Alpha Precision: Estimating the Significant System Behavior in a Model.

Benoît Depaire[1] [0000−0003−4735−0609], Gert Janssenswillen[1] [0000−0002−7474−2088], and Sander J.J. Leemans[2] [0000−0002−5201−7125]

[1] Business Informatics, Hasselt University, Hasselt, Belgium
{benoit.depaire, gert.janssenswillen}@uhasselt.be
[2] Queensland University of Technology, Brisbane, Australia
s.leemans@qut.edu.au

Abstract. One of the goals of process discovery is to construct, from a given event log, a process model which correctly represents the underlying system. As with any abstraction, one does not necessarily want to represent all possible behavior, but only the significant behavior. While various discovery algorithms support this use case of discovering the significant process behavior, proper evaluation measures for this use case appear to be missing.
Therefore, this paper presents a new precision metric that quantifies to what extent the discovered model contains significant system behavior. Besides being a metric with a clear and intuitive interpretation, the metric distinguishes itself in two other areas. Firstly, it introduces the concept of α-significance, which only measures precision with respect to significant behavior. Secondly, it is designed as a system measure and estimates the precision with respect to the underlying system rather than the observed log. This work introduces a new precision measure and a statistical estimation method. Additionally, an empirical demonstration and evaluation of the metric are provided, which creates initial insights and knowledge about the performance and characteristics of the new measure. The results show that the α-precision measure provides a solid foundation for future work on developing quality measures for this particular use case.

Keywords: Process Discovery · Precision · Stochastic Process Models.

1 Introduction

Various information systems increasingly support current business processes, and create a digital trail of process execution information. These digital trails can be transformed into an event log, which records at a minimum the executed activities and their order for each case. Given such event logs, the goal of process discovery is then to discover a model representing the underlying process (also called system) as closely as possible from the event log.

Event logs are only a sample of the possible process or system behavior. Therefore, most process discovery algorithms try to generalize the observed behavior to capture the whole system behavior rather than the log behavior only.

At the same time, a system can contain a large amount of infrequent behavior and trying to represent all this behavior in a single (visual) model quickly results in non-interpretable spaghetti models.

Hence, we focus on the particular use case where one wants to rediscover only the system's significant — typical — behavior. Fortunately, various process discovery algorithms exist that contain mechanisms and parameters that support this use case. That is, discovery algorithms have introduced different ways to classify and filter insignificant behavior: [18] classifies traces of the log as insignificant if they traverse little-used parts of an intermediate behavior abstraction; [8, 7] classify little-used model edges as insignificant; [22, 15] classify edges of a behavior abstraction as insignificant based on frequency; [2, 16] search for a most likely model, thereby implicitly classifying behavior that does not fit that intermediate result as insignificant; and [7] hides insignificant details in hierarchy.

Following the model-log-system quality paradigm in process mining [3], two criteria exist to evaluate the quality of a process model against the system, model-system fitness, and model-system precision. This paper focuses on model-system precision, which quantifies to what extent the process model only contains system behavior. Unfortunately, the existing precision measures fall short of the presented use case for three reasons.

Firstly, existing precision measures do not distinguish between significant (typical) and insignificant (infrequent) process behavior. Consequently, a model that contains a lot of insignificant behavior is still considered to be very precise by these measures, as long as that insignificant behavior is part of the system or log.

Secondly, most precision measures are developed as model-log measures. Consequently, they do not measure to what extent the model only contains behavior from the system, but rather quantify to what extent the model only contains behavior observed in the log. Research has also shown that these model-log measures have limited value when used as proxies for model-system measures [9].

Thirdly, many quality measures in process mining became so advanced over time that an unambiguous interpretation of the precision value is no longer possible. For many measures, the precision value has become a number that is the result of complex computation. While it still correlates to the precision of the model, it lacks a meaningful and unambiguous interpretation.

This research aims to design and introduce a new precision measure that tackles these limitations and (indirectly) supports the use case of discovering significant system behavior. The paper makes three main contributions:

- A first-of-its-kind precision measure is introduced, quantifying the amount of significant behavior in a process model and providing measurement values that have a meaningful and unambiguous interpretation.
- A statistical method based on Bayesian Inference is provided to efficiently estimate the system precision based on a given event log.
- Initial empirical insights into the performance of this new precision measure are provided, which opens up avenues for follow-up research.

