Identifying Cohorts: Recommending Drill-Downs Based on Differences in Behaviour for Process Mining

Sander J. J. Leemans[1], Shiva Shabaninejad[2], Kanika Goel[1], Hassan Khosravi[2], Shazia Sadiq[2], and Moe T. Wynn[1]

[1] Queensland University of Technology, Brisbane, Australia
[2] University of Queensland, Brisbane, Australia
{s.leemans,k.goel,m.wynn}@qut.edu.au, {s.shabaninejad, h.khosravi}@uq.edu.au, shazia@itee.uq.edu.au

Abstract. Process mining aims to obtain insights from event logs to improve business processes. In complex environments with large variances in process behaviour, analysing and making sense of such complex processes becomes challenging. Insights in such processes can be obtained by identifying sub-groups of traces (cohorts) and studying their differences. In this paper, we introduce a new framework that elicits features from trace attributes, measures the stochastic distance between cohorts defined by sets of these features, and presents this landscape of sets of features and their influence on process behaviour to users. Our framework differs from existing work in that it can take many aspects of behaviour into account, including the ordering of activities in traces (control flow), the relative frequency of traces (stochastic perspective), and cost. The framework has been instantiated and implemented, has been evaluated for feasibility on multiple publicly available real-life event logs, and evaluated on real-life case studies in two Australian universities.

Keywords: Process mining · drill-down recommendation · filter recommendation · stochastic comparative process mining

1 Introduction

In organisational processes, users and computers interact with information systems to handle cases such as orders, claims and applications. These interactions are logged as events, and process mining utilises such event data in the form of an event log to gain evidence-based insights in the structure and performance of organisational processes. In order to gain insights in a process, typically first a process model is discovered from a log, then this model is evaluated and finally additional perspectives such as performance, costs or resources are projected on the process model such that analysts can identify potential problems and gain insights [8]. Many processes are highly variable, which may be due to geographically different entities executing the process, different customers interacting with it, or different backgrounds of students being involved

in it. Analysing highly variable processes challenges analysts, as conclusions and insights might get watered down or might not hold in all variants [7,5].

To address high variability in processes, in a typical process mining project, discussions with stakeholders would be required to identify a set of key attributes, after which these attributes and their values (*trace features*) can be used to split the log into *cohorts*. For instance, the trace $\langle$receive claim, decide claim$\rangle \left[\begin{smallmatrix} ID & 1234 \\ amount & 10000 \end{smallmatrix} \right]$ has two trace attributes, and a trace feature of amount ≥ 5000 applies to this trace. These cohorts can then be analysed separately (*drill down*) or compared to one another [21,7]. To reduce variability, trace features should be chosen such that behavioural variability within cohorts is minimised, while between cohorts it is maximised. We refer to this technique as *cohort identification*, and automation makes it more objective, enables explorative approaches that might reveal yet-unknown cohorts, and increases feasibility. To the best of our knowledge, cf. data mining [9], no process mining techniques were published that recommend trace features for drill downs based on behaviour.

In this paper, we introduce a new framework called Cohort Identification (CI) that takes a log as input and outputs a landscape of sets of trace features which maximise between-cohort behavioural variability. Figure 1 shows an overview of the framework: first, CI elicits trace features from the log. Second, CI combines them into feature sets exhaustively and, for each set, quantifies the differences between the cohort defined by the feature set and the other traces in the log. CI can take many aspects of behaviour into account, such as the ordering of activities in traces (control flow), cost of executions, and relative frequency of traces (stochastic perspective [12]). We evaluate CI on feasibility using real-life public event logs, and we illustrate its applicability using two case studies.

The remainder of this paper discusses related work in Section 2, introduces existing concepts in Section 3, introduces the framework and an open source implementation in Section 4; evaluates our approach using 21 publicly available logs and two illustrative case studies in Section 5, and concludes in Section 6.

2 Related Work

In this section, we first discuss three types of techniques (trace clustering, concept drift detection, cohort identification) to reduce variance in logs. Second, we discuss how our approach complements process comparison techniques.

