Abstract-and-Compare: A Family of Scalable Precision Measures for Automated Process Discovery

Adriano Augusto[1,2], Abel Armas-Cervantes[2], Raffaele Conforti[2], Marlon Dumas[1], Marcello La Rosa[2], and Daniel Reissner[2]

[1] University of Tartu, Estonia
{adriano.augusto, marlon.dumas}@ut.ee
[2] University of Melbourne, Australia
{raffaele.conforti, marcello.larosa, abel.armas}@unimelb.edu.au

Abstract. Automated process discovery techniques allow us to extract business process models from event logs. The quality of models discovered by these techniques can be assessed with respect to various criteria related to simplicity and accuracy. One of these criteria, namely *precision*, captures the extent to which the behavior allowed by a process model is observed in the log. While several measures of precision have been proposed, a recent study has shown that none of them fulfills a set of five axioms that capture intuitive properties behind the concept of precision. In addition, existing precision measures suffer from scalability issues when applied to models discovered from real-life event logs. This paper presents a family of precision measures based on the idea of comparing the k-th order Markovian abstraction of a process model against that of an event log. We demonstrate that this family of measures fulfils the aforementioned axioms for a suitably chosen value of k. We also empirically show that representative exemplars of this family of measures outperform a commonly used precision measure in terms of scalability and that they closely approximate two precision measures that have been proposed as possible ground truths.

1 Introduction

Contemporary enterprise information systems store detailed records of the execution of the business processes they support, such as records of the creation of process instances (a.k.a. *cases*), the start and completion of tasks, and other events associated with a case. These records can be extracted as event logs consisting of a set of traces, each trace itself consisting of a sequence of events associated with a case. Automated process discovery techniques [3] allow us to extract process models from such event logs. The quality of process models discovered in this way can be assessed with respect to several quality criteria related to simplicity and accuracy.

Two commonly used criteria for assessing accuracy are fitness and precision. *Fitness* captures the extent to which the behavior observed in an event log is allowed by the discovered process model (i.e. Can the process model generate every trace observed in the event log?). Reciprocally, *precision* captures the extent to which the behavior allowed by a discovered process model is observed in the event log. A low precision indicates that the model under-fits the log, i.e. it can generate traces that are unrelated

or only partially related to traces observed in the log, while a high precision indicates that it over-fits (i.e. it can only generate traces in the log and nothing more).[1]

While several precision measures have been proposed, a recent study has shown that none of them fulfils a set of five axioms that capture intuitive properties behind the concept of precision [15]. In addition, most of the existing precision measures suffer from scalability issues when applied to models discovered from real-life event logs.

This paper presents a family of precision measures based on the idea of comparing the k^{th}-order Markovian abstraction of a process model against that of an event log using a graph matching operator. We show that the proposed precision measures fulfil four of the aforementioned axioms for any k, and all five axioms for a suitable k dependent on the log. In other words, when measuring precision, we do not need to explore the entire state space of a process model but only its state space up to a certain memory horizon.

The paper empirically evaluates exemplars of the proposed family of measures using: (i) a synthetic collection of models and logs previously used to assess the suitability of precision measures, and (ii) a set of models discovered from 20 real-life event logs using three automated process discovery techniques. The synthetic evaluation shows that the exemplar measures closely approximate two precision measures that have been proposed as ground truths. The evaluation based on real-life logs shows that for values of up to $k = 5$, the kth-order Markovian precision measure is considerably more efficient than a commonly used precision measure, namely alignments-based ETC precision [1].

The rest of the paper is structured as follows. Section 2 introduces existing precision measures and the axioms defined in [15]. The family of Markovian precision measures is presented in Section 3 and evaluated in Section 4. Finally, Section 5 draws conclusions and directions for future work.

2 Background and Related Work

One of the earliest precision measures was proposed by Greco et al. [8], based on the *set difference* (SD) between the model behavior and the log behavior, each represented as a set of traces. This measure is a direct operationalization of the concept of precision, but it is not applicable to models with cycles since the latter have an infinite set of traces.

Later, Rozinat and van der Aalst [14] proposed the *advanced behavioral appropriateness* (ABA) precision. The ABA precision is based on the comparison between the sets of activity pairs that sometimes but not always follow each other, and the set of activity pairs that sometimes but not always precede each other. The comparison is performed on the sets extracted both from the model and the log behaviors. The ABA precision does not scale to large models and it is undefined for models with no routing behavior (i.e. models without concurrency or conflict relations) [15].

De Weerdt et al. [7] proposed the *negative events* precision measure (NE). This method works by inserting inexistent (so-called negative) events to enhance the traces in the log. A negative event is inserted after a given prefix of a trace if this event is never observed preceded by that prefix anywhere in the log. The traces extended with

[1] A third accuracy criterion in automated process discovery is *generalization*: the extent to which the process model captures behavior that, while not observed in the log, is implied by it.

negative events are then replayed on the model. If the model can parse some of the negative events, it means that the model has additional behavior. This approach is however heuristic: it does not guarantee that all additional behavior is identified.

Muñoz-Gama and Carmona [13] proposed the *escaping edges* (ETC) precision. Using the log behavior as reference, it builds a *prefix automaton* and, while replaying the process model behavior on top of it, counts the number of *escaping edges*, i.e. edges not in the prefix automaton which represent extra behavior of the process. Subsequently, to improve the robustness of the ETC precision for logs containing non-fitting traces, the ETC precision evolved into the *alignments-based ETC* precision (ETC_a) [1] where the replay is guided by alignments.

Despite its robustness, ETC_a does not scale well to real-life datasets. To address this issue, Leemans et al. [12] proposed the *projected conformance checking* (PCC) precision. This precision, starting from the log behavior and the model behavior builds a projected automaton (an automaton where a reduced number of activities are encoded) from each of them, i.e. A_l and A_m. These two automata are then used to generate a third automaton capturing their common behavior, i.e. $A_{l,m}$. The precision value is then computed as the ratio between the number of outgoing edges of each state in $A_{l,m}$ and the number of outgoing edges of the corresponding states occurring in A_m.

Finally, van Dongen et al. [16] proposed the *anti-alignment* precision (AA). This measure analyses the anti-alignments of the process model behavior to assess the model's precision. An anti-alignment of length n is a trace in the process model behavior of length at most equal to n, which maximizes the Levenshtein distance from all traces in the log.

In a recent study, Tax et al. [15] proposed five axioms to capture intuitive properties behind the concept of precision advising that any precision measure should fulfill these axioms. We start by introducing preliminary concepts and notations, and then proceed to present the five axioms.

Definition 1. ***[Trace]*** *Given a set of activity labels Σ, we define a trace on Σ as a sequence $\tau_\Sigma = \langle t_1, t_2, \ldots, t_{n-1}, t_n \rangle$, such that $\forall 1 \leq i \leq n, t_i \in \Sigma$.* [2] *Furthermore, we denote with τ_i the activity label in position i, and we use the symbol Γ_Σ to refer to the universe of traces on Σ. With abuse of notation, hereinafter we refer to any $t \in \Sigma$ as an activity instead of an activity label.*

Definition 2. ***[Subtrace]*** *Given a trace $\tau = \langle t_1, t_2, \ldots, t_{n-1}, t_n \rangle$, with the notation $\tau^{i \to j}$, we refer to the subtrace $\langle t_i, t_{i+1}, \ldots, t_{j-1}, t_j \rangle$, where $0 < i < j \leq n$. We extend the subset operator to traces, i.e., given two traces τ and $\widehat{\tau}$, $\widehat{\tau}$ is contained in τ, shorthanded as $\widehat{\tau} \subset \tau$, if and only if (iff) $\exists i, j \in \mathbb{N} \mid \tau^{i \to j} = \widehat{\tau}$.*

Definition 3. ***[Process Model Behavior]*** *Given a process model P (regardless of its representation) and being Σ the set of its activities. We refer to the model behavior as $\mathscr{B}_P \subseteq \Gamma_\Sigma$, where $\forall \langle t_1, t_2, \ldots, t_{n-1}, t_n \rangle \in \mathscr{B}_P$ there exists an execution of P that allows to execute the sequence of activities $\langle t_1, t_2, \ldots, t_{n-1}, t_n \rangle$, where t_1 is the first activity executed, and t_n the last.* [3]

Definition 4. ***[Event Log Behavior]*** *Given a set of activities Σ, an event log L is a finite multiset of traces defined over Σ. The event log behavior of L is defined as $\mathscr{B}_L = support(L)$.* [4]

[2] To enhance the readability, in the rest of this paper we refer to τ_Σ as τ, omitting the set Σ.

