take into account the four quality dimensions of a mining discovery algorithm [9]. Due to its independence from asset labels, the methodology is applicable both in the case of clustering collections of logs representing variants of the same process and in the case of collections referring to several processes, even different in scope. It is therefore a flexible instrument, ready to respond to the needs of the process analyst. The methodology was tested on the process logs of a municipality of twenty thousand inhabitants. The rest of the paper is organised as follows. Section 2 introduces the proposed data-driven methodology, while Section 3 illustrates the main results applied to real logs. Finally, Section 4 concludes the paper by touching upon directions for future work.

2. Methodology

The aim of this work is to propose a very simple and completely data-driven, label independent methodology for clustering Logs in an agnostic way, complementary to process mining. In order to do that we followed the general guidelines for performing data analysis that were developed during the Da.Re. project [10]. According to this methodology, the archetypal data lifecycle is a four-step process: 1) searching for relevant data, which consists in providing clean and useful data; 2) data preparation or feature engineering, in which data are modified and treated in such a way to be "good and ready" for the analysis; 3) data analysis, consisting in the definition of the most suitable learning algorithm and 4) data visualization, for presenting them in an impacting way. In our methodology, starting from Logs, we extracted features from data and then we used them to setting up an unsupervised clustering algorithm. The features were selected to be label-independent, or agnostic, i.e. those features do not depend on activity names. The approach is fully data-driven: logs can be compared without having to extrapolate a model and without having to take into account the four quality dimensions of the discovery algorithms. The minimum requirement for a Log to be eligible for process mining is that each event must be both case/activity-related [11]. The same assumption has been made for feature extraction, thus creating a common starting point for the two approaches to be compared. The methodology is based on 4 main features: the Lempel-Ziv complexity of a Log (the number of distinct substrings that can be found in it Lempel, (1976) [12]), the average number of the predecessors of activities that are not "start" activities (α), the average number of successors of activities that are not "end" activities (β), and, finally, the number of unique activities in the

Log. The presented features are selected to be the minimal set useful for describing Logs and balancing global (Lempel-Ziv) and local (α and β) information, following the Occam's razor principle to not add complexity.

Lempel-Ziv Complexity. The Lempel-Ziv complexity (LZC) is the number of distinct substrings which can be found in a string [12]. Actually, given a sequence of length n containing the symbols from an alphabet Ω, the idea is to parse the sequence into a number $c(n)$ of distinct strings, by considering as a new word any subsequence never met before. Considering the sequence $AABABBBABAABABBBABBABB$, the final LZ algorithm output is: $A|AB|ABB|B|ABA|ABAB|BB|ABBA|BB$, with a complexity $c(n) = 9$. To perform the LZC algorithm, all Log traces were merged into a single sequence, and a delimiter was inserted between each trace. The LZ algorithm was modified so that only substrings between two delimiters were detected. This was necessary to prevent the detection of non-existent behaviour, and to limit the maximum length of detected substrings, which is otherwise potentially infinite and completely dependent on the number of log traces and their sorting.

Unique activities. Unique activities, v, are defined as the total number of distinct activities in the Log. Considering, for example, the following Log: $\mathscr{L}1 = ABCED, BCED, ACBDE$ then we can compute, $V(\mathscr{L}1) = 5$

α and β. For each activity in the sequence Log, a predecessor and successor set is derived. Considering the Log $\mathscr{L}1$ two sets are generated: successors = $A : B, C, B : C, D, C : D, E, B, D :, E :$ predecessors = $A :, B :, C : B, A, D : E, B, E : C, D$. Instead, with α/β we indicate the average number of direct distinct predecessors/successors not considering the starting and ending activities. Formally they are defined as $\alpha = \sum_{i=1}^{v} m \cdot a_i$, $\beta = \sum_{i=1}^{v} m \cdot b_i$ where m = number of activities of the Log without start (end) events, v = number of Unique activities of the Log and a and b are respectively the number of different predecessors and successors of the activity i. Hence, taking into account successors and predecessors sets, $\alpha = 7/5$ and $\beta = 6/5$. Clustering. Clusters are homogeneous groups of objects that are grouped on the base of a similarity criterion, i.e. a similarity function, in such a way that the similarity among objects belonging to the same cluster is greater than the one among objects belonging to different clusters. K-Means algorithm, equipped with a Euclidean similarity function, was used for clustering; the optimal number of clusters was assessed by using the Silhouette Index. Process clustering is a well-established problem in business process management. In particular, the authors in [8] develop a methodology based on ad-hoc similarity that allows the identification of variants in a large collection of logs. However, the methodology is based on the assumption that the activity labels belong to the same universe and are therefore comparable across different logs.

