Applying Clustering in Process Mining to Find Different Versions of a Business Process That Changes over Time

Daniela Luengo and Marcos Sepúlveda

Computer Science Department
School of Engineering
Pontificia Universidad Católica de Chile
Vicuña Mackenna 4860, Macul, Santiago, Chile
dlluengo@uc.cl, marcos@ing.puc.cl

Abstract. Most Process Mining techniques assume business processes remain steady through time, when in fact their underlying design could evolve over time. Discovery algorithms should be able to automatically find the different versions of a process, providing independent models to describe each of them. In this article, we present an approach that uses the starting time of each process instance as an additional feature to those considered in traditional clustering approaches. By combining control-flow and time features, the clusters formed share both a structural similarity and a temporal proximity. Hence, the process model generated for each cluster should represent a different version of the analyzed business process. A synthetic example set was used for testing, showing the new approach outperforms the basic approach. Although further testing with real data is required, these results motivate us to deepen on this research line.

Keywords: Temporal dimension, Clustering, Process Mining.

1 Introduction and Related Work

Real-life business processes are dynamic, flexible and adaptable over time, so in different periods of time could exist different execution versions of a given process. For example, the sales process of a retail store may vary its operation between the Christmas season and the summer holidays. It can also happen that a process changes over time in order to adapt to market conditions. By having a model that describes the behavior of each version separately, it is possible to analyze them separately.

A challenge that has arisen in the literature is how to use the time recorded in the event logs to improve Process Mining techniques. In [1], the authors propose the use of time for two purposes: adding information to the process model, and improving the quality of process model discovery.

In this article, we present an approach that uses the starting time of each process instance as an additional feature to those considered in traditional Clustering in Process Mining approaches, in order to group in different clusters, process instances that are apart in time. By combining control-flow features with the starting time, the clusters formed share both a structural similarity and a temporal proximity.

F. Daniel et al. (Eds.): BPM 2011 Workshops, Part I, LNBIP 99, pp. 153–158, 2012.

Different approaches of Clustering in Process Mining have been developed to solve the problem of getting “spaghetti” process models. The clustering algorithms instead of generating a single model to explain the behavior of the process, as traditional approaches do, generate several models simpler to understand [2][3]. Clustering algorithms group together in the same cluster a consistent set of process instances based on common control-flow features, so that each cluster could later on be used to generate a more understandable process model.

In Process Mining, interest in Clustering techniques is becoming ever stronger. There are several approaches to Clustering in Process Mining, some of them are: Bag of Activities and K-gram model, which are techniques that analyze each process instance by transforming it into a vector, where each dimension of the vector corresponds to an activity instance. These techniques lack information about the context and the order in which the activities are performed. Some authors [4] have proposed that the vectors be considered as a combination of different perspectives (such as control-flow, data, performance, etc.), which could lead to better results than approaches that consider isolated perspectives, but does not solve the problem of lack of context. Another set of techniques have tried to solve this problem using the complete sequence of activities. One technique is Edit Distance that compares two process instances, assigning a cost to the difference between the two sequences [2]. On the other hand, Sequence Clustering assigns an instance to a cluster according to the probability that the cluster is capable of producing the sequence [3].

Trace Clustering is a technique that uses a robust set of features for measuring the similarity between the process instances to create the different clusters [6]. This approach assumes that if two or more instances of the process share a subset of activities, there is evidence that they have common features and have similar functionalities and could be in the same cluster. This approach, like Bag of Activities and K-gram model, maps every process instance to a vector.

In [5], a general schema is presented that proposes features and a statistical technique to detect changing points and to identify regions of change in a process based on the control-flow perspective. Instead, our work is based on Trace Clustering techniques, mainly because they add context information and they consider the order in which activities are performed, but also because the time it takes to compute is linear, unlike the techniques that work with the complete sequences. Additionally, our approach consider the different type of changes that may occur in a process according to [5], including sudden, recurring, gradual and incremental changes.

