Interactive Trace Clustering to Enhance Incident Completion Time Prediction in Process Mining

Thais Rodrigues Neubauer[1], Alexandre Gastaldi Lopes Fernandes[1], Marcelo Fantinato[1] and Sarajane Marques Peres[1]

[1]*School of Arts, Sciences and Humanities – University of Sao Paulo, Arlindo Béttio, 1000, 03828-000, Sao Paulo, SP, Brazil*

Abstract

When it comes to process mining, the more singularities there are in a business process, the more human-in-the-loop strategies are needed. In order to apply discovery, compliance, profile identification, prediction or recommendation algorithms, domain experts are often involved in various tasks, such as pre-processing event logs, parameterizing algorithms and post-processing the results. However, experts' knowledge is rarely used to directly influence the internal decision making of process mining algorithms. The knowledge-oriented adjustment of the behavior of the algorithm facilitates data analysis, corrects distortions, and produces results more adherent to the expectations of a domain analyst. In this paper, we introduce the approach named *interactive trace clustering*, in which the human-in-the-loop strategy is implemented through experts' knowledge based constraint rules, modeled and merged with the decisions of the *k-Means* algorithm by means of the *Cop-k-Means* algorithm. We discuss our proposal through a proof of concept built on an incident completion time prediction problem. A real-world event log was used and the results when applying interactive trace clustering showed the usefulness of our approach.

Keywords

Trace Clustering, Interactive Clustering, Business Processes, Process Mining, Completion Time Prediction

1. Introduction

Process mining currently plays a central and strategic role in many organizations, particularly when automatic analyses of business processes are needed. As for descriptive analysis of processes, clustering strategies have been explored in the form of trace clustering [1, 2]. Profiles discovered by trace clustering provide knowledge about process particularities that facilitate the subsequent process mining tasks, contributing to achieve results more aligned with business needs [3]. However, the specialized literature shows that trace clustering algorithms do not deliver solutions that satisfactorily meet all these expectations. Appice and Malerba [4] reported results in which clustering quality and the process model quality are not always positively correlated. This kind of misalignment shows a room for trace clustering improvements.

Several technical decisions need to be made to carry out a clustering, such as different data representations, similarity functions or the clustering algorithm and its hyperparameters. Each choice may affect the revealed profiles, hence influencing the quality and usefulness of the results. Knowing which options leads to adequate profiles is challenging, especially in the context of

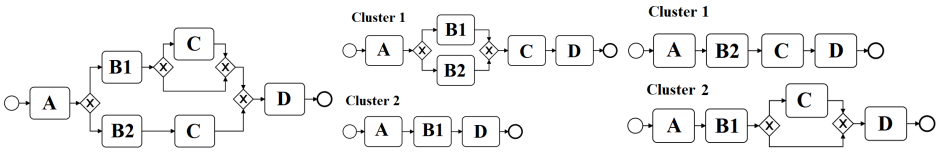
Woodstock'21: Symposium on the irreproducible science, June 07–11, 2021, Woodstock, NY

thais.neubauer@usp.br (T. R. Neubauer); aglfern@gmail.com (A. G. L. Fernandes); m.fantinato@usp.br (M. Fantinato); sarajane@usp.br (S. M. Peres)

unsupervised learning [5, 6] due to its nature of not using supervised information to validate the results. When it comes to business process analyses, even if we assume organizations are not fully aware of their actual process, experts have relevant knowledge about the organization's processes. Thus, instead of taking exclusively unsupervised assumptions, applying experts' knowledge to trace clustering might produce results more aligned with business needs.

Experts' knowledge can be addressed in clustering with *interactive* strategies, which also represent ways to implement a *human-in-the-loop* strategy. Interactive clustering refers to involving human experts in key decisions[1] of the clustering procedure [7, 8, 9]. The combination of interactive and trace clustering, which we call *interactive trace clustering*, seeks to achieve results more aligned with business goals by incorporating experts' knowledge in clustering. It is expected to reduce the harmful effects of decisions purely based on statistical or arbitrary assumptions, without causing drastic losses in the quality of the solution.

For instance, consider the business process model of Figure 1(a) and suppose we want to cluster its process instances into two clusters to investigate how they could be managed by two different departments. A possible clustering result is shown in Figure 1(b). Analyzing it, someone could argue that would be better to separate in different clusters process instances in which activity B1 occurs and instances in which B2 occurs since B1 and B2 are related to procedures commonly carried out by different departments. We could use this expert utterance as a constraint during the clustering process and get the results presented in Figure 1(c).