3. Results

In order to show the feasibility of the approach, we applied the described methodology to a collection of execution Logs provided by a European company. Those Logs refer to public administration processes of a municipality of about twenty thousand inhabitants. The whole dataset is composed of about 40 logs. After an inspection of the Logs (preprocessing), 29 Logs, which met the minimum quality requirements for analysis, were used. In addition, only the minimum requirements, i.e. each activity is linked to an id case and a unique indicator of

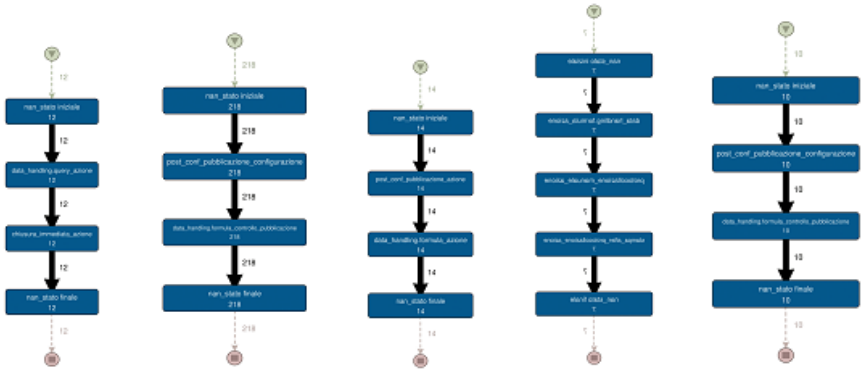

Figure 1: Cluster 0, the models show a linear control with a number of activities between 4 and 5

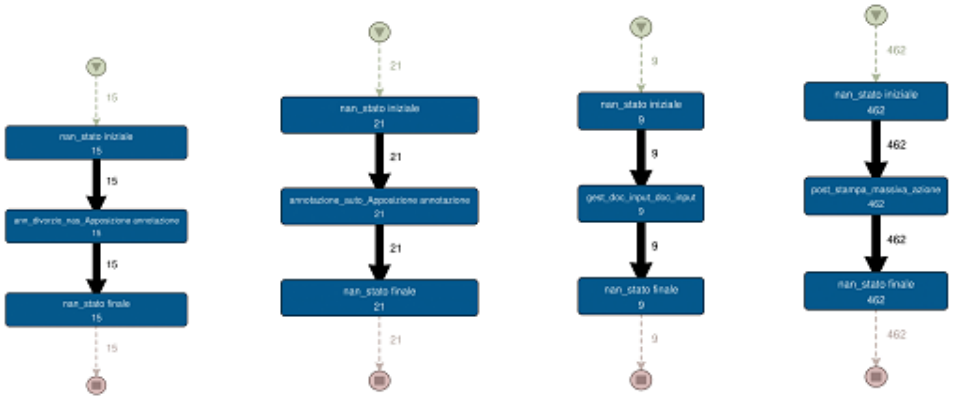

Figure 2: Cluster 8, the models show a linear control with three activities

process instantiation, have been extracted from Logs for the feature engineering. Identified features were calculated and normalised using the min-max scaling method. Cluster analysis identifies 12 clusters, with a Silhouette Index of 0.54. The clusters seem to group processes with comparable control flows, e.g. the cluster in Figures 1, 2 3 and 4. The processes in Figure 1 show linear control flow and have more activities than the models, also linear, in Figure 2. Figure 3 shows that also complex processes with numerous activities and non-linear control flow are detected by our methodology.

Clustering was also able to discern models with the same number of activities but different control flow. In Figure 4a, a three-activity model and a four-activity model are clustered together.

In fact, both have the same number of end events and therefore similar control flow. This is due to an emergent property of the two features α and β: if α cannot reveal the part of the model preceding a start activity, β will be able to reveal the contribution of the arc and vice versa. The conjunction of the two features is also able to take into account the contribution of multiple starts and ends. Model b in Fig. 4 also has four activities, but there are two loops, which is why it is clustered alone.