Trace Clustering aims to find groups of structurally similar traces, such that these groups can be studied in isolation [6,21,23]. Secondary, the relation between these groups and trace attributes can be analysed using standard data
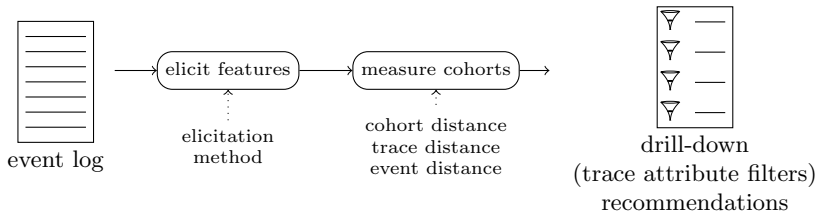

Fig. 1: Overview of the Cohort Identification (CI) framework.

2

mining techniques. For instance, a recent trace clustering approach [17] clusters traces based on control flow and performance patterns, and recommends clusters of traces to users, after which users can inspect KPIs of selected clusters. While trace clustering techniques take similar behaviour and search for relations with trace features, our approach does the opposite: it takes trace features and searches for relations with behaviour. Trace clustering inherently cannot consider the frequency of trace variants within groups, typically do not provide an explanation to justify their grouping ("black box"), provide little additional insights into the log, and are challenging to use for drill-down recommendations.

Concept Drift Detection aims to identify points in time where the process changed. The behaviour before and after the drift point might have a lower variance, and the changes can be studied comparatively (e.g. [15]).CI could be seen as a generalisation of concept drift techniques as it can detects changes in behaviour related to trace features, including when traces happened.

Cohort Identification [2] uses decision points in a process model and event attribute clustering to identify groups of traces, and performs significance tests to limit the reported groups. In contrast with our approach, this technique requires a process model and operates on event features (rather than trace features), however it would be interesting to extend CI with similar significance tests in future work.

Attribute clustering techniques first cluster the features of traces, after which the corresponding groups of traces can be studied in isolation. For instance, in [14], an extensive framework for correlating, predicting and clustering is proposed. This framework supports the splitting of logs based on extensive analyses on event features, however has a different focus than CI: event vs trace features and clustering rather than recommending drill downs.

To the best of our knowledge, no techniques have been published that recommend trace features for drill downs in process mining, based on behaviour. In addition: none of the mentioned techniques considers the relative frequency of behaviour (i.e. the stochastic behaviour recorded in logs) as a source of behavioural differences.

Complementing: Process Comparison. The CI framework recommends feature sets that relate to differences in stochastic behaviour, however does not explain what these differences are, for which a process comparison technique can be used. In a recent literature review of such techniques [22], three types of techniques were identified: discriminative (pattern extraction), generative (comparison through process models) [21] and hybrid (a combination of both). For instance, differences between logs have been visualised on their transition systems [3] or compared simply by hand [20].

Process cubes provide skeletons for process comparison by selections (cells) of traces based on trace features, and enable slicing and dicing the log to find cohorts of interests. For instance, in [1,4], process cubes were used to study differences in students' online learning activities (similar to one of our case studies). CI recommends *which* cells might be of interest, based on their behaviour, without having to resort to process discovery techniques.

CI acts as an enabler for these techniques, making them applicable to a single log cf. at least two logs. Finally, these techniques could *also* be used within CI, if they return a number expressing the equivalence between logs.

3   Preliminaries

In this section, we introduce existing concepts.

3.1   Event logs

An *event log* or *log* L is a collection of traces. A *trace* represents a case handled by the business process, and is a sequence of events. A trace is annotated with *trace attributes*, which represent properties of the trace. An *event* represents the execution of a process step (an *activity*). An event is annotated with *event attributes* that denote properties of the event. For instance, the trace

$$\langle \text{receive claim} \begin{bmatrix} date & \text{20-02-20} \\ resource & online \end{bmatrix}, \text{ decide claim} \begin{bmatrix} date & \text{21-02-20} \\ resource & Dave \\ decision & approved \end{bmatrix} \rangle \begin{bmatrix} ID & 1234 \\ amount & 10000 \\ gender & F \end{bmatrix}$$

consists of a single trace, which consists of two events, indicating the execution of two process steps ("receive claim" and "decide claim"). This trace has three trace attributes: an ID, a claim amount and the gender of the claimant. The first event has two event attributes: a date that it was executed and the resource that executed it. Given an event $e \begin{bmatrix} a_1 & v_1 \\ \dots & \dots \\ a_n & v_n \end{bmatrix}$, we write $e^{a_1} = v_1$ to retrieve the value of an attribute that e is annotated with. If e is not annotated with an attribute x, then we define $e^x = \bot$. Retrieving trace attributes is similar.