This article is organized as follows. Section 2 presents the extensions made to the Clustering in Process Mining techniques. Section 3 shows the performed experiments and main results. The final section presents the findings of our research and future work.

2 Extending Trace Clustering Techniques to Include the Temporal Dimension

Our work is based on the Trace Clustering approach proposed by Bose and van der Aalst [6]. A trace is defined as an ordered list of activities (a sequence) invoked by a process instance from its start to its end. This approach uses the sequence of activities to group the traces in different clusters. In [6] different types of activity sequences are discussed. We consider only one of them, called *Maximal Repeat (MR).* A *MR* in a sequence T is defined as a substring that occurs in a *Maximal Pair (MP)* in T. A *MP* in a sequence T is a pair of identical substring such that the symbol to the immediate left (right) of the substring are different.

This approach uses each *MR* found in the event log as a dimension of the vector space used to find clusters. We will call the set of all *MRs* found in the event log as Feature Set, and this baseline approach as Approach A.

Our approach adds an additional dimension to the vector space of Approach A, which is the starting time of each process instance (trace) in the event log, calculated as the number of days that have elapsed since a reference timestamp, e.g., January 1st, 1970, to the timestamp in which starts the first activity of the trace.

As a clustering strategy we use the Agglomerative Hierarchical Clustering (AHC) with the minimum variance criterion, using the Euclidean distance between vectors.

Based on the Approach A and the new dimension time, we developed two new approaches, which we call Approach B and Approach C. For all approaches, the distance between two traces (TA and TB) is calculated as shown in Eq. 1.

$$dist \quad T_A \, T_B = \sqrt{\sum_{i=1}^{n} \left(\frac{T_{Ai} - T_{Bi}}{\max\limits_{j \in T} (T_{ji}) - \min\limits_{k \in T} (T_{ki})} \right)^2 + \left(\frac{T_{A(n+1)} - T_{B(n+1)}}{\max\limits_{j \in T} (T_{j(n+1)}) - \min\limits_{k \in T} (T_{k(n+1)})} \right)^2 \bullet \mu} \qquad (1)$$

Where T_{ij} corresponds to the trace *i* and the feature *j* in the Feature Matrix [6]. The number of elements in the feature set is represented by *n* and the time weight by μ. The basic approach A considers μ as 0. Approach B uses as time weight the number of elements in the *Feature Set* ($\mu = n$); this approach is aimed at giving the same weight to the control-flow features and the time feature. Finally, approach C uses as time weight the factor α described in Eq. 2 with the purpose of giving an equivalent average weight to the control-flow features and the time feature.

$$\mu = \alpha = \frac{average \;\; distance \;\; between \;\; traces \;\; considering \;\; only \; the \; Feature \; Set}{average \;\; distance \;\; between \;\; traces \;\; considering \;\; only \; the \, Time \; dimension} \qquad (2)$$

3 Experimental Set and Result Analysis

To test the new approaches we used three synthetic examples created with CPN Tools [7]; as seen in Fig.1. Each example consists of 2000 process instances executed over a period of one year. In example 1, we consider a process with two very similar versions. In example 2, the process has four versions, in which some are very similar and the other ones are not. Finally, in example 3, the process has three very unlike versions. In all cases, the different versions correspond to different time periods, but there is some overlap between some versions.

For each example, we used the three approaches described previously. The clusters generated by each approach are evaluated based on a metric that measures the accuracy with which each approach is able to classify the different traces; the metric varies from 0% to 100%. A 100% value is obtained when all traces assigned to a cluster correspond to the same version of the process. The accuracy metric is calculated as the sum of all true positive and true negative in the confusion matrix, divided by the total number of traces.