[3] In the case $\mathscr{B}_P = \Gamma_\Sigma$, P corresponds to the flower model.

[4] The support of a multiset is the set containing the distinct elements of the multiset.

Definition 5. ***[Precision Axioms]***

- ***Axiom-1.*** *A precision measure is a deterministic function* $prec : \mathscr{L} \times \mathscr{P} \rightarrow \mathbb{R}$*, where* $\mathscr{L}$ *is the universe of event logs, and* $\mathscr{P}$ *is the universe of processes.*
- ***Axiom-2.*** *Given two process models* P_1, P_2 *and a log L, if the behavior of L is contained in the behavior of* P_1*, and this latter is contained in the behavior of* P_2*, the precision value of* P_1 *must be equal to or greater than the precision value of* P_2*. Formally, if* $\mathscr{B}_L \subseteq \mathscr{B}_{P_1} \subseteq \mathscr{B}_{P_2} \Longrightarrow prec(L,P_1) \geq prec(L,P_2)$*.*
- ***Axiom-3.*** *Given two process models* P_1, P_2 *and a log L, if the behavior of L is contained in the behavior of* P_1*, and* P_2 *is the flower model, the precision value of* P_1 *must be greater than the precision value of* P_2*. Formally, if* $\mathscr{B}_L \subseteq \mathscr{B}_{P_1} \subset \mathscr{B}_{P_2} = \Gamma_\Sigma \Longrightarrow prec(L,P_1) > prec(L,P_2)$*.*
- ***Axiom-4.*** *Given two process models* P_1, P_2 *and a log L, if the behavior of* P_1 *is equal to the behavior of* P_2*, the precision values of* P_1 *and* P_2 *must be equal. Formally, if* $\mathscr{B}_{P_1} = \mathscr{B}_{P_2} \Longrightarrow prec(L,P_1) = prec(L,P_2)$*.*
- ***Axiom-5.*** *Given a process model P and two event logs* L_1, L_2*, if the behavior of* L_1 *is contained in the behavior of* L_2*, and the behavior of* L_2 *is contained in the behavior of P, the precision value of the model measured over* L_2 *must be equal to or greater than the precision value measured over* L_1*. Formally, if* $\mathscr{B}_{L_1} \subseteq \mathscr{B}_{L_2} \subseteq \mathscr{B}_P \Longrightarrow prec(L_2,P) \geq prec(L_1,P)$*.*

Tax et al. [15] showed that none of the existing measures fulfils all the axioms.

3 Markovian Abstraction-based Precision (MAP)

This section presents a family of precision measures based on k$^{\text{th}}$-order Markovian abstractions. Intuitively, precision measures try to estimate how much of the behavior captured in a process model can be found in the behavior recorded in an event log. The computation of our precision measures can be divided into three steps: i) abstraction of the behavior of a process model, ii) abstraction of the behavior recorded in an event log, and iii) comparison of the two behavioral abstractions. We start by defining the k$^{\text{th}}$-order Markovian abstraction, as well as its features, and then introduce the algorithm to compare a pair of Markovian abstractions. Finally, we show that our precision measures satisfy four of the five precision axioms, while the fifth axiom is also satisfied for specific values of k.

3.1 Markovian Abstraction

A k$^{\text{th}}$-order Markovian abstraction (M$^{\text{k}}$-abstraction) is a graph composed by a set of states (S) and a set of edges ($E \subseteq S \times S$). In an M$^{\text{k}}$-abstraction, every state $s \in S$ represents a (sub)trace of at most length k, e.g. $s = \langle b,c,d \rangle$, while two states $s_1, s_2 \in S$ are connected via an edge $e = (s_1, s_2) \in E$ iff s_1 and s_2 satisfy the following three properties: i) the first activity of the (sub)trace represented by s_1 can occur before the (sub)trace represented by s_2, ii) the last activity of the (sub)trace represented by s_2 can occur after the (sub)trace represented by s_1, and iii) the two (sub)traces represented by s_1 and s_2 overlap with the exception of their first and last activity, respectively, e.g. $e = (\langle b,c,d \rangle, \langle c,d,e \rangle)$. Every state of an M$^{\text{k}}$-abstraction is unique, i.e. there are no two states representing the same (sub)trace. An M$^{\text{k}}$-abstraction is defined w.r.t. a given order

k, which defines the size of the (sub)traces encoded in the states. An M$^{\mathrm{k}}$-abstraction contains a fresh state (denoted as $-$) representing the sink and source of the M$^{\mathrm{k}}$-abstraction. Intuitively, every state represents either a trace of length less than or equal to k or a subtrace of length k, whilst every edge represents an existing subtrace of length $k+1$ or a trace of length less than or equal to $k+1$. Thus, M$^{\mathrm{k}}$-abstraction captures how all the traces of the input behavior evolves in chunks of length k. The definitions below show the construction of a M$^{\mathrm{k}}$-abstraction from a given $\mathscr{B}_X$, and a fundamental property of the M$^{\mathrm{k}}$-abstractions to show that our precision measure fulfils the 5 precision axioms.

Definition 6. ***[k^{th}-order Markovian Abstraction]*** *Given a set of traces $\mathscr{B}_X$, the k-order Markovian Abstraction is the graph $M^k_X = (S,E)$ where S is the set of the states and $E \subseteq S \times S$ is the set of edges, such that*

- $S = \{-\} \cup \{\tau \,:\, \tau \in \mathscr{B}_X \wedge |\tau| \leq k\} \cup \{\tau^{i \to j} \,:\, \tau \in \mathscr{B}_X \wedge |\tau| > k \wedge \left|\tau^{i \to j}\right| = k\}$
- $E = \{(-,\tau) \,:\, \tau \in S \wedge |\tau| \leq k\} \cup \{(\tau,-) \,:\, \tau \in S \wedge |\tau| \leq k\} \cup$
  $\{(-,\tau) \,:\, \exists \widehat{\tau} \in \mathscr{B}_X \text{ s.t. } \tau = \widehat{\tau}^{1 \to k}\} \cup \{(\tau,-) : \exists \widehat{\tau} \in \mathscr{B}_X \text{ s.t. } \tau = \widehat{\tau}^{(|\widehat{\tau}|-k+1) \to |\widehat{\tau}|}\} \cup$
  $\{(\tau',\tau'') \,:\, \tau',\tau'' \in S \wedge \tau' \oplus \tau''_{|\tau''|} = \tau'_1 \oplus \tau'' \wedge \exists \widehat{\tau} \in \mathscr{B}_X \text{ s.t. } \tau'_1 \oplus \tau'' \subseteq \widehat{\tau}\}$ [5]