3.2   Cohorts

Let L be a log, and let a be a trace attribute. Then, let $\mathcal{F}^a$ denote all the values of a in L: $\mathcal{F}^a = \{v = t^a \mid t \in L \wedge v \neq \bot\}$. We refer to the combination of a trace attribute a and a range of its values $F \subseteq \mathcal{F}^f$ as a *trace feature* a^F. Given a log L, we refer to the sub-log of traces that have trace feature a^F as the *projection* $L|_{a^F}$. For instance, let $L = [\langle g_1, \dots g_n \rangle^{[amount\ 100]}, \langle h_1 \dots h_m \rangle^{[amount\ 50]}]$ be a log. Then, $L|_{amount \geq 70} = [\langle g_1, \dots g_n \rangle^{[amount\ 100]}]$. We refer to such a sub-log of L as a *cohort* of L.

4   A Framework for Cohort Identification (CI)

Given a log, CI aims to recommend trace features to drill down on, that is, to identify cohorts whose behaviour differs considerably from the other traces in the log in terms of process, frequency of paths and event attributes. The main input of CI is a log, and its output is list of recommendations (that is, a list of feature sets with their cohort distances).

Figure 1 shows an overview of the framework, while Algorithm 1 shows its pseudocode. First, trace features are elicited from the log using an elicitation method E. Second, for each sub-set of features of size at most k and of which the corresponding sub-logs are large enough (according to a cohort imbalance threshold α), a distance between the cohorts is computed, using a cohort distance δ_c parameterised with a trace distance δ_t, which in turn is parameterised with an event distance δ_e. Instantiating the framework requires the mentioned functions E, δ_e, δ_t, δ_c and D. In the remainder of this section, we discuss the steps of CI in more detail and we introduce an instantiation.

Algorithm 1 Cohort Identification framework

1: **procedure** $\text{CI}_{E,\delta_c,\delta_t,\delta_e,D}(\log L$, cohort imbalance threshold α, max feature set size k, correction repetition φ)
2:     $F \leftarrow E(L)$ $\triangleright$ elicit features
3:     $M \leftarrow \emptyset$
4:     **for** $S \subseteq F^*$ such that $|S| \leq k$ **do** $\triangleright$ for each feature set
5:         $L_1, L_2 \leftarrow L|_S, L \setminus L|_S$
6:         **if** $\min(|L_1|, |L_2|) \geq \alpha \times |L|$ **then** $\triangleright$ prune too-small cohorts
7:             $d \leftarrow \dfrac{\delta_{c(\delta_{t(\delta_e)})}(L_1, L_2)}{\frac{1}{\varphi}\sum_{L_1', L_2' \varphi \text{ randomly from} L \text{ s.t. } |L_1'| \approx |L_1|} \delta_{c(\delta_{t(\delta_e)})}(L_1', L_2')}$
8:                                   $\triangleright$ compute corrected cohort distance
9:             $M \leftarrow M \cup \{(S, d)\}$
10:        **end if**
11:     **end for**
12:     **return** M
13: **end procedure**

4.1 Feature Elicitation

The first step of CI is to obtain a collection of features from the log. Trace attributes with literal values straightforwardly yield one feature for each value, or could be clustered for reduced complexity. Numerical trace attributes need to be discretised, after which a feature can be added for each discrete value. The framework supports any discretisation, for instance using numerical value clustering or by using quartiles, though it is important that sufficiently large groups of traces remain.