(a) Original (b) Resulting clustering 1 (c) Resulting clustering 2

Figure 1: How interactive clustering can be applied: (a) a base business process model; (b) a possible result for clustering process instances into two clusters; (c) a possible interactive clustering result, i.e., involving an expert in the clustering steps who suggests separating B1 and B2 into different clusters. The process models presented in this figure follow Business Process Model and Notation (BPMN).

In this context, we introduce the *interactive trace clustering* strategy and discuss its applicability through a proof of concept. We propose to implement this approach using *Cop-k-Means*, a constrained clustering algorithm [10]. The proof of concept was modeled on a problem of completion time prediction [11, 12] and evaluated from the perspective of data and domain analysts[2]. By applying a constrained clustering algorithm to the particular context of trace clustering, we developed the main contribution discussed: an interactive trace clustering strategy. Additional contributions include the presentation of a strategy to interact with experts and discussions considering evaluations from data and domain analysis perspectives[3]. This paper is organized as follows: Section 2 summarizes the theoretical background; Section 3

[1]Interactive clustering refers herein to strategies that incorporate experts' knowledge strictly in the clustering algorithm loop, i.e., disregarding pre- and post-processing.

[2]Process model quality and process analyst perspective analysis are outside the scope of this paper.

[3]This study was partially supported by CAPES (Finance Code 001).

introduces our interactive trace clustering approach; Section 4 provides the proof of concept of our approach; and Section 5 concludes the paper.

2. Theoretical Background

After analysing examples of interactive clustering application in the literature, we propose to organize the **interactive clustering** strategies in a general framework comprising four basic steps: *(1)* an initial clustering is carried out without any supervision; *(2)* then, selected information about each cluster obtained is presented to experts; *(3)* through an interaction with the experts, their knowledge is collected and used to perform a new clustering iteration; *(4)* these last two steps are re-run a finite and pre-determined number of times, or until a minimum error is reached, or until the experts inform the procedure must complete.

In this study, interactive clustering is implemented through **constrained clustering**, which is one approach to collect and apply human knowledge in clustering. One of the ways to implement this approach comprises adding, to the classic clustering procedure, *must-link/cannot-link* constraints between pairs of data points. These constraints show whether two data points must be associated with the same cluster (for *must-link* constraints) or whether two data points must be associated with different clusters (for *cannot-link* constraints). In this paper, $d_i \sim d_j$ denotes the existence of a *must-link* constraint between data points d_i and d_j, and $d_i \bowtie d_j$, the existence of a *cannot-link* constraint between data points d_i and d_j.

Constrained clustering was implemented in this study through the *Cop-k-Means* [10, 13] algorithm. The algorithm applied receives as input: (i) the data set's vector representation D, the *must-link* constraint set $m \subseteq D \times D$, and the *cannot-link* constraint set $c \subseteq D \times D$. A new iteration occurs as long as there is no convergence (we assume convergence when the centroids' displacement $\leq 1e^{-04}$). In each iteration, each data point d_i is assigned to the closest cluster C_j that does not violate the constraints in m or c; if C_j cannot be found, the clustering does not have a solution. By the end of each iteration, the centroid of each cluster is updated. Euclidean distance was used as the similarity function in this study.

Process mining and **trace clustering** rely on the concepts of events, cases, traces, and event logs [14]. An *event* e is the occurrence of a process activity at a given time. Events may be characterized by attributes such as timestamp, activity label, resource, cost etc in the way that for each event e in a universe of events $\mathscr{E}$, $\#_n(e)$ is the value of the attribute n for the event e. In addition, an event can have a label $\underline{e}$, which is associated with it through a function, called *classifier*, applied to the set of attributes of the event. In general, this label determines the context in which the process mining algorithms are carried out. A *case* corresponds to a process instance and consists of events such that each event relates exactly to a case. Cases in a universe of cases $\mathscr{C}$ may be characterized by attributes, and $\#_n(c)$ is the value of the attribute n for the case c and each case has a mandatory attribute called *trace*, $\#_{trace}(c) \in \mathscr{E}^*$. A *trace* corresponds to a finite sequence of events $\sigma \in \mathscr{E}^*$, i.e., each event appears only once in σ. Lastly, an *event log* is a set of cases $L \subseteq \mathscr{C}$ such that each event appears at most once in the entire event log [14].

In the trace clustering strategy adopted herein, *traces* are transformed to *simple traces* and the *event log* is transformed to a *simple event log* $\underline{e} = (\#_{attribute_1}(e), \#_{attribute_2}(e), ..., \#_{attribute_m}(e))$. A *simple trace* is a sequence of attribute names, and a *simple event log* is a multi-set of simple

traces [14]. For the sake of simplicity, *trace* and *event log* are used in the remainder of this text as synonyms for *simple trace* and *simple event log*. Then, *traces* are mapped to a vector space model based on the occurrence of activities in both the *trace* and the *event log*.