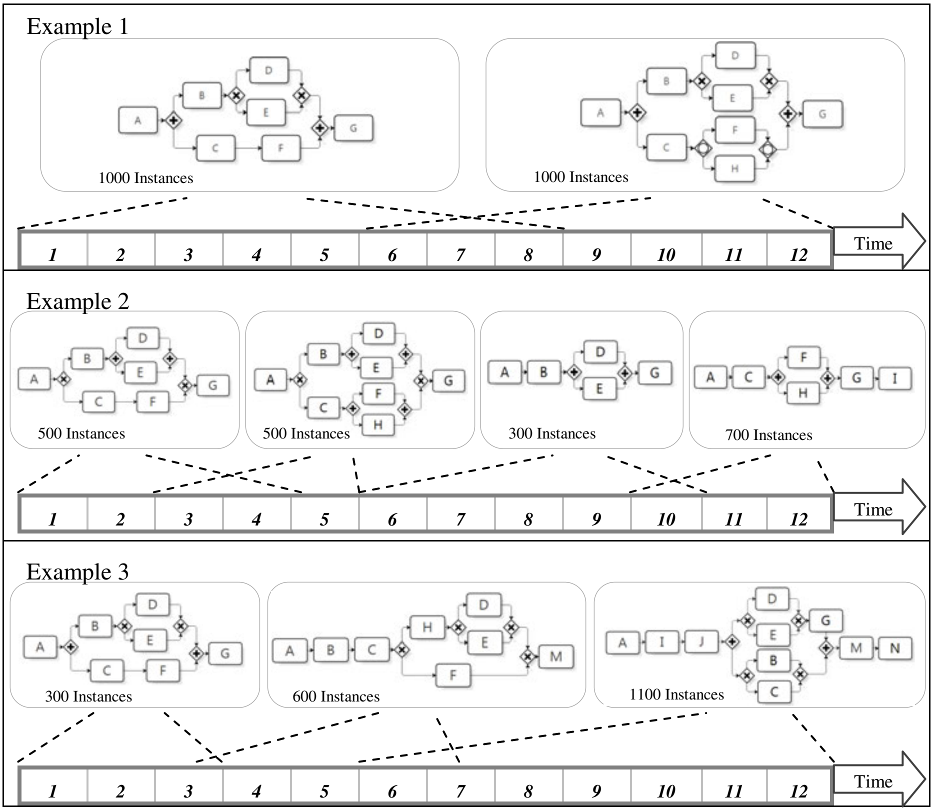


Fig. 1. Experimental set. Numbers represents the months of a year

Each approach is able to split the event log in n different clusters, where n varies from 1 to the number of process instances contained in the event log. In the results outlined, we have partitioned the event logs for each example in a number of clusters equal to the amount of original models (versions) that each example has (Fig.1). This simplification illustrates the quality of the three different approaches (Fig.2).

Example 1 consists of two different versions of a process; however, both versions differ only in one activity and run on different time periods, but during three months of the year (June, July and August) these versions overlap. Looking at the generated models with the three different approaches, we can see the Approach A, which does not consider the time dimension, does not work very well. Of the two approaches that consider the time dimension, only the Approach C is able to separate correctly the traces in the two clusters, so as to allow discovering exactly the two original models.

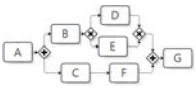
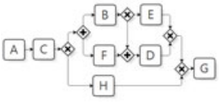
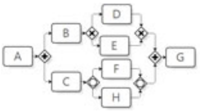
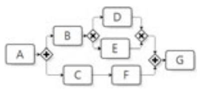
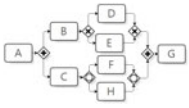
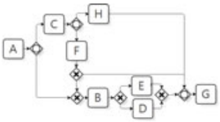
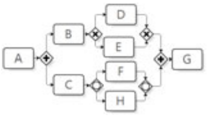
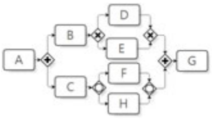
	Original Model (different versions)	Generated Model – Approach A	Generated Model – Approach B	Generated Model – Approach C
Model 1	 1000 instances	 519 instances	 1097 instances	 622 instances
Model 2	 1000 instances	 1481 instances	 903 instances	 1378 instances

Fig. 2. Original models (corresponding to different versions of the process) and models generated using the three different approaches on Example 1. Models were generated with the Heuristic Miner Algorithm in ProM 6, the same method used in [6] to generate the process models, and then transformed into the BPMN notation.