Our instantiation elicits features for numerical attributes by discretising them in two bins, separated by the median. That is, let L be a log and let a be a numerical trace attribute in L. Then, two features are added: $a^{\geq m}$ and $a^{<m}$, in which m is the median of $\mathcal{F}^a$. Finally, for any typed attribute a, a feature $a^\perp$ is added, which expresses that a is missing.

4.2 Measure Cohorts

Second, CI aims to find the cohorts that differ the most from the other traces in the log, in terms of process, frequency of paths and other event attributes. To this end, it considers all possible feature sets S that can be made of combinations of the elicited features. Each such feature set S defines two cohorts: one cohort of traces that possess each feature in S ($L_1 = L|_S$) and the other cohort of traces that do not possess any feature ($L_2 = L \setminus L_1$).

In the cohort measuring step, CI measures the distance between pairs of such L_1 and L_2, using event and trace distance measures:

Definition 1 (Distance measures). *Let δ_e be an* event distance *measure, such that for all events v, v' it holds that $0 \leq \delta_e(v, v') \leq 1$. Let $\delta_{t(\delta_e)}$ be a* trace distance *measure, such that for all traces u, u' it holds that $0 \leq \delta_{t(\delta_e)}(u, u') \leq 1$. Let L, L' be logs. Then, $0 \leq \delta_{c(\delta_{t(\delta_e)})}(L, L') \leq 1$ is a* cohort distance *measure.*

These measures are parameters to CI, and can take any event attribute into account, which makes the CI flexible.Thus, any event, trace and (aggregated)

log attribute could contribute to the cohort distance measure given appropriate distance measures. Still, instantiations can opt not to if desired.

On small cohorts, distance measures may be meaningless, as too-small cohorts merely represent outliers. Also, as the number of possible feature sets is exponential in the number of features, CI takes a cohort imbalance threshold $0 \leq \alpha \leq \frac{1}{2}$, which sets the maximum imbalance in the two cohorts: if either $|L_1| < \alpha \times |L|$ or $|L_2| < \alpha \times |L|$, then the feature set S is discarded.

In our instantiation, we use the Earth Movers' Stochastic Conformance (EMSC) [12] as the cohort distance measure. EMSC considers both logs as piles of earth, and computes the effort to transform one pile into the other, in terms of the number of traces to be transformed into other traces times the trace distances of these transformations. Thus, the stochastic perspective is inherently taken into account by EMSC. Another option for δ_c could be to discover two process models, and cross-evaluating these models with the cohorts [21], although this has the downsides of incorporating model discovery trade-offs, asymmetry of the cohort distance measure, and the inability to include the stochastic perspective.

For the trace distance, as in [12], our instantiation uses the normalised Levenshtein distance, which expresses the minimum cost in terms of insertions, deletions and swaps of events to transform one trace into the other. Normalisation is achieved by dividing the minimum cost by the maximum possible cost, which is the length of the longest trace. The cost of insertions, deletions and swaps of events is determined by the event distance. Where EMSC uses unit event distances, we generalise this to any event distance measure δ_e, thus enabling analysts to compare traces based on any event attribute. Our instantiation allows a user to choose several event attributes, and uses a generic event distance measure for each attribute, based on whether the attribute is textual or numerical.

For a textual attribute a (δ_{e_T}) and a numerical attribute b (δ_{e_N}), in which m is the difference of the minimum and maximum of b over the entire log:

$$\delta_{e_T}(v, v') = \begin{cases} 0 & \text{if } v^a = v'^a \\ 1 & \text{otherwise} \end{cases} \qquad \delta_{e_N}(v, v') = \begin{cases} \frac{|v^b - v'^b|}{m} & \text{if } v^b \neq \perp \wedge v'^b \neq \perp \\ 0 & \text{if } v^b = \perp \wedge v'^b = \perp \\ 1 & \text{otherwise} \end{cases}$$

An extension would be to choose multiple attributes and taking their weighted average. Even though any event attribute can be used, one should be careful to avoid the curse of dimensionality: when the number of involved attributes increases, the expected overall distance between arbitrary events approaches 0, which could render the comparison less useful.