Appice and Malerba [4] present two strategies for building vector representations for *traces*. The first strategy is based on the presence or absence of activities on a *trace* (BIN, for *Binary*) while the second strategy is based on the frequency of activities on a *trace* (AF, for *Activity Frequency*). We added a third strategy based on the relationship between the frequency of activities on a *trace* and the number of *traces* in which the activities are present (AF-ITF, for *Activity Frequency - Inverted Trace Frequency*), in the same way followed in the well-known text data representation with TF-IDF scheme [15]. Based on the trace vector representation, similarity functions used in the classic clustering can be used to calculate the similarity between traces. Moreover, classic clustering algorithms, such as *k-Means* and *Cop-k-Means*, also apply[4].

3. Interactive Trace Clustering

To implement interactive trace clustering, we developed a constrained trace clustering-based approach comprising the four steps presented in Section 2. The first step requires the choice of a clustering algorithm by the data analyst. The fourth step depends on factors extrinsic to our approach (i.e., performance of the clustering solution in the problem under analysis and availability of experts' time). The second and third steps require more detailed discussions, since they carry decisions specifically adopted in the proposed interactive trace clustering approach.

Regarding the visualization of the clusters (Step *(2)*, cf. Section 2), the designed approach is based on the values of event attributes and the frequency with which such values occur within a cluster. The information can be viewed in tables or graphs for each resulting cluster. In these views, lines in a table or bars in a graph represent attribute values and properties in each cluster. The experts' knowledge will be collected based on the visualization of the clusters presented to them. Considering the information used to characterize the clusters and its relation with event attributes, the experts' knowledge will be indirectly collected based on the occurrence or frequency of activities in both the *trace* and the *event log*, and a mapping between clusters properties and *traces* in the *event log* is possible (Step *(3)*, cf. Section 2). In trace clustering, a data point corresponds to a *trace*. Thus, the experts' knowledge collected in terms of attribute values must be mapped to *trace*-based pairwise constraints. All *traces* involving that attribute value should be selected so that specific pairwise constraints are created for each *trace*.

Interactive clustering approaches have been widely used in many types of data mining applications [9]. In process mining, there are studies combining experts' knowledge, which was acquired independently from the trace clustering process, with the results obtained by applying trace clustering. Koninck et al. [16] proposed a semi-supervised approach combining manual profile definition made *a priori* by an expert with the trace clustering results. Weerdt et al. [17] proposed to apply the expert's knowledge through an active learning approach by using selective sampling of supervised information: a sample of *traces* is chosen to create initial clusters and the remaining *traces* are added to the clusters if the accuracy of the process model

[4]Each *trace*'s frequency is taken into consideration in the experiments presented due to the influence it may cause on the clustering results when using a partitioning clustering algorithm.

underlying each cluster is not significantly decreased.

From the studies we identified, Koninck et al. [18] is the one applying the knowledge of experts into trace clustering in a closer way of what we defined in this paper as interactive trace clustering[1]. The authors present experiments with two novel constrained trace clustering techniques to leverage expert knowledge in the form of instance-level constraints and concluded that the techniques are indeed capable of producing clustering solutions that are more justifiable without a substantial negative impact on their quality. Although constrained clustering has also been used in such a study, the authors simulated the experts' knowledge. The approach proposed herein differs from that because it includes the interactions with real-world experts and an evaluation strategy based on a specific task related to process mining.

4. Proof of Concept

In this section, we present a proof of concept to evaluate whether the application of interactive trace clustering improves the results of a specific task related to process mining, while maintaining clustering quality. We chose the application domain of incident completion time prediction. Issue tracking systems, as Incident Ticket Systems, are used in organizations to manage the various issues that come up in daily operations [19]. We have three main reasons for choosing such an application domain: **the first one** is that a key task in such context is to accurately predict the completion time of the tracked objects (the incidents) [19]. As pointed out by Folino et al. [19], clustering as a pre-processing step can help obtain more accurate time series predictors. The underlying goal is to have a set of specific predictors, one for each resulting cluster, which are more accurate than a predictor build upon all data points; **secondly**, according to Amaral et al. [11], solutions for incident completion time prediction are influenced by the attributes that characterize the incident life-cycle and their respective values. Therefore, we can hypothesize that arbitrary choices about how to use such attributes and values to build a predictor model may lead to results inferior to the results that can be obtained with conscious choices based on the knowledge of an expert; **lastly**, the execution of our proof of concept requires the availability of domain experts and, in this application domain, we had access to a set of experts with knowledge of ITIL framework, incident management process or business process management.