Theorem 1. ***[Equality and Containment Inheritance]*** *Given two sets of traces $\mathscr{B}_X$ and $\mathscr{B}_Y$, and their respective M^k-abstractions $M^k_X = (S_X,E_X)$ and $M^k_Y = (S_Y,E_Y)$, any equality or containment relation between $\mathscr{B}_X$ and $\mathscr{B}_Y$ is inherited by E_X and E_Y. I.e., if $\mathscr{B}_X = \mathscr{B}_Y$ then $E_X = E_Y$, or if $\mathscr{B}_X \subset \mathscr{B}_Y$ then $E_X \subseteq E_Y$.*

Proof. (Sketch) This follows by construction. Specifically, every edge $e \in E_X$ represents either a subtrace $\tau^{x \to y} : \tau \in \mathscr{B}_X \wedge |\tau^{x \to y}| = k+1$, or it represents a trace $\tau : \tau \in \mathscr{B}_X \wedge |\tau| < k+1$. The last implies that from the same sets of traces the corresponding M$^{\mathrm{k}}$-abstractions contain the same sets of edges. □

Note, however, that the theorem above cannot say anything for the traces in $\mathscr{B}_Y \setminus \mathscr{B}_X$, i.e. adding new traces to $\mathscr{B}_X$ does not imply that new edges are added to E_X. As a result the relation $\mathscr{B}_X \subset \mathscr{B}_Y$ guarantees only $E_X \subseteq E_Y$, instead of $E_X \subset E_Y$.

Note that, M^1-abstraction is equivalent to a *directly-follows graph* (a well-known behavior abstraction used as starting point by many process discovery approaches [10, 5, 17, 18]). Instead, if k approaches to infinite then M$^\infty$-abstraction is equivalent to listing all the traces. The M$^{\mathrm{k}}$-abstraction of a process model can be built from its reachability graph by replaying it. The time complexity of such operation strictly depends on k, and it ranges from polynomial time ($k = 1$) to double exponential time for greater values of k. Instead, the M$^{\mathrm{k}}$-abstraction of an event log can be built always in polynomial time, since the log behavior is a finite set of traces.

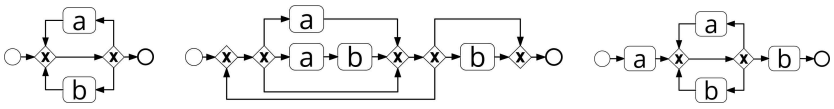
Traces
$\langle a,a,b \rangle$
$\langle a,b,b \rangle$
$\langle a,b,a,b,a,b \rangle$

Table 1: Log L^*.

One can tune the level of behavioral approximation by varying the order k of the M$^{\mathrm{k}}$-abstraction. For example, let us consider the event log L^* as in Tab. 1, and the Process-X (P_x) in Fig. 1c. Their respective M^1-abstractions: $M^1_{L^*}$ and $M^1_{P_x}$ are shown in Fig. 2d and 2c. We can notice that $M^1_{L^*} = M^1_{P_x}$, though $\mathscr{B}_{P_x}$ is infinite whilst $\mathscr{B}_{L^*}$ is not. This is an example on how the M^1-abstraction can over-approximate the behavior it represents. However, the increase of k can lead to

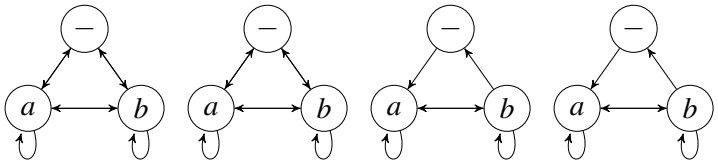
[5] The operator $\oplus$ is the *concatenation* operator.

more accurate representations (decreasing the degree of over-approximation) and thus to behavioral differences between behaviorally-similar abstractions, e.g., L^* and P_x, can be detected, see Fig. 3d and 3c. We remark that for k equal to the length of the longest trace in the log, the behavioral abstraction of this latter is exact. However, a similar reasoning cannot be done for the model behavior, since its longest trace may be infinite.



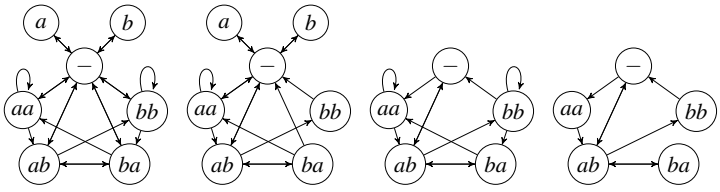
(a) Flower Proc. (b) Process Y (c) Process X

Fig. 1: Examples of processes in the BPMN language.



(a) $1-\frac{2}{8}=0.75$ (b) $1-\frac{2}{8}=0.75$ (c) $1-\frac{0}{6}=1.00$ (d) -

Fig. 2: From left to right: the M^1-abstraction of the Flower Process, Process-Y, Process-X and the event log L^*. The respective labels report the value of their MAP^1.



(a) $1-\frac{12}{20}=0.40$ (b) $1-\frac{8}{16}=0.50$ (c) $1-\frac{4}{12}=0.66$ (d) -

Fig. 3: From left to right, the M^2-abstraction of the Flower Process, Process-Y, Process-X and the event log L^*. The respective labels report the value of their MAP^2.

3.2 Comparing Markovian Abstractions

The third and final step of our precision measure is the comparison of the M^K-abstractions of the process model and the event log. In short, given two M^k-abstractions, we compare them using a weighted edge-based graph matching algorithm.

Definition 7. ***[Weighted Edge-based Graph Matching Algorithm (GMA)]*** *A Weighted Edge-based Graph Matching Algorithm (GMA) is an algorithm that receives as input two graphs $G_1 = (N_1, E_1)$ and $G_2 = (N_2, E_2)$, and outputs a mapping function $\mathscr{I}_C : E_1 \rightarrow (E_2 \cup \{\varepsilon\})$. The*

function $\mathscr{I}_C$ maps pairs of edges matched by a graph matching algorithm or, if no mapping was found, the edges in E_1 are mapped to ε, i.e., $\forall e_1,e_2 \in E_1 : \mathscr{I}_C(e_1) = \mathscr{I}_C(e_2) \Rightarrow (e_1 = e_2) \vee (\mathscr{I}_C(e_1) = \varepsilon \wedge \mathscr{I}_C(e_2) = \varepsilon)$. A GMA is characterised by an underlying cost function $C : E_1 \times (E_2 \cup \{\varepsilon\}) \rightarrow [0,1]$, s.t. $\forall e_1 \in E_1$ and $\forall e_2 \in E_2 \Longrightarrow C(e_1,e_2) \in [0,1]$ and $\forall e_1 \in E_1 \Longrightarrow C(e_1,\varepsilon) = 1$. Hereinafter we refer to any GMA as its mapping function $\mathscr{I}_C$.

Given a GMA $\mathscr{I}_C$, an event log L and a process P as inputs, the k[th]-order Markovian abstraction-based precision (hereby MAP^k) is estimated applying Equation 1.

$$MAP^k(L,P) = 1 - \frac{\Sigma_{e \in E_P} C(e, \mathscr{I}_C(e))}{|E_P|} \tag{1}$$

The selected GMA for the implementation of our MAP^k is an adaptation of the Hungarian method [9], where: the cost of a match between two edges is defined as the average of the Levenshtein distance between the source states and the target states; and the final matching is the one minimising the total costs of the matches.

Figure 1 shows three models in BPMN notation. Their respective Markovian abstractions are captured in Figs. 2a-2c and 3a-3c, for $k = 1$ and $k = 2$. We can observe that by increasing k, the quality of the behavior approximation decreases. Consequently, the MAP^k achieves a finer result.

Note that each of the proposed precision measures fulfills the properties of an ordinal scale. Specifically, given an event log $\mathscr{L}$ and for a given k, MAP^k induces an order over the possible process models that fit log $\mathscr{L}$. This property is desirable given that the purpose of a precision measure is to allow us to compare two possible process models in terms of their additional behavior.