4.3 Corrected Cohort Distance

For each feature set S and cohorts L_1, L_2 of log L, a cohort distance measure $\delta_{c(\delta_{t(\delta_e)})}(L_1, L_2)$ is computed, which is parameterised with a trace measure function δ_t, which in turn is parameterised with an event measure function δ_e.

A log with a high variety of traces will in general have higher cohort distances than a log with a low variety of traces, as cohorts of the first will inherently differ more than cohorts of the second. To correct for this, CI scales the computed cohort distance for variance in the log as follows: the traces of L are randomly divided over sub-logs L_1' and L_2', such that $|L_1'| \approx |L_1|$. The distance

$\delta_{c(\delta_{t(\delta_e)})}(L_1', L_2')$ between these sub-logs is measured, the procedure is repeated φ times and the average cohort distance is taken as the *corrected cohort distance* (see line 8 of Algorithm 1). Intuitively, the corrected cohort distance shows the *gain in information* about process behaviour of a feature set S: given L, a value of 0 indicates that S provides no extra information over a random division of L, while a value of 1 indicates that S fully distinguishes all behaviour of L.

The output of the cohort measuring step is a list of feature sets annotated with their corresponding corrected cohort distance, sorted on this distance. That is, the feature set that relates the most to differences in stochastic behaviour, thus providing the most information, is on top (see an example in Section 5).

4.4 Implementation

CI and our instantiation of it have been implemented as a plug-in of the ProM framework (see http://promtools.org), and are open source https://svn.win.tue.nl/repos/prom/Packages/CohortAnalysis/Trunk, SVN revision 43272. On top of the trace attributes present in the log, CI considers the total trace duration as well. To save time computing the corrected cohort distance $\delta_{c'}$, our implementation caches the cohort distance between randomly divided sub-logs of 10 sub-log sizes, rather than for each feature set separately.

5 Evaluation

We evaluate CI and our instantiation threefold. First, we illustrate its feasibility and use on publicly available logs. Second, we study its explorative and question-driven capabilities, and its embedding in process mining projects in two case studies, with a brief empirical evaluation with stakeholder feedback.

5.1 Public Logs

To illustrate our instantiation, we applied it to 21 real-life publicly available event logs. We applied CI to each of them with the maximum feature set size (k) ranging from 1 to 5. The results show that our implementation of CI is feasible on real-life logs for small k (for $k = 1$, 18/21 logs finished in 74 seconds or less, with a maximum of 36 hours; the maximum feasible number of feature sets was around 10^7). The experimental results indicate that while the number of activities does not seem to have an influence on the run time, an exponentially increasing number of feature sets does. Hence, it suggests that our instantiation of CI could be used to derive initial insights into logs without analysts having to consider models with hundreds of activities. For more details, please refer to [13].

5.2 Case Study I: Digital Learning Environment Interactions

Course Insights is a learning analytics dashboard (LAD) that provides comparative visualisations of students interacting with a digital learning environment [18], by filtering on trace attributes such as residential status, gender, program or assessment grade. Drill-downs has been under-utilised: rarely more than one feature is filtered for [18]. We applied CI to a log from a calculus and linear algebra course offered to 736 undergraduate students at the University of Queensland, containing 18,883 events, 51 activities (per chapter: read material, submit quiz and review solutions), and 2,447 trace feature sets ($k = 4$). In collaboration with the instructor, who was not otherwise involved in this study, we

selected a cohort with a high distance: international students with a high exam grade (9% of the traces, 0.16 distance, IH), and compared it with the other students (¬IH). We found clear differences in their process: while IH alternates between the types of activities troughout the semester, ¬IH mainly performs quizzes sequentially at the end of the semester. To verify that these patterns are not also present in IH's super-cohorts of international students (I) or students with high exam grades (H) (to avoid a drill-down fallacy [10]), we repeated the analysis on I and H, and did not find these patterns.

The feedback of the instructor can be summarised as follows: (1) the recommendation of filters enables finding the differences between students' cohorts, while the current number of filtering choices in the LAD is too overwhelming to use; (2) the findings of learning behaviour that have led to successful outcome can be used for positive deviance [16] purposes. The instructor showed interest in sharing the findings about IH's learning process as a successful learning pattern with students to encourage early engagement; (3) getting notified (by the system) during the semester of any deviation of cohorts which might lead to learning failure could help with supporting students in-time. Thus, the integration of CI in the LAD has been considered [19]. This case study illustrates that certain insights might only be found using $k > 1$.