In the proof of concept, we used an event log extracted from a real-world business process. The event log [11] comprises events of an incident management business process and consists of 24,918 incidents and 29 descriptive attributes, extracted from an instance of the *ServiceNow*^{TM} platform used in an IT company, from March 2016 to February 2017. Each incident event is described in terms of the incident state in the incident management standard process (*incident_state*). The event description is enriched through a set of attributes that describe the incident (the process instance), such as *priority*, *reassingment_count*, etc[5]. The completion time of an incident is the difference between the attributes *closed_at* and *opened_at*.

We combined clustering and completion time prediction algorithms with experts' interaction. To evaluate, we compared the results obtained through *interactive trace clustering* with those obtained through *classic trace clustering* and without using clustering, in terms of clustering

[5]A full attributes' description is presented in https://archive.ics.uci.edu/ml

quality (data analysts' standpoint) and prediction accuracy (domain analysts' standpoint). The purpose of this proof of concept is to show the feasibility and usefulness of our approach. We consider a successful proof of concept if we obtain improvements in the accuracy of the forecast of the time of completion of incidents without being accompanied by a great degradation in the quality of clustering. The results[6] are discussed in Sections 4.2 and 4.3.

4.1. Procedures and Resources

Figure 2 shows the two phases of this proof of concept: *(i)* preprocessing and clustering setup and *(ii)* run of interactive trace clustering.

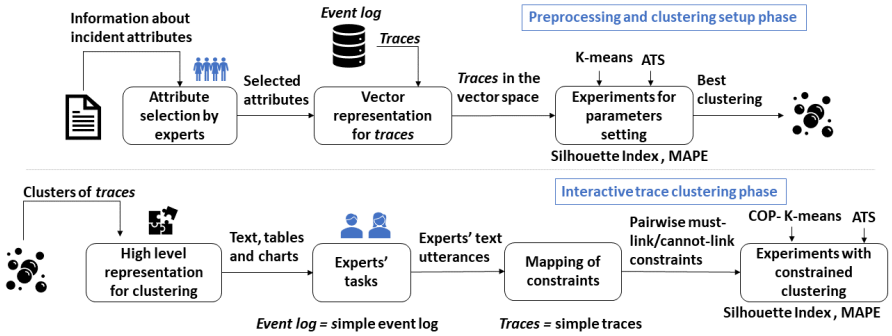


Figure 2: Proof of concept: overview

In the first phase, six domain experts[7] took part in. Each expert selected from one to five descriptive attributes (of 29) relevant to implement an incident completion time predictor. The selected attributes were used to create *trace* vector representations (cf. Section 2). In the vector space, the *traces* were submitted to *k-Means* (with k = 3, 5, 10 and Euclidean distance). As a result, clusters of *traces* were created, *event sublogs* were built from each cluster, and then Annotated Transition Systems [20] (ATS) were built on such *sublogs* to implement completion time predictors [11][8]. Clustering and prediction results were evaluated from two points of view: through the Silhouette Index (SI) [21], to analyze the quality of similarity among the *traces* allocated in the same cluster; and through the Mean Absolute Percentage Error (MAPE) [22], to analyze the quality of the prediction obtained for each *sublog*. As a result of this first phase, we obtained the most appropriate combination for *trace* representation and number of clusters to conduct the second phase of this proof of concept.

In the second phase, interactive trace clustering was applied based on the clusters resulting from the first phase. For this, tabular and graphical representations were elaborated to present and explain the context of each group of incidents to four experts[9]. The experts analyzed the

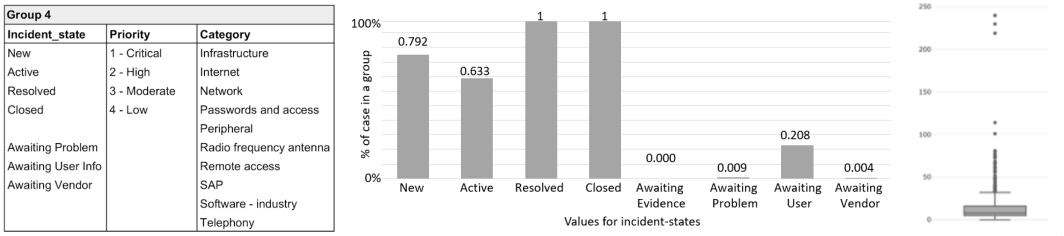
[6]The codes used are available in https://github.com/pm-usp/ITC-completion-time-prediction

[7]Software industry practitioners with background in BPMN, ITIL framework, and incident management process.