3.3 Proofs of the 5-Axioms

We now turn our attention to show that our Markovian abrastraction-based precision measure fulfils the axioms presented in Section 2. For the remaining part of the section, let L_x be a log, P_x be a process model, and $M^k_{L_x} = (S_{L_x}, E_{L_x})$ and $M^k_{P_x} = (S_{P_x}, E_{P_x})$ be the M[k]-abstractions of the log and the model, respectively.

Axiom-1. $MAP^k(L,P)$ is a deterministic function. Given a log L and a process P, The construction of M^k_L and M^k_P is fully deterministic for $\mathscr{B}_P$ and $\mathscr{B}_L$ (see Definition 6). Furthermore, being the graph matching algorithm $\mathscr{I}_C$ deterministic, and being $MAP^k(L,P)$ function of E_L, E_P and $\mathscr{I}_C$ (see Equation 1), it follows that $MAP^k(L,P)$ is also deterministic with codomain $\mathbb{R}$.

Axiom-2. Given two processes P_1, P_2 and an event log L, s.t. $\mathscr{B}_L \subseteq \mathscr{B}_{P_1} \subseteq \mathscr{B}_{P_2}$, then $MAP^k(L,P_1) \geq MAP^k(L,P_2)$. First, the following relation holds, $E_L \subseteq E_{P_1} \subseteq E_{P_2}$ (see Theorem 1). Then, we distinguish two possible cases:

1. if $E_{P_1} = E_{P_2}$, then it follows straightforward $MAP^k(L,P_1) = MAP^k(L,P_2)$, because $MAP^k(L,P)$ is a deterministic function of E_L, E_P and $\mathscr{I}_C$ (see Axiom-1 proof and Equation 1).
2. if $E_{P_1} \subset E_{P_2}$, then $E_L \subset E_{P_2} \wedge \ (|E_{P_2}| - |E_{P_1}|) > 0$. In this case, we show that $MAP^k(L,P_2) - MAP^k(L,P_1) < 0$ is always true, as follows.

$$1 - \frac{\Sigma_{e_2 \in E_{P_2}} C(e_2, \mathscr{I}_C(e_2))}{|E_{P_2}|} - \left(1 - \frac{\Sigma_{e_1 \in E_{P_1}} C(e_1, \mathscr{I}_C(e_1))}{|E_{P_1}|}\right) = \frac{\Sigma_{e_1 \in E_{P_1}} C(e_1, \mathscr{I}_C(e_1))}{|E_{P_1}|} - \frac{\Sigma_{e_2 \in E_{P_2}} C(e_2, \mathscr{I}_C(e_2))}{|E_{P_2}|} < 0$$

For each edge e_1 that can be found both in E_{P_1} and E_L, the cost $C(e_1, \mathscr{I}_C(e_1))$ is 0, being $\mathscr{I}_C(e_1) = e_1$. Instead, for each edge e_1 that can be found in E_{P_1} but not in E_L, the cost $C(e_1, \mathscr{I}_C(e_1))$ is 1, being $\mathscr{I}_C(e_1) = \varepsilon$. It follows that the total cost of matching E_{P_1} over $_L$ is $\Sigma_{e_1 \in E_{P_1}} C(e_1, \mathscr{I}_C(e_1)) = |E_{P_1}| - |E_L|$. A similar reasoning can be done for the matching of E_{P_2} over $_L$. Indeed, $\forall e_2 \in E_{P_2} \cap E_L \Longrightarrow C(e_2, \mathscr{I}_C(e_2)) = 0$ and $\forall e_2 \in E_{P_2} \setminus E_L \Longrightarrow C(e_2, \mathscr{I}_C(e_2)) = C(e_2, \varepsilon) = 1$, therefore $\Sigma_{e_2 \in E_{P_2}} C(e_2, \mathscr{I}_C(e_2)) = |E_{P_2}| - |E_L|$.
Applying these results to the above inequality, it turns into the following:

$$\frac{|E_{P_1}| - |E_L|}{|E_{P_1}|} - \frac{|E_{P_2}| - |E_L|}{|E_{P_2}|} = \frac{|E_L|\,(|E_{P_1}| - |E_{P_2}|)}{|E_{P_1}|\,|E_{P_2}|} < 0$$

This latter is always true, since the starting hypothesis of this second case is $(|E_{P_1}| - |E_{P_2}|) < 0$.

Axiom-3. Given two processes P_1, P_2 and an event log L, s.t. $\mathscr{B}_L \subseteq \mathscr{B}_{P_1} \subset \mathscr{B}_{P_2} = \Gamma_\Sigma$ then $MAP^k(L,P_1) > MAP^k(L,P_2)$. For any $k \in \mathbb{N}$, the relation $MAP^k(L,P_1) \geq MAP^k(L,P_2)$ holds for Axiom-2. The case $MAP^k(L,P_1) = MAP^k(L,P_2)$ occurs when $M^k_{P_2}$ over-approximates the behavior of P_2, i.e. $\mathscr{B}_{P_1} \subset \mathscr{B}_{P_2}$ and $E_{P_1} = E_{P_2}$. Nevertheless, for any $\mathscr{B}_{P_1}$ there always exists a k^* s.t. $E_{P_1} \subset E_{P_2}$. This is true since being $\mathscr{B}_{P_1}$ strictly contained in $\mathscr{B}_{P_2}$, there exists a trace $\widehat{\tau} \in \mathscr{B}_{P_2}$ s.t. $\widehat{\tau} \notin \mathscr{B}_{P_1}$. Choosing $k^* = |\widehat{\tau}|$, the $M^{k^*}_{P_2}$ would produce an edge $\widehat{e} = (-, \widehat{\tau}) \in E_{P_2}$ s.t. $\widehat{e} \notin E_{P_1}$ because $\widehat{\tau} \notin \mathscr{B}_{P_1}$ (see also Definition 6).[6] Consequently, for any $k \geq k^*$, we have $E_{P_1} \subset E_{P_2}$ and $MAP^k(L,P_1) > MAP^k(L,P_2)$ holds, being this latter the case 2 of Axiom-2.

Axiom-4. Given two processes P_1, P_2 and an event log L, s.t. $\mathscr{B}_{P_1} = \mathscr{B}_{P_2}$ then $MAP^k(L,P_1) = MAP^k(L,P_2)$. If $\mathscr{B}_{P_1} = \mathscr{B}_{P_2}$, then $E_{P_1} = E_{P_2}$ (see Theorem 1). It follows straightforward that $MAP^k(L,P_1) = MAP^k(L,P_2)$ (see proof Axiom-1 and Equation 1).

Axiom-5. Given two event logs L_1, L_2 and a process P, s.t. $\mathscr{B}_{L_1} \subseteq \mathscr{B}_{L_2} \subseteq \mathscr{B}_P$, then $MAP^k(L_2,P) \geq MAP^k(L_1,P)$. Consider the two following cases:

1. if $\mathscr{B}_{L_1} = \mathscr{B}_{L_2}$, then $E_{L_1} = E_{L_2}$ (see Theorem 1). It follows $MAP^k(L_2,P) = MAP^k(L_1,P)$, because $MAP^k(L,P)$ is a deterministic function of E_L, E_P and $\mathscr{I}_C$ (see Axiom-1 proof and Equation 1).
2. if $\mathscr{B}_{L_1} \subset \mathscr{B}_{L_2}$, then $E_{L_1} \subseteq E_{L_2}$ (see Theorem 1). In this case, the graph matching algorithm would find matchings for either the same number or a larger number of edges between M^k_P and $M^k_{L_2}$, than between M^k_P and $M^k_{L_1}$ (this follows from $E_{L_1} \subseteq E_{L_2}$). Thus, a smaller or equal number of edges will be mapped to

[6] Formally, $\exists \widehat{\tau} \in \mathscr{B}_{P_2} \setminus \mathscr{B}_{P_1}$, s.t. for $k^* = |\widehat{\tau}| \Longrightarrow \exists (-, \widehat{\tau}) \in E_{P_2} \setminus E_{P_1}$.