The accuracy of each approach to correctly classify the different traces in the corresponding versions of each process can be seen in Table 1.

Table 1. Accuracy metric of the different approaches for examples 1, 2 and 3

	Approach A	Approach B	Approach C
Example 1	57%	81%	81%
Example 2	55%	81%	70%
Example 3	99%	74%	100%

In Example 1 and 2, important improvements are achieved in the accuracy by the incorporation of time, due to the similarity among the original process models. But in Example 3, where exists structural differences among the different models, it is not necessary to incorporate time to obtain satisfactory results. Manually, it is possible to observe and analyze the differences between the obtain process models; or analyze the time distribution of the generated clusters to detect changing points.

4 Conclusions

In this paper, we proposed a new strategy for using existing clustering algorithms in Process Mining for analyzing processes that change over time. By incorporating the temporal dimension to the control-flow perspective traditionally considered by these algorithms, it is possible to find different versions of a process that changes over time.

This is relevant because the real-life business processes are dynamic in time, and over a long period of time may have different versions.
The incorporation of the time dimension to the Trace Clustering technique, shows positive results when the different versions of the process are similar in the control-flow perspective. When there is a greater difference, all approaches (A, B and C) show good results. This represents a motivation to deepen on this research line.

Our future work in this research line is testing this novel strategy with real processes. We would also like to enhance the clustering algorithm so as it is able to decide which approach (A, B or C) provides the best results automatically and also that is able to determine automatically the optimal number of clusters. These enhancements require defining new metrics that do not depend on a priori knowledge of the process versions, such as the accuracy metric used in this article.

References

1. van der Aalst, W.M.P.: Process Mining: a research agenda. Computers and Industry 53, 231–244 (2004)
2. Jagadeesh Chandra Bose, R.P., van der Aalst, W.M.P.: Context Aware Trace Clustering: Towards Improving Process Mining Results. In: Proceedings of the SIAM International Conference on Data Mining, SDM, pp. 401–412 (2009)
3. Veiga, G.M., Ferreira, D.R.: Understanding Spaghetti Models with Sequence Clustering for ProM. In: Rinderle-Ma, S., Sadiq, S., Leymann, F. (eds.) BPM 2009. LNBIP, vol. 43, pp. 92–103. Springer, Heidelberg (2010)
4. Song, M., Günther, C.W., van der Aalst, W.M.P.: Trace Clustering in Process Mining. In: Ardagna, D., Mecella, M., Yang, J. (eds.) BPM 2008. LNBIP, vol. 17, pp. 109–120. Springer, Heidelberg (2009)
5. Bose, R.P.J.C., van der Aalst, W.M.P., Žliobaitė, I., Pechenizkiy, M.: Handling Concept Drift in Process Mining. In: Mouratidis, H., Rolland, C. (eds.) CAiSE 2011. LNCS, vol. 6741, pp. 391–405. Springer, Heidelberg (2011)
6. Bose, R.P.J.C., van der Aalst, W.M.P.: Trace Clustering Based on Conserved Patterns: Towards Achieving Better Process Models. In: Rinderle-Ma, S., Sadiq, S., Leymann, F. (eds.) BPM 2009. LNBIP, vol. 43, pp. 170–181. Springer, Heidelberg (2010)
7. Alves de Medeiros, A.K., Günther, C.W.: Process Mining: Using CPN Tools to Create Test Logs for Mining Algorithms. In: Jensen, K. (ed.) Proceedings of the Sixth Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools, pp. 177–190 (2005)