[8]ATS was chose to be used herein because of its simplicity in terms of parameters to be set: we used the abstraction "set" and horizon = 1 [20].

[9]Two senior software industry practitioners and two senior academic researchers with background in business process or incident management.

context of each group of incidents and performed four tasks. Thus, the experts: *(i)* proposed improvements in the incident profile represented by each group, analyzing the values of the incident attributes, based on information similar to Figure 3(a); *(ii)* proposed improvements to the distribution of incidents in groups based on information similar to Figure 3(b); *(iii)* analyzed the quality of the individual groups based on information similar to Figures 3(b;c); and *(iv)* analyzed the overall organization of the groups based on information similar to Figures 3(b;c). The experts' responses were provided as text utterances. These utterances were manually interpreted and mapped to pairwise *must-link/cannot-link* constraints to apply constrained clustering (with *Cop-k-Means*). From the results of this constrained clustering, the procedures to build *sublogs* and predictors (ATS) were performed again. In this phase, the SI and MAPE were applied, as the goal was to verify whether the experts' knowledge contributed to generate better clusters, i.e., *sublogs* more adequate to build incident completion time predictors.



(a) (b) (c)

Figure 3: Examples of graphs to present the clustering results to the experts: (a) values of attributes which occurs in a group; (b) the bars represent the count of each attribute value in a group; (c) boxplot of the incidents' duration in a group. We avoided using the word *cluster* when interacting with the experts.

4.2. First Phase: Trace Clustering Results

In the first phase, all experts chose the attributes *category* and *priority*, and half of the experts chose the attribute *incident_state*. This result corroborates with the attributes stated by Amaral et al. [11] as the most likely choices when following the ITIL framework. These three attributes were used in a composite way to characterize the events in a trace, i.e., $\underline{e} = (\#_{incidente\_state}(e), \#_{category}(e), \#_{priority}(e))$. Then, the *traces* were mapped to the vector representations BIN, AF and AF-ITF and the *k-Means* algorithm was applied to such vector space models. The resulting clusterings were evaluated in terms of SI and MAPE (Table 1). Both measures were obtained as the average on the k clusters. For MAPE, the standard deviation was calculated as a way to express the uniformity in the quality of the set of clusters.

Different quality measures lead to different conclusions about the best combination of trace vector representation and number of clusters. While the SI points to the AF-ITF representation, MAPE points to the AF representation. However, the MAPE superiority of AF over AF-ITF is not as relevant as the SI superiority of AF-ITF over AF. The results on number of clusters are also not conclusive: the SI points to a larger number of clusters while MAPE points to k = 5 for BIN and AF and to k = 10 for AF-ITF. The choice of the number of clusters for the second phase of the proof of concept should also consider that a large number of clusters would

Table 1
Clustering evaluation based on SI and MAPE

	Silhouette Index			**MAPE - ATS** μ (σ)		
	$k = 3$	$k = 5$	$k = 10$	$k = 3$	$k = 5$	$k = 10$
BIN	0.17	0.18	0.28	3.38 (1.48)	2.86 (1.76)	3.60 (3.83)
AF	0.07	0.17	0.23	**2.71** (0.76)	**2.38** (1.30)	3.16 (2.52)
AF-ITF	**0.52**	**0.62**	**0.70**	4.10 (3.73)	3.41 (2.78)	**3.10** (2.72)

increase the experts' effort. Thus, considering the analysis of the evaluation measures and the human-in-the-loop context, the parameters chosen were the trace vector representation AL-ILF and k = 5.

4.3. Second Phase: Interactive Trace Clustering Results

The second phase was conducted using the clustering results referring only to the trace representation AL-ILF and $k = 5$. From the interaction with the experts, we obtained a set of utterances, which were consolidated in three high level rules, and then manually mapped to five *must-link/cannot-link* conceptual constraints, as shown in Table 2. These conceptual constraints were used to establish the three set of pairwise constraints: $C_{state}, C_{priority}, C_{category}$.

Table 2
Constraints obtained through interaction with experts

High level rule #1: "*Incidents that pass through 'awaiting...' states should be grouped together as they may last longer because of the waiting time.*" **Constraint *must-link* on values of attribute *incident_state*:** {*awaiting evidence*, *awaiting problem*, *awaiting user*, *awaiting vendor*} ~ {*awaiting evidence*, *awaiting problem*, *awaiting user*, *awaiting vendor*}
High level rule #2: "*Incidents with high (1 and 2), medium (3) and low (4) priorities should be organized into different groups.*" **Constraints *cannot-link* on values of attribute *priority*:** {*priority 1*, *priority 2*} ⋈ {*priority 3*}; {*priority 1*, *priority 2*} ⋈ {*priority 4*}; {*priority 3*} ⋈ {*priority 4*}
High level rule #3: "*Incidents involving requests for 'password and access', 'remote access' and 'office pack', which are easily remotely resolved, must be in the same group.*" **Constraint *must-link* on values of attribute *category*:** {*password and access*, *remote access*, *office pack*} ~ {*password and access*, *remote access*, *office pack*}