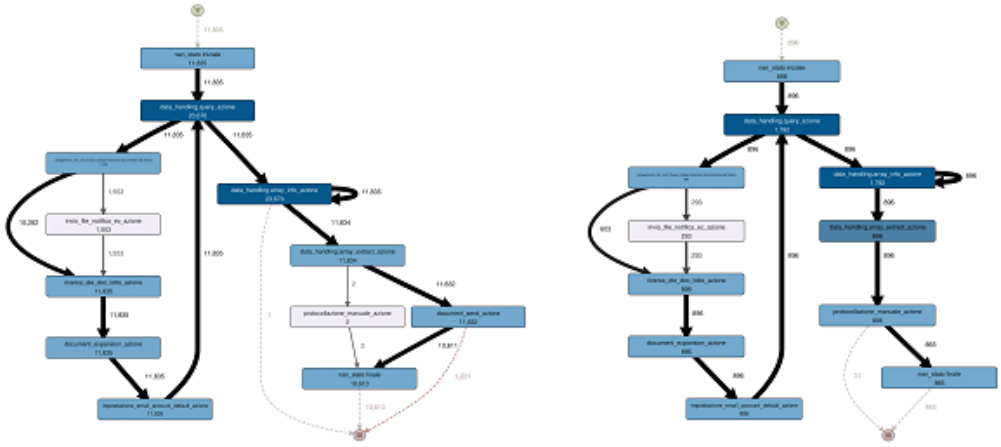

Figure 3: Cluster 3, the models show a non-linear control flow

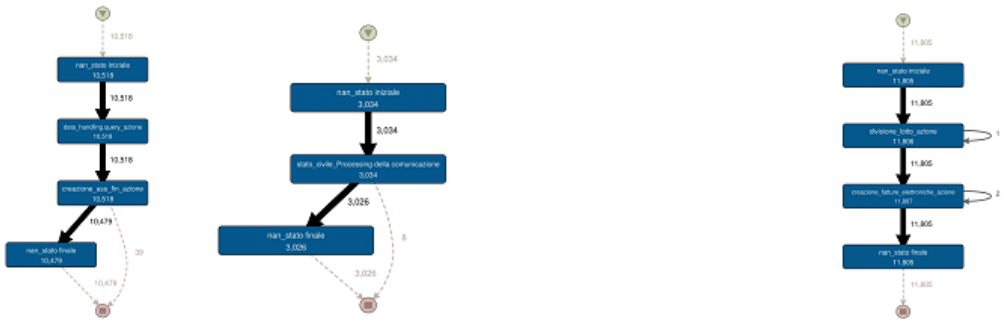

Figure 4: Clusters 4 and 5, similar in number of activities but different in control flow

4. Conclusions and future works

In this paper, an agnostic clustering methodology of process logs was presented, based on features that do not depend on activity labels. The methodology, which is completely data-driven, is applicable both in the analysis of collections of processes with the same purpose (variability analysis) and in the analysis of processes with different purposes. The methodology proposed should be considered complementary to process mining, since it is suitable to manage and analyze big amount of logs coming from different organizations, overcoming the problems related to the syntactic and semantic comparison of labels. The methodology was applied to clustering processes (with different purposes) of a European municipality of twenty thousand inhabitants. The logs were then discovered with process mining techniques and the elements of each group were compared to each other. The identified clusters highlight the potential of the methodology, which can be applied to a wide range of datasets. In the future, we aim to apply it to other known datasets or application fields [13], to evaluate further features, while maintaining the simplicity criterion adopted in the selection that was made in this work. Finally, further investigation on topology-based clustering approaches, could be of interest [14, 15].

References

[1] M. Dumas, M. Rosa, J. Mendling, H. Reijers, Fundamentals of business process management, volume 1, Springer, Heidelberg, 2013.

[2] A. Schoknecht, T. Thaler, P. Fettke, A. Oberweis, R. Laue, Similarity of business process models—a state-of-the-art analysis, ACM Computing Surveys (CSUR 50 (2017) 1–33.

[3] W. van der Aalst, Process mining: Overview and opportunities, ACM Transactions on Management Information Systems (TMIS 3 (2012) 1–17.

[4] A. Bolt, M. Leoni, W. van der Aalst, A visual approach to spot statistically-significant differences in event logs based on process metrics, in: International Conference on Advanced Information Systems Engineering, Springer, Cham, 2016, p. 151–166.

[5] F. Corradini, C. Luciani, A. Morichetta, A. Polini, Process variance analysis and configuration in the public administration sector, 2020.

[6] S. Suriadi, R. Andrews, A. Hofstede, M. Wynn, Event log imperfection patterns for process mining: Towards a systematic approach to cleaning event logs, Information systems 64 (2017) 132–150.

[7] F. Richter, L. Zellner, I. Azaiz, D. Winkel, T. Seidl, Liproma: label-independent process matching, in: International Conference on Business Process Management, Springer, Cham, 2019, p. 186–198.

[8] F. Corradini, C. Luciani, A. Morichetta, M. Piangerelli, A. Polini, $tlv - diss_\gamma$: A dissimilarity measure for public administration process logs, in: International Conference on Electronic Government, Springer, Cham, 2021, p. 301–314.

[9] J. Buijs, B. Dongen, W. van der Aalst, On the role of fitness, precision, generalization and simplicity in process discovery, in: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems", Springer, Berlin, Heidelberg, 2012, p. 305–322.

[10] J. Johnson, L. Tesei, M. Piangerelli, E. Merelli, R. Paci, N. Stojanovic, P. Leitão, J. Barbosa,

M. Amador, in: Big data: Business, technology, education, and science: Big data (ubiquity symposium), Ubiquity 2018, 2018, p. 1–13.
[11] W. van der Aalst, Process mining: data science in action, volume 2, Springer, Heidelberg, 2016.
[12] A. Lempel, J. Ziv, On the complexity of finite sequences, IEEE Transactions on information theory 22 (1976) 75–81.
[13] F. Corradini, F. Marcantoni, A. Morichetta, A. Polini, B. Re, M. Sampaolo, Enabling auditing of smart contracts through process mining, in: From Software Engineering to Formal Methods and Tools, and Back, Springer, Cham, 2019, p. 467–480.
[14] M. Rucco, A. Mamuye, M. Piangerelli, M. Quadrini, L. Tesei, E. Merelli, Survey of topdrim applications of topological data analysis, in: CEUR Workshop Proceedings, volume 1748, RWTH Aachen University, Aachen, Germany, 2016, p. 1814.
[15] M. Piangerelli, S. Maestri, E. Merelli, Visualising 2-simplex formation in metabolic reactions, Journal of Molecular Graphics and Modelling 97 (2020) 107576.