ε in the case of $MAP^k(L_2,P)$ not decreasing the value for the precision, i.e., $MAP^k(L_2,P) \geq MAP^k(L_1,P)$.

In Axiom-3 we showed that there exists a specific value of k, namely k^*, for which $MAP^{k^*}(L_x,P_x)$ satisfies Axiom-3 and we identified such value being $k^* = |\widehat{\tau}|$, where $\widehat{\tau}$ can be any trace of the set difference $\Gamma_\Sigma \setminus \mathscr{B}_{P_x}$. In the following, we show how to identify the minimum value of k^* such that all the 5-Axioms are satified. To identify the lowest value of k^*, we have to consider the traces $\widehat{\tau} \in \Gamma_\Sigma$ such that does not exists a $\tau \in \mathscr{B}_{P_x}$ where $\widehat{\tau} \subseteq \tau$. If a trace $\widehat{\tau} \in \Gamma_\Sigma$ that is not a sub-trace of any other trace of the process model behavior ($\mathscr{B}_{P_x}$) is found, by setting $k^* = |\widehat{\tau}|$ would mean that in the M^{k^*}-abstraction of Γ_Σ there will be a state $\widehat{s} = \widehat{\tau}$ and an edge $(-,\widehat{\tau})$ that are not captured by the M^{k^*}-abstraction of $\mathscr{B}_{P_x}$. This difference will allow us to distinguish the process P_x from the flower model (i.e. the model having a behavior equal to Γ_Σ), satisfying in this way the Axiom-3. At this point, considering the set of the lengths of all the subtraces not contained in any trace of $\mathscr{B}_{P_x}$, $Z = \{|\widehat{\tau}| \ : \ \widehat{\tau} \in \Gamma_\Sigma \ \wedge \ \nexists\, \tau \in \mathscr{B}_{P_x} \mid \widehat{\tau} \subseteq \tau\}$, we can set the lower-bound of $k^* \geq min(Z)$.

Note that the value of k^* is equal to 2 for any process model with at least one activity that cannot be executed twice in a row. If we have an activity $\widehat{t}$ that cannot be executed twice in a row, it means that $\left|\langle \widehat{t},\widehat{t} \rangle\right| \in Z$ and thus we can set $k^* = 2$. In practice, $k^* = 2$ satisfies all the 5-Axioms in real-life cases, since it is very common to find process models that have the above topological characteristic.

4 Evaluation

In this section, we report on a two-pronged evaluation we performed to assess the following two objectives: i) comparing our family of precision measures to state-of-the-art precision measures; and ii) analysing the role of the parameter k.

To do so, we implemented the Markovian Abstraction-based Precision (MAP^k) as a standalone open-source tool[7] and used it to carry out a qualitative evaluation on synthetic data and a quantitative evaluation on real-life data.[8] All experiments were executed on an Intel Core i5-6200U @2.30GHz with 16GB RAM running Windows 10 Pro (64-bit) and JVM 8 with 12GB RAM (8GB Stack and 4GB Heap).

4.1 Qualitative evaluation

In a previous study, van Dongen et al. [16] showed that their anti-alignment precision was able to improve on a range of state-of-the-art precision measures. To qualitatively assess our MAP^k, we decided to repeat the experiment carried out in [16] using the same synthetic dataset. Table 2 and Figure 4 show the synthetic event log and a model, called "original model", that was used to generate eight variants: a *single trace* model capturing the most frequent trace; a model incorporating all *separate traces*; a *flower model* of all activities in the log; a model with activities G and H in parallel (*Opt. G* ||

[7] Available at `http://apromore.org/platform/tools`

[8] The public data used in the experiments can be found at `https://doi.org/10.6084/m9.figshare.6376592.v1`

Opt. H, see Fig. 5); one with G and H in self-loop (↻*G,* ↻*H*, Fig. 6); a model with D in self-loop (↻*D*, Fig. 7); a model with all activities in parallel (*All parallel*); and a model where all activities are in round robin (*Round robin*, Fig. 8). Using each log-model

Traces	**#**
$\langle A,B,D,E,I\rangle$	1207
$\langle A,C,D,G,H,F,I\rangle$	145
$\langle A,C,G,D,H,F,I\rangle$	56
$\langle A,C,H,D,F,I\rangle$	23
$\langle A,C,D,H,F,I\rangle$	28

Table 2: Test log [16].

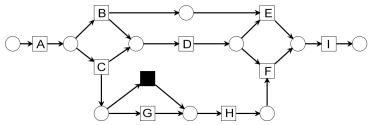


Fig. 4: Original model [16].

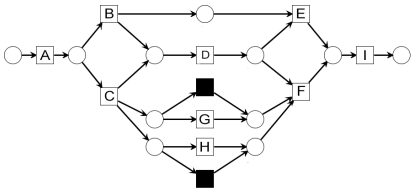


Fig. 5: Opt. G || Opt. H model [16].

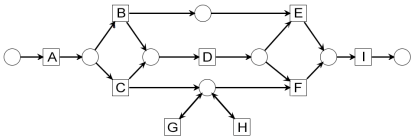


Fig. 6: ↻G, ↻H model [16].

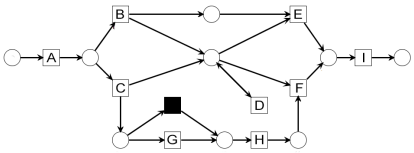


Fig. 7: ↻D model [16].

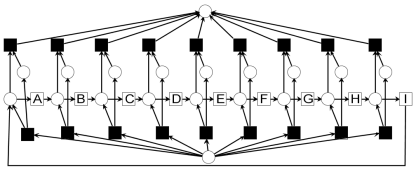


Fig. 8: Round robin model [16].

pair, we compared our precision measure MAP^k to the precision measures discussed in Section 2 (these include those evaluated by van Dongen et al. [16]), namely: traces set difference precision (SD), alignment-based ETC precision (ETC_a), negative events precision (NE), projected conformance checking (PCC), anti-alignment precision (AA). We left out the advanced behavioral appropriateness (ABA) as it is not defined for some of the models in this dataset. We limited the order k to 7, because it is the length of the longest trace in the log. Setting an order greater than 7 would only (further) penalise the cyclic behavior of the models, which is not necessary to assess the models' precision.

Table 3 reports the results of our qualitative evaluation.[9] To discuss these results, we use two precision measures as a reference, as these have been advocated as possible ground truths of precision, though none of them satisfies the axioms in [15]. The first

[9] Some values differ from those in [16] as we used each measure's latest implementation.

one is AA. This measure has been shown [16] to be intuitively more accurate than other precision measures. The second one is SD, as it closely operationalizes the definition of precision by capturing the exact percentage of model behavior that cannot be found in the log. As discussed in Section 2 though, this measure can only be computed for acyclic models, and uses a value of zero for cyclic models by design.