To evaluate the effect of the interactive trace clustering, with each set of pairwise constraints separately, on predicting incident completion time, the average MAPE (and the corresponding standard deviation) obtained for every five ATS predictors built for each clustering was contrasted with: *(i)* the MAPE of the ATS predictor built on the full *event log* (i.e., without using clustering); and *(ii)* the average MAPE (and corresponding standard deviation) obtained for the five ATS predictors built on the five clusters resulting from the classic trace clustering. These contrasts are presented in Figure 4. The best results are the ones with lower average MAPE and the corresponding standard deviation, i.e., closer to the origin.

We explored the *Cop-k-Means* algorithm stability by executing it ten times for each set of pairwise constraints. A total of 30 executions of *Cop-k-Means* were carried out. Ten executions

of *k-Means* were also conducted, using the same parameters chosen in the first phase. As shown in Figure 4 through the MAPE average variation, there is an instability among results of classic trace clustering executions. The result used in the first phase (marked as [×]) is the closest to the average point of the classic trace clustering executions (marked as faded [⋆]). Even though the average point of classic trace clustering executions (faded [⋆] symbol) presented a MAPE average similar to the MAPE obtained using the full event log, 70% of the executions (three out of ten) presented MAPE average smaller than the obtained using the full *event log*.

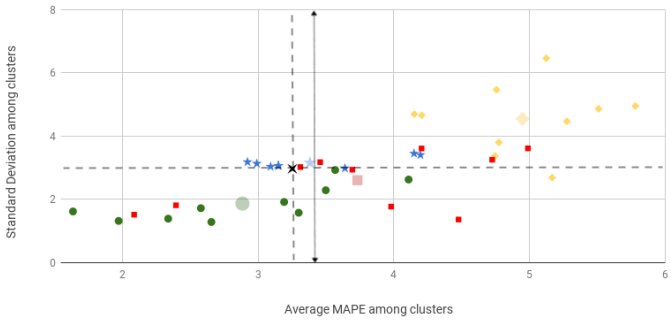


Figure 4: Average MAPE and standard deviation of the ATS predictors built on the results of interactive trace clustering, classic trace clustering of the first phase [×], and the result without clustering [⟷]). [★] refers to classic trace clustering executions; [●] refers to interactive trace clustering executions using $C_{category}$ constraints; [■] refers to using C_{state} constraints; [♦] refers to using $C_{priority}$ constraints. The faded symbols represent the average point for executions represented by that same symbol.

Considering the results corresponding to the application of each set of pairwise constraints, we observed the following: (a) when applying $C_{category}$, 70% of executions presented a MAPE average smaller than the obtained using the full *event log* and than the average point of classic trace clustering executions. Standard deviations were smaller than the obtained with classic trace clustering executions, thus representing more uniformity in the quality of the results; (b) the application of C_{state} resulted in a more disperse distribution of MAPE averages and standard deviations. Even though some average MAPE reductions can be observed when comparing this results to the MAPE obtained from the full *event log* and from classic trace clustering executions, the average point of $C_{category}$ results shows this set of pairwise constraints might not be adequate for the desired increase of prediction's accuracy; (c) the results obtained with the application of $C_{priority}$ led to the worst results, with no result with smaller average than the one obtained with the full *event log* or with the result of the first phase. The standard deviation values were all greater than the obtained with classic trace clustering. Thus, this set of pairwise constraints is not adequate. We suppose this issue is related to the existent issue in IT services' environments: priority setting by who inserted the incident details in the system.

The results of classic and interactive trace clustering executions from the data analyst's perspective, in terms of SI, are presented in Figure 5. The SI values of classic trace clustering results reveal lower instability among executions when compared to the values obtained among the executions of each set of pairwise constraints. For the interactive trace clustering results, corroborating with the analysis of MAPE, better average SI values were observed when applying $C_{category}$ and worse results were observed applying $C_{priority}$. These SI results suggest that as

long as the right set of pairwise constraints is selected, it is possible to implement interactive trace clustering without significant quality damage from the data analyst's perspective.