5.3 Case Study II: Research Student Journeys

The Queensland University of Technology utilises electronic forms (e-forms) to support higher degree research students with milestones. A log was extracted of 1,520 traces (students), 42,426 events, 15 activities (submission of forms, checks, approvals, . . .) and 4 trace attributes: faculty, scholarship, study mode (full/part time), and residency. Stakeholders' questions were whether processing was consistent across (q1) faculties and (q2) other student groups. To answer these questions, we applied CI with $k = 1$. The results showed that faculty B had the highest distance (0.19). In answer to q1, process models [11] of B and the other traces ($¬B$) are similar, indicating little difference in control flow. However, the likelihoods of choices and rework were different. Second, the results conveyed that there was minimal variation related to other demographic factors such as mode of study or domestic and international students: their distances were close to 0, which indicates that these features contribute no more to behaviour than randomly selected cohorts.

As a brief empirical evaluation, we presented key findings to stakeholders who were otherwise not involved in this study. The results were well-received: (1) insights regarding high variations across faculties were used to propose standardisation of e-forms processing, and (2) objective evidence was found that student demographic factors had no influence on the stochastic behaviour of processing e-forms. In summary, this case study demonstrates the utility of CI in a question-driven context in which its outputs were used to directly answer stakeholders' questions and can lead to actionable insights.

5.4 Discussion & Limitations

In Section 5.1, we showed that CI is feasible on real-life logs, however for higher k, its exponential nature kicks in. Further pruning steps could use mono-

tonicity, which however does not hold for our current instantiation with EMSC: a cohort's distance is not a bound for the distance of including additional features. Our instantiation could be extended to include smarter elicitation of numerical ranges using e.g. clustering, and to include more elaborate ways to combine several event-distance measures into δ_e without hitting the curse of dimensionality.

The two case studies showed that CI can lead to insights by itself, and that CI can assist explorative and question-driven process mining efforts as a first step before existing process comparison techniques are applied (for which the case studies provided examples). Furthermore, the case studies highlight that the recommended cohorts cannot be identified by existing techniques: trace clustering techniques (e.g. [17]) would not be able to provide clear-cut trace feature drill-down recommendations, and existing cohort identification techniques either require a process model and operate on event features only [2], or do not provide trace feature recommendations at all [14]. While there are currently no published techniques to recommend trace attributes for filtering based on behaviour, it would be interesting to compare CI with existing approaches of log variance reduction (using e.g. variance metrics), or commercial feature recommendations (using e.g. user studies), all of which we leave as future work.

6   Conclusion

Applying process mining techniques to event logs with high variance is challenging, which can be addressed by filtering sub-logs of traces (*cohorts*) defined by trace attribute value ranges (*features*), in order to compare behavioural differences between cohorts or to drill down into a particular cohort. To the best of our knowledge, no *cohort identification* techniques have been published, that is, techniques that recommend feature sets for logs, based on behavioural differences, which may include control flow and frequency of trace variants. In this paper, we proposed the Cohort Identification (CI) framework to automatically recommend feature sets that correspond to the largest differences in behaviour, where users can select what data or information constitutes behaviour. The framework was instantiated and implemented as a plug-in of the ProM framework.

Our evaluation found that CI can be applied in reasonable time to public real-life logs. Furthermore, we reported on two case studies in two Australian universities, showing that question-driven application led to addressing questions from stakeholders that could only be answered using stochastic-aware techniques. Stakeholders who were not otherwise involved in this work verified the findings.

Future extensions of CI could include clustering of numerical values in the feature elicitation step, and further heuristics to prune the feature sets to be considered. Second, it would be interesting to integrate CI in process mining methodologies such as [8] to assist in drill-down cycles. Third, stochastic log-log comparison techniques could be used to highlight differences in the stochastic behaviour of cohorts. Finally, we intend to perform elaborate user studies into the usefulness of the recommendations, similar to [17], in combination with an evaluation of, potentially new, log-log comparison techniques.