Process Variant	*Model Traces (#)*	*SD*	ETC_a	*NE*	*PCC*	*AA*	MAP^1	MAP^2	MAP^3	MAP^4	MAP^5	MAP^6	MAP^7
Original model	6	0.833	0.900	0.995	1.000	0.871	1.000	0.909	0.880	0.852	0.852	0.852	0.852
Single trace	1	1.000	1.000	0.893	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Separate traces	5	1.000	1.000	0.985	0.978	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Flower model	986,410	0.000	0.153	0.117	0.509	0.000	0.189	0.024	0.003	0.000	0.000	0.000	0.000
Opt. G || Opt. H	12	0.417	0.682	0.950	0.974	0.800	0.895	0.645	0.564	0.535	0.535	0.535	0.535
↻G, ↻H	362	0.000	0.719	0.874	0.896	0.588	0.810	0.408	0.185	0.080	0.034	0.015	0.006
↻D	118	0.000	0.738	0.720	0.915	0.523	0.895	0.556	0.349	0.223	0.145	0.098	0.069
All parallel	362,880	0.000	0.289	0.158	0.591	0.033	0.210	0.034	0.006	0.001	0.000^*	0.000^*	0.000^*
Round robin	27	0.000	0.579	0.194	0.594	0.000	0.815	0.611	0.496	0.412	0.350	0.306	0.274

Table 3: Comparison of different precision measures over synthetic dataset (* indicates a rounded-down value: $0.000^* > 0.000$).

From the results in Table 3, we can observe that MAP^1 does not penalise enough the extra behavior of some models, such as the *original* model, which cannot be distinguished from the *single trace* and the *separate traces* models (all have a precision of 1). Also, the values of MAP^1 are far away from those of both AA and SD (with the exception of the simplest models, i.e. *single trace* and *separate traces*). As we increase k, MAP^k tends to get closer to AA and to SD, barring a few exceptions. In particular, the more is the cyclic behavior allowed by a model, the quicker MAP^k tends to zero. In this respect, let us consider the cyclic models in our datasets: i) the *flower* model, ii) the $\circlearrowleft G$, $\circlearrowleft H$ model (Fig. 6), iii) the $\circlearrowleft D$ model (Fig. 7), and iv) the *round robin* (Fig. 8). The value of our precision measure tends to zero faster in the *flower* model (k=3) than in the other cyclic models, because the *flower* model allows the greatest amount of cyclic behavior, due to all the possible combinations of activities being permitted. At $k = 7$ this is consistent with both SD and AA. Similarly, our measure tends to zero slower in the *round robin* model because this model is very strict on the order in which activities can be executed, despite having infinite behavior. In fact, it only allows the sequence $\langle A,B,C,D,F,G,H,I\rangle$ to be executed, with the starting activity and the number of repetitions being variable. This is taken into account by our measure, since even with $k = 7$ we do not reach a value of zero for this model, as opposed to SD and AA. This allows us to discriminate the *round robin* model from other models with very large behavior such as the *flower* model. This is not possible with SD and AA, because both models have a precision of zero in these two measures. As for the other two cyclic models in our dataset, MAP^k tends to zero with speeds between those of the *flower* model and the *round robin* model, with the $\circlearrowleft G$, $\circlearrowleft H$ model being faster to drop than the $\circlearrowleft D$, due to the former allowing more cyclic behavior than the latter. Similar considerations as above apply to these two models: even at $k = 7$ their precision does not reach zero, which allows us to distinguish these models from other models such as the *all parallel* model, which has a very large behavior (360K+ distinct traces). While in SD the precision of these two models is set to zero by design, for AA these two models have a precision greater than zero, though the $\circlearrowleft G$, $\circlearrowleft H$ model has a higher precision than the $\circlearrowleft D$ model

(0.588 vs. 0.523). This is counter-intuitive, since the former model allows more model behavior not permitted by the log (in terms of number of different traces) than the latter model does. In addition, AA penalizes more the *round robin* model, despite this has less model behavior than the two models with self-loop activities. Altogether, these results show that the higher the k, the more the behavioral differences that our measure can catch and penalise.

In terms of ranking (see Table 4), our measure is the most consistent with the ranking of the models yielded by both SD (for acyclic models) and AA (for all models), than all other measures. As discussed above, the only differences with AA are in the swapping of the order of the two models with self loops, and in the order of the *round robin* model. Note that given that both the round robin and the flower model have a value of zero in AA, the next model in the ranking (*all parallel*) is assigned a rank of 3 instead of 2 in MAP^k. This is just the way the ranking is computed and is not really indicative of a ranking inconsistency between the two measures. Another observation is that the ranking yielded by our family of metrics remains the same for $k > 1$. This indicates that as we increase k, while the extent of behavioral differences we can identify and penalize increases, this is not achieved at the price of changing the ranking of the models.

Process Variant	*SD*	ETC_a	*NE*	*PCC*	*AA*	MAP^1	MAP^2	MAP^3	MAP^4	MAP^5	MAP^6	MAP^7
Original model	7	7	9	8	7	7	7	7	7	7	7	7
Single trace	8	8	6	8	8	7	8	8	8	8	8	8
Separate traces	8	8	8	7	8	7	8	8	8	8	8	8
Flower model	1	1	1	1	1	1	1	1	1	1	1	1
Opt. G || Opt. H	6	3	7	6	6	5	6	6	6	6	6	6
↻G, ↻H	1	5	5	4	5	3	3	3	3	3	3	3
↻D	1	6	4	5	4	5	4	4	4	4	4	4
All parallel	1	2	2	2	3	2	2	2	2	2	2	2
Round robin	1	4	3	3	1	4	5	5	5	5	5	5

Table 4: Models ranking yielded by the precision measures over the synthetic dataset.

On average it took less than a second per model to compute MAP^k, except for the *all parallel* model, for which it took 3.8 seconds at $k = 7$, due to the large number of distinct traces yielded by this model.

4.2 Quantitative evaluation

In our second evaluation, we used two datasets for a total of 20 logs. The first dataset is the collection of real-life logs publicly available from the 4TU Centre for Research Data, as of March 2017.[10] Out of this collection, we retained twelve logs related to business processes, as opposed to e.g. software development processes. These include the *BPI Challenge* (BPIC) logs (2012-17), the *Road Traffic Fines Management Process* (RTFMP) log, and the *SEPSIS* log. These logs record executions of business processes from a variety of domains, e.g. healthcare, finance, government and IT service management. In seven logs (BPIC14, the BPIC15 collection, and BPIC17), we applied the filtering technique proposed in [6] to remove infrequent behavior. The second dataset is composed of eight proprietary logs sourced from several companies in the education,

[10] `https://data.4tu.nl/repository/collection:event_logs_real`

insurance, IT service management and IP management domains. Table 5 reports the characteristics of both datasets, highlighting the heterogeneous nature of the data.

Log		BPIC12	$BPIC13_{cp}$	$BPIC13_{inc}$	$BPIC14_{f}$	$BPIC15_{1f}$	$BPIC15_{2f}$	$BPIC15_{3f}$	$BPIC15_{4f}$	$BPIC15_{5f}$
Total Traces		13,087	1,487	7,554	41,353	902	681	1,369	860	975
Dist. Traces		33.4	12.3	20	36.1	32.7	61.7	60.3	52.4	45.7
Total Events		262,200	6,660	65,533	369,485	21,656	24,678	43,786	29,403	30,030
Dist. Events		36	7	13	9	70	82	62	65	74
Tr. length	**(min)**	3	1	1	3	5	4	4	5	4
	(avg)	20	4	9	9	24	36	32	34	31
	(max)	175	35	123	167	50	63	54	54	61

Log		$BPIC17_{f}$	RTFMP	SEPSIS	PRT1	PRT2	PRT3	PRT4	PRT6	PRT7	PRT9	PRT10
Total Traces		21,861	150,370	1,050	12,720	1,182	1,600	20,000	744	2,000	787,657	43,514
Dist. Traces		40.1	0.2	80.6	8.1	97.5	19.9	29.7	22.4	6.4	0.01	0.01
Total Events		714,198	561,470	15,214	75,353	46,282	13,720	166,282	6,011	16,353	1,808,706	78,864
Dist. Events		41	11	16	9	9	15	11	9	13	8	19
Tr. length	**(min)**	11	2	3	2	12	6	6	7	8	1	1
	(avg)	33	4	14	5	39	8	8	8	8	2	1
	(max)	113	2	185	64	276	9	36	21	11	58	15

Table 5: Descriptive statistics of the real-life logs (public and proprietary).