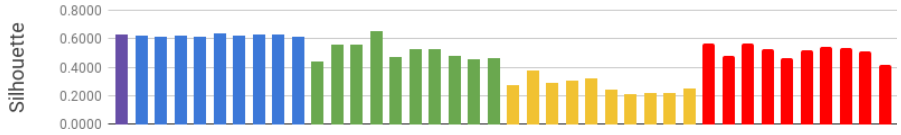


Figure 5: Average SI of the results of classic trace clustering executions and interactive trace clustering executions. [■] refers to the result of the first phase and [■] refers to the other classic trace clustering executions; [■] refers to *Cop-k-means* executions with constraints related to attribute *category*; [■] constraints related to attribute *priority*; [■] related to attribute *incident_state*

Following the interactive trace clustering concept, we could present the results of this phase to experts and collect more inputs to further refine the clustering result. Unfortunately, due to the experts' unavailability, another phase is not in the scope of this work.

Table 3 shows the contrasts of the results presented in the first phase and in the interactive trace clustering phase, considering the execution with the lowest MAPE average. Due to the relation of the applied set of pairwise constraints with the category attribute, the contrast is presented with ten most common values for the category attribute and the MAPE average value for each group. From this kind of analysis, the experts could, for example, suggest that in addition to the *must-link* constraints in $C_{category}$, there should be *cannot-link* constraints between {*password and access*, *remote access*, *office pack*} ⋈ {*SAP*, SAP ECC}.

Table 3
Category attribute values and MAPE average (μ) in each cluster of the result of phase 1 in contrast with the result of phase 2. The list of category values is presented in a descending order of value occurrence in the incidents of the group and the list of groups is in a descending order of μ. SW = software

	Group 1	**Group 2**	**Group 3**	**Group 4**	**Group 5**
Phase 1	Desktop	Others	Peripheral	Infrastructure	Desktop
	Internet	**Passwords**	Radio frequency	Internet	Internet
	Notebook	**Remote access**	SW - industry	Network	**Office pack**
	Passwords	SAP	SAP	**Passwords**	**Passwords**
	Printer	SAP CRM	Telephony	Peripheral	**Remote access**
	SAP ECC	SAP ECC	**Passwords**	Radio frequency	SAP
	SAP ECC EN	SAP ECC EN	Internet	**Remote access**	SAP CRM
	MAPE $\mu = 8.16$	*MAPE* $\mu = 3.61$	*MAPE* $\mu = 2.49$	*MAPE* $\mu = 1.35$	*MAPE* $\mu = 0.65$
Phase 2	SAP	Software	Database	Desktop	Internet
	Office pack	SAP ECC EN	Software	Software	Software
	Passwords	SW - industry	SW - industry	Peripheral	Network
	Printer	SAP	Others	Network	Infrastructure
	Network	SAP ECC	Peripheral	Printer	Notebook
	Telephony	SAP CRM	SAP ECC EN	Notebook	Security
	Remote access	Internet		Server	Radio frequency
	MAPE $\mu = 4.35$	*MAPE* $\mu = 1.92$	*MAPE* $\mu = 0.74$	*MAPE* $\mu = 0.71$	*MAPE* $\mu = 0.46$

4.4. Cost of the proposed strategy

Based on the proposed mapping strategy, the number of constraints grows exponentially in relation to the number of affected *traces*. To evaluate the effect of the quantity of constraints on the algorithm runtime, we defined the *data space rate* concept, which refers to the choice of a percentage of the set of all *traces* to be clustered. For instance, if the expert says all incidents in which "activity *A*" occurs must be assigned to the same group, an alternative seeking algorithm runtime reduction could be to add *must-link* constraints to 50% of the *traces* in which "activity *A*" occurs. Such *traces* can be chosen by random. We refer this selection as *space* since the *traces* to be selected should be those associated with data points positioned in different regions of the vector space model. The obtained results revealed low correlation between the quantity of pairwise constraints and runtime. Our assumption is that depending on the order of data assignment to clusters and the type of the pairwise constraints, the runtime can actually be accelerated by the constraints which narrow down, or even totally define, the options of the assignment of data points to clusters.

5. Final remarks

In this paper, we introduced the interactive trace clustering as a way to use experts' knowledge to improve process mining results. A proof of concept on completion time prediction showed the potential of the approach to obtain better results from the domain analysts' standpoint, without significant losses from data analysts' standpoint. The use of interactive clustering brings more meaningful trace clustering results, given the possibility of replacing or complementing unsupervised assumptions with experts' knowledge. Our experiments showed the advantage of interactive trace clustering depends on the choice of the set of pairwise constraints. However this decision would involve the process stakeholders' knowledge which might guide decision making more aligned with business goals. Even if the stakeholders are not fully aware of the underlying business process, knowledge of how they perceive the process is useful. Results obtained from a successful proof of concept provide evidence for analytical generalization. The experts' time consumption is a challenge to overcome in order to execute other interactive phases (similar to section 4.3), including experiments with different combination of the constraints extracted by the end of each phase. Another future work identified is conducting experiments that enable comparing our results to the ones presented in Koninck et al. [18].