References

1. van der Aalst, W.M.P., et al.: Comparative process mining in education: An approach based on process cubes. In: SIMPDA. vol. 203, pp. 110–134 (2013)
2. Bolt, A., van der Aalst, W.M.P., de Leoni, M.: Finding process variants in event logs - (short paper). In: CoopIS. vol. 10573, pp. 45–52 (2017)
3. Bolt, A., de Leoni, M., van der Aalst, W.M.P.: Process variant comparison: Using event logs to detect differences in behavior and business rules. IS **74**, 53–66 (2018)
4. Bolt, A., et al.: Exploiting process cubes, analytic workflows and process mining for business process reporting: A case study in education. In: SIMPDA. vol. 1527, pp. 33–47 (2015)
5. Bolt, A., et al.: A visual approach to spot statistically-significant differences in event logs based on process metrics. In: CAiSE. vol. 9694, pp. 151–166 (2016)
6. Bose, R.P.J.C., van der Aalst, W.M.P.: Context aware trace clustering: Towards improving process mining results. In: SDM. pp. 401–412 (2009)
7. Buijs, J.C.A.M., Reijers, H.A.: Comparing business process variants using models and event logs. In: BPMDS. pp. 154–168 (2014)
8. van Eck, M.L., Lu, X., Leemans, S.J.J., van der Aalst, W.M.P.: PM^2: A process mining project methodology. In: CAiSE. pp. 297–313 (2015)
9. Joglekar, M., Garcia-Molina, H., Parameswaran, A.G.: Interactive data exploration with smart drill-down. TKDE **31**(1), 46–60 (2019)
10. Lee, D.J.L., et al.: Avoiding drill-down fallacies with *VisPilot*: assisted exploration of data subsets. In: IUI. pp. 186–196. ACM (2019)
11. Leemans, S.J.J., Poppe, E., Wynn, M.T.: Directly follows-based process mining: Exploration & a case study. In: ICPM. pp. 25–32 (2019)
12. Leemans, S.J.J., Syring, A.F., van der Aalst, W.M.P.: Earth movers' stochastic conformance checking. In: BPM forum. pp. 127–143 (2019)
13. Leemans, S.J.J., et al.: Results with identifying cohorts: Recommending drill-downs based on differences in behaviour for process mining. Tech. rep., Queensland University of Technology (2020)
14. de Leoni, M., et al.: A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs. Inf. Syst. **56**, 235–257 (2016)
15. Maaradji, A., Dumas, M., Rosa, M.L., Ostovar, A.: Detecting sudden and gradual drifts in business processes from execution traces. TKDE **29**(10), 2140–2154 (2017)
16. Marsh, D.R., Schroeder, D.G., Dearden, K.A., Sternin, J., Sternin, M.: The power of positive deviance. Bmj **329**(7475), 1177–1179 (2004)
17. Seeliger, A., Guinea, A.S., Nolle, T., Mühlhäuser, M.: Processexplorer: Intelligent process mining guidance. In: BPM. vol. 11675, pp. 216–231 (2019)
18. Shabaninejad, S., et al.: Automated insightful drill-down recommendations for learning analytics dashboards. In: LAK. p. 41–46 (2020)
19. Shabaninejad, S., et al.: Recommending insightful drill-downs based on learning processes for learning analytics dashboards. In: AIED. pp. 486–499 (2020)
20. Suriadi, S., et al.: Measuring patient flow variations: A cross-organisational process mining approach. In: AP-BPM. vol. 181, pp. 43–58 (2014)
21. Syamsiyah, A., et al.: Business process comparison: A methodology and case study. In: BIS. vol. 288, pp. 253–267 (2017)
22. Taymouri, F., Rosa, M.L., Dumas, M., Maggi, F.M.: Business process variant analysis: Survey and classification. CoRR **abs/1911.07582** (2019)
23. Weerdt, J.D., vanden Broucke, S.K.L.M., Vanthienen, J., Baesens, B.: Active trace clustering for improved process discovery. TKDE **25**(12), 2708–2720 (2013)