First, we discovered different process models from each log, using three state-of-the-art automated process discovery methods [3]: Split Miner [4] (SM), Inductive Miner [11] (IM), and Structured Heuristics Miner [2] (SHM). Then, we measured the precision for each model with our MAP^k measure, by varying the order k in the range 2–5. Unfortunately, we were not able to use any of the previous reference measures, because SD does not work for cyclic models (all models discovered by IM were cyclic) and AA does not scale to real-life models [16]. Thus, we resorted to ETC_a as a baseline, since this is, to date, the most-scalable and widely-accepted precision measure for automated process discovery in real-life settings [3].

Table 6 shows the results of the quantitative evaluation. In line with the former evaluation, the value of MAP^k decreases when k increases. However, being the behavior of the real-life models more complex than the one of the synthetic models, for some logs (e.g. the BPIC15 logs), it was not possible to compute MAP^4 and MAP^5 for the models discovered by IM. This was due to scalability issues, as the models discovered by IM exhibit flower-like behavior (with more than 50 distinct activities per flower construct). This is reflected by the very low values of MAP^2 and MAP^3 for IM. However, we recall that by design, for small values of k, MAP^k compares small chunks of the model behavior to small chunks of the log behavior. Thus, low values of MAP^k can already indicate poorly-precise models. ETC_a and MAP^5 agreed on the precision ranking 50% of the times. This result is consistent with our qualitative evaluation. Also in-line with the former evaluation, ETC_a showed to be very tolerant to infinite model behavior, regardless of the type of such behavior. The clearest example supporting this flaw is the SEPSIS log case. The models discovered by IM and SM are shown in Fig. 9 and 10. We can see that more than the 80% of the activities in the IM model are skippable and over 60% of them are inside a long loop, resembling a flower construct with some constraints, e.g. the first activity is always the same. Instead, the model discovered by SM, even if cyclic, does not allow many variants of behavior. Consequently, for the IM model, the value of MAP^k drastically drops when increasing k from 2 to 3, whilst it

remains 1 for the SM model. In constrast, ETC_a reports a precision of 0.445 for IM, which is counter-intuitive considering the flower-like model.

Log	BPIC12			$BPIC13_{cp}$			$BPIC13_{inc}$			$BPIC14_f$			$BPIC15_{1f}$		
Miner	**SM**	**IM**	**SHM**	**SM**	**IM**	**SHM**	**SM**	**IM**	**SHM**	**SM**	**IM**	**SHM**	**SM**	**IM**	**SHM**
ETC_a	0.762	0.502	-	0.974	1.000	0.992	0.979	0.558	0.978	0.673	0.646	-	0.880	0.566	-
MAP^2	1.000	0.089	0.083	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.775	0.285	1.000	0.020	0.016
MAP^3	1.000	0.014	0.021	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.754	0.168	1.000	0.003	0.005
MAP^4	0.546	0.002	0.010	1.000	1.000	1.000	1.000	0.990	1.000	1.000	0.750	0.116	1.000	-	0.002
MAP^5	0.234	-	-	1.000	1.000	1.000	1.000	0.861	1.000	1.000	0.718	-	1.000	-	-

Log	$BPIC15_{2f}$			$BPIC15_{3f}$			$BPIC15_{4f}$			$BPIC15_{5f}$			$BPIC17_f$		
Miner	**SM**	**IM**	**SHM**	**SM**	**IM**	**SHM**	**SM**	**IM**	**SHM**	**SM**	**IM**	**SHM**	**SM**	**IM**	**SHM**
ETC_a	0.901	0.556	0.594	0.939	0.554	0.671	0.910	0.585	0.642	0.943	0.179	0.687	0.846	0.699	0.620
MAP^2	1.000	0.024	0.899	1.000	0.035	0.872	1.000	0.017	0.810	1.000	0.007	0.826	0.764	0.604	0.170
MAP^3	1.000	0.003	0.629	1.000	0.004	0.561	1.000	0.002	0.546	1.000	-	0.584	0.533	0.399	0.080
MAP^4	1.000	-	0.380	1.000	-	0.310	1.000	-	0.333	1.000	-	0.371	0.376	0.268	0.039
MAP^5	1.000	-	0.212	1.000	-	0.154	1.000	-	0.189	1.000	-	0.226	0.255	0.172	0.019

Log	RTFMP			SEPSIS			PRT1			PRT2			PRT3		
Miner	**SM**	**IM**	**SHM**	**SM**	**IM**	**SHM**	**SM**	**IM**	**SHM**	**SM**	**IM**	**SHM**	**SM**	**IM**	**SHM**
ETC_a	1.000	0.700	0.952	0.859	0.445	0.419	0.985	0.673	0.768	0.737	-	-	0.914	0.680	0.828
MAP^2	1.000	0.554	0.323	1.000	0.226	0.227	1.000	1.000	0.796	1.000	0.873	1.000	1.000	0.970	0.978
MAP^3	1.000	0.210	0.093	1.000	0.051	0.072	1.000	1.000	0.578	1.000	0.633	1.000	1.000	0.843	0.652
MAP^4	1.000	0.084	0.027	1.000	0.009	0.021	1.000	1.000	0.386	1.000	0.240	0.438	1.000	0.643	0.328
MAP^5	1.000	0.039	0.008	1.000	-	-	1.000	1.000	0.241	1.000	-	0.151	1.000	0.529	0.157

Log	PRT4			PRT6			PRT7			PRT9			PRT10		
Miner	**SM**	**IM**	**SHM**	**SM**	**IM**	**SHM**	**SM**	**IM**	**SHM**	**SM**	**IM**	**SHM**	**SM**	**IM**	**SHM**
ETC_a	0.995	0.753	0.865	1.000	0.822	0.908	0.999	0.726	0.998	0.999	0.611	0.982	0.972	0.790	-
MAP^2	1.000	1.000	1.000	1.000	0.938	0.984	1.000	0.922	0.973	1.000	0.602	0.680	1.000	0.065	-
MAP^3	1.000	1.000	1.000	1.000	0.916	0.946	1.000	0.709	0.742	1.000	0.277	0.294	1.000	0.007	-
MAP^4	1.000	1.000	0.972	1.000	0.622	0.641	1.000	0.596	0.700	1.000	0.121	0.098	0.666	0.001	-
MAP^5	1.000	1.000	0.854	1.000	0.314	0.318	1.000	0.556	0.673	1.000	0.062	0.029	0.434	0.000^*	-

Table 6: Comparison of MAP^k results with $k = 2$–5 using three discovery methods on 20 real-life logs.

	Split Miner				Inductive Miner				Struct. Heuristics Miner			
Precision	**avg**	**max**	**min**	**total**	**avg**	**max**	**min**	**total**	**avg**	**max**	**min**	**total**
ETC_a	60.0	351.9	0.3	720.3	84.2	642.7	0.1	1009.8	34.0	101.4	0.2	305.9
MAP^2	1.9	7.3	0.1	23.2	5.4	15.2	0.1	65.3	6.2	24.4	0.4	74.3
MAP^3	2.0	7.7	0.1	22.5	109.6	426.7	0.1	1205.7	18.5	59.9	0.2	203.7
MAP^4	3.7	16.9	0.2	44.7	927.9^+	3970.5^+	0.1^+	6495.0^+	102.8	476.2	0.1	1233.7
MAP^5	7.3	24.7	0.2	87.9	-	-	-	-	29.8^+	102.2^+	0.2^+	238.1^+

Table 7: Time performance statistics (in seconds) using the twelve public logs (+ indicates a result obtained on a subset of the twelve logs, due to some of the models not being available).