References

[1] A. R. C. Maita, L. C. Martins, C. R. L. Paz, L. Rafferty, P. C. K. Hung, S. M. Peres, A systematic mapping study of process mining, Enterp. Inf. Syst. 12 (2017) 1–45.

[2] C. d. S. Garcia, A. Meincheim, E. R. F. Junior, M. R. Dallagassa, D. M. V. Sato, D. R. Carvalho, E. A. P. Santos, E. E. Scalabrin, Process mining techniques and applications - A systematic mapping study, Exp. Sys. Appl. 133 (2019) 260–295.

[3] M. de Leoni, W. M. P. van der Aalst, M. Dees, A general process mining framework for

correlating, predicting and clustering dynamic behavior based on event logs, Inf. Syst. 56 (2015) 235–257.
[4] A. Appice, D. Malerba, A co-training strategy for multiple view clustering in process mining, IEEE Trans. Serv. Comput. 9 (2016) 832–845.
[5] P. Awasthi, M. F. Balcan, K. Voevodski, Local algorithms for interactive clustering, J. of Mach. Learn. Res. 18 (2017) 1–35.
[6] Y. Lei, D. Yu, Z. Bin, Y. Yang, Interactive k-means clustering method based on user behavior for different analysis target in medicine, Comput. Math. Methods Med. 2017 (2017).
[7] Y. Hu, E. E. Milios, J. Blustein, Interactive document clustering with feature supervision through reweighting, Intell. Data Anal. 18 (2014) 561–581.
[8] F. Schwenker, E. Trentin, Pattern classification and clustering: A review of partially supervised learning approaches, Pattern Recognit. Lett. 37 (2014) 4–14.
[9] T. R. Neubauer, S. M. Peres, M. Fantinato, X. Lu, H. A. Reijers, Interactive clustering: a scoping review, Artif. Intell. Rev. 54 (2021) 2765–2826.
[10] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl, Constrained k-means clustering with background knowledge, in: Int. Conf. on Mach. Learn., 2001, p. 577–584.
[11] C. A. L. Amaral, M. Fantinato, H. A. Reijers, S. M. Peres, Enhancing completion time prediction through attribute selection, in: Lecture Notes in Bus. Inf. Process., volume 346, Springer, 2019, pp. 03–23.
[12] G. Blokdijk, I. Menken, ITIL V3: How to Develop, Implement and Enforce ITIL V3 Best Practices, 2nd ed., Emereo Pty, London, 2009.
[13] T. Rutayisire, Y. Yang, C. Lin, J. Zhang, A modified cop-kmeans algorithm based on sequenced cannot-link set, in: Rough Sets and Knowl. Technol., Springer, 2011, pp. 217–225.
[14] W. M. P. van der Aalst, Process Mining - Data Science in Action, Springer, 2016.
[15] G. Salton, A. Wong, C. S. Yang, A vector space model for automatic indexing, Commun. ACM 18 (1975) 613–620.
[16] P. D. Koninck, K. Nelissen, B. Baesens, S. van den Broucke, M. Snoeck, J. Weerdt, An approach for incorporating expert knowledge in trace clustering, in: Int. Conf. on Adv. Inf. Syst. Eng., 2017, pp. 561–576.
[17] J. de Weerdt, S. vanden Broucke, J. Vanthienen, B. Baesens, Active trace clustering for improved process discovery, IEEE Trans. Knowl. Data Eng. 25 (2013) 2708–2720.
[18] P. D. Koninck, K. Nelissen, S. vanden Broucke, B. Baesens, M. Snoeck, J. de Weerdt, Expert-driven trace clustering with instance-level constraints, Knowl. Inf. Syst. 63 (2021) 1197–1220.
[19] F. Folino, M. Guarascio, L. Pontieri, A framework for the discovery of predictive fix-time models, in: Int. Conf. on Enterp. Inf. Syst., 2014, p. 99–108.
[20] W. M. P. van der Aalst, M. Schonenberg, M. Songa, Time prediction based on process mining, Inf. Syst. 36 (2011) 450–475.
[21] P. J. Rousseeuw, Silhouettes: A graphical aid to the interpretation and validation of cluster analysis, J. Computational and Appl. Math. 20 (1986) 53–65.
[22] J. S. Armstrong, F. Collopy, Error measures for generalizing about forecasting methods: Empirical comparisons, Int. J. Forecasting 8 (1992) 69–80.