As discussed in Section 3, $k = 2$ is sufficient to satisfy all the 5-Axioms in practice. However, as we also observe from the results of this second experiment, higher values of k lead to finer results for MAP^k. In fact, the notable drops of value from $k = 2$ to $k = 3$ (e.g. in SEPSIS, $BPIC17_f$ and PRT9), confirm that the 5-Axioms are a necessary but not sufficient condition for a reliable precision measure [15].

Finally, Tables 7 and 8 report statistics on the time performance of MAP^k and ETC_a. We divided the results by public and private logs to allow the reproducibility of the

experiments for the set of public logs. We can see that MAP^k scales well to real-life logs, being quite fast for models with a reasonable state-space size (i.e. with non-flower constructs), as those produced by SM and SHM, while ETC_a remains slower even when compared to MAP^5. However, as expected, by increasing k the performance of MAP^k reduces sharply for flower-like models, as those produced by IM.

	Split Miner				Inductive Miner				Struct. Heuristics Miner			
Precision	**avg**	**max**	**min**	**total**	**avg**	**max**	**min**	**total**	**avg**	**max**	**min**	**total**
ETC_a	16.1	106.5	0.2	129.1	16.4	99.2	0.2	114.9	74.3	350.2	0.7	520.1
MAP^2	4.8	32.1	0.1	38.3	6.3	35.6	0.1	50.7	10.6	57.6	0.1	85.2
MAP^3	7.3	51.3	0.1	58.5	11.4	42.6	0.1	91.2	11.7	55.1	0.1	93.6
MAP^4	9.3	58.8	0.1	74.5	121.8	604.7	0.4	974.5	60.9	382.4	0.4	486.9
MAP^5	15.3	71.8	0.1	122.4	711.1	4841.7	0.8	4977.6	75.1	267.8	0.7	525.8

Table 8: Time performance statistics (in seconds) using the eight proprietary logs.

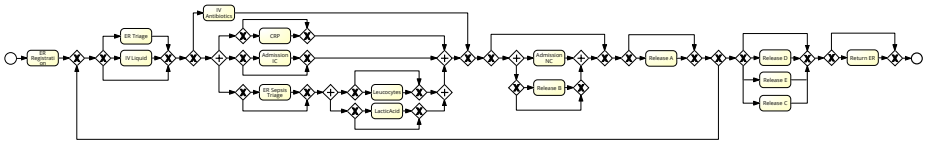


Fig. 9: Model discovered by IM from the SEPSIS log.

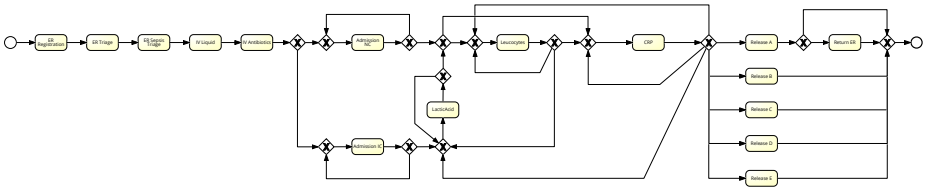


Fig. 10: Model discovered by SM from the SEPSIS log.

5 Conclusion

This paper presented a family of precision measures based on the idea of comparing the k^{th}-order Markovian abstraction of a process model against that of an event log using graph matching algorithms. We showed that this family of precision measures, namely MAP^k, fulfils four of the five axioms of precision of [15] for any value of k and all five axioms for a suitable value of k, dependent on the event log. The empirical evaluation on real-life logs shows that the execution times of the MAP^k (with k up to 5) are considerably lower than those of the ETC_a precision, which is commonly used to evaluate automated process discovery techniques. We also showed on synthetic model-log pairs, that the proposed measure approximates two (unscalable) measures of precision that have been previously advocated as possible ground truths in this field.

Given that our measure abstracts from the model structure and focuses only on its behavior, though in chunks, the only limitation to its usage is scalability, which

indirectly affects also the quality of the results. Even if MAP^k is scalable for acyclic process models, for cyclic real-life models, MAP^k showed to be scalable only for low values of k. Despite the evaluation highlights that low k-orders are sufficient to compare (rank) different models discovered from the same log, higher values of k may return more accurate results.

Possible avenues for future work include the design of more efficient and formally grounded instances of this family of precision measures by exploring alternative behavioral abstractions (besides Markovian ones) and alternative comparison operators.

Acknowledgements. This research is partly funded by the Australian Research Council (DP180102839) and the Estonian Research Council (IUT20-55).

References

1. A. Adriansyah, J. Munoz-Gama, J. Carmona, B. van Dongen, and W. van der Aalst. Measuring precision of modeled behavior. *ISeB*, 13(1), 2015.
2. A. Augusto, R. Conforti, M. Dumas, and M. La Rosa. Automated Discovery of Structured Process Models From Event Logs: The Discover-and-Structure Approach. *DKE*, 2017.
3. A. Augusto, R. Conforti, M. Dumas, M. La Rosa, F.M. Maggi, A. Marrella, M. Mecella, and A. Soo. Automated discovery of process models from event logs: Review and benchmark. *TKDE (to appear)*, 2018.
4. A. Augusto, R. Conforti, M. Dumas, M. La Rosa, and A. Polyvyanyy. Split miner: automated discovery of accurate and simple business process models from event logs. *KAIS*, 2018.
5. A. Augusto, R. Conforti, M. Dumas, and M. La Rosa. Split miner: Discovering accurate and simple business process models from event logs. In *IEEE ICDM*. IEEE, 2017.
6. R. Conforti, M. La Rosa, and A. ter Hofstede. Filtering out infrequent behavior from business process event logs. *IEEE TKDE*, 29(2), 2017.
7. J. De Weerdt, M. De Backer, J. Vanthienen, and B. Baesens. A robust f-measure for evaluating discovered process models. In *IEEE Symposium on CIDM*. IEEE, 2011.
8. G. Greco, A. Guzzo, L. Pontieri, and D. Sacca. Discovering expressive process models by clustering log traces. *IEEE TKDE*, 18(8), 2006.
9. H.W. Kuhn. The hungarian method for the assignment problem. *NRL*, 2(1-2), 1955.
10. S. Leemans, D. Fahland, and W. van der Aalst. Discovering block-structured process models from event logs - a constructive approach. In *Petri Nets*. Springer, 2013.
11. S. Leemans, D. Fahland, and W. van der Aalst. Discovering block-structured process models from event logs containing infrequent behaviour. In *BPM Workshops*. Springer, 2014.
12. S. Leemans, D. Fahland, and W. van der Aalst. Scalable process discovery and conformance checking. *Software & Systems Modeling*, 2016.
13. J. Munoz-Gama and J. Carmona. A fresh look at precision in process conformance. In *BPM*. Springer, 2010.
14. A. Rozinat and W. van der Aalst. Conformance checking of processes based on monitoring real behavior. *ISJ*, 33(1), 2008.
15. N. Tax, X. Lu, N. Sidorova, D. Fahland, and W. van der Aalst. The imprecisions of precision measures in process mining. *Information Processing Letters*, 135, 2018.
16. B. van Dongen, J. Carmona, and T. Chatain. A unified approach for measuring precision and generalization based on anti-alignments. In *BPM*. Springer, 2016.
17. S. vanden Broucke and J. De Weerdt. Fodina: a robust and flexible heuristic process discovery technique. *DSS*, 2017.
18. A. Weijters and J. Ribeiro. Flexible heuristics miner (FHM). In *CIDM*. IEEE, 2011.