The following section provides basic notation and formalization for the remainder of the paper. Section 3 then introduces the rationale, design, and estimation method of the new precision measure. Subsequently, Section 4 provides the empirical evaluation and discussion. Finally, after a brief overview of the related work, the overall conclusions are provided in Section 6

2 Preliminaries

Activities and Traces. A process consists of activities that are represented by their activity labels. The set of all possible activity labels in the system form the activity alphabet $\mathscr{A}$. A trace $\sigma_i \in \mathscr{A}^*$ is a sequence of activity labels, where $\mathscr{A}^*$ is the set of all finite sequences over $\mathscr{A}$. The length of a trace is represented as $|\sigma_i|$.

System. The system $\mathcal{S} = (S, \pi_S)$ represents the underlying process and consists of two components. The first component is the support of the system $S \subseteq \mathscr{A}^*$ which consists of all traces that can be produced by the system. The second component is the system probability distribution $\pi_S : S \rightarrow (0, 1]$, which is a categorical distribution and assigns a probability of occurrence $\pi_S(\sigma_i)$ to each trace σ_i in the system support S, such that $\sum_{\sigma_i \in S} \pi_S(\sigma_i) = 1$. The size of the system corresponds to the number of traces σ_i in the system support and is represented as $K = |S|$. Note that we thus assume S to be finite.

Log. The log $L \in \mathcal{B}(\mathscr{A}^*)$ is a multi-set of traces. The frequency of trace σ_i in the log is denoted by n_{σ_i}, and $N = \sum_{\sigma_i \in L} n_{\sigma_i}$ denotes the size of the log. Note that, as we interpret the system as a categorical distribution over the system support S, the log L is a sample of N drawings from this distribution π_S.

Model. Two type of process models are considered: non-probabilistic and stochastic process models. A non-probabilistic model does not hold any information about the model probability $\pi_M(\sigma_i)$ of a trace σ_i and is simply a set of traces $M \subseteq \mathscr{A}^*$. A stochastic process model $\mathcal{M} = (M, \pi_M)$ is more informative as it consists of a set of traces $M \subseteq \mathscr{A}^*$ that represents the models support and a model probability distribution $\pi_M : M \rightarrow (0, 1]$ where $\pi_M(\sigma_i)$ represents the probability of trace σ_i according to the model, such that $\sum_{\sigma_i \in M} \pi_M(\sigma_i) = 1$. The number of traces in the model is denoted as $|M|$.

3 Alpha Precision

The overall goal is to design a model-system precision measure for the use case of discovering a process model which only contains significant system behavior. Furthermore, the measure should meet the following three design requirements:

Requirement 1 *The precision measure should quantify to which extent the process model (only) contains significant process behavior.*

Requirement 2 *The precision measure should quantify the precision of the model with respect to the system.*

Requirement 3 *The precision measure should produce values that have an unambiguous and human-interpretable meaning.*

3.1 Rationale and Design

In order to meet the first two design requirements, the following definition first introduces the concept of α-significance, which identifies a trace as significant if its system probability $\pi_s(\sigma_i)$ exceeds a user-defined threshold α.

Definition 1 (α-significance). *A trace σ_i is α-significant iff $\pi_s(\sigma_i) > \alpha$*

Based on this concept, we can define the α-indicator function I_α.

Definition 2 (α-indicator function).

$$I_\alpha(\sigma_i) = \begin{cases} 1, & \textit{if } \pi_S(\sigma_i) \geq \alpha \\ 0, & \textit{otherwise} \end{cases} \tag{1}$$

For a stochastic model, $\mathcal{M}$, α-precision P_α is then defined as the probability that the model produces a trace that is α-significant.

Definition 3 (α-precision (stochastic model)). *Let $\mathcal{M}$ be a stochastic process model, then*

$$P_\alpha(\mathcal{M}, \mathcal{S}) = \sum_{\sigma_i \in M} \pi_M(\sigma_i) I_\alpha(\sigma_i) \tag{2}$$

For non-probabilistic models, α-precision is defined as the portion of α-significant traces in the model M.

Definition 4 (α-precision (non-probabilistic model)). *Let M be a non-probabilistic process model, then*

$$P_\alpha(M, \mathcal{S}) = \frac{1}{|M|} \sum_{\sigma_i \in M} I_\alpha(\sigma_i) \tag{3}$$

The third design requirement involves a meaningful and clear interpretation of the new precision measure. In order to illustrate the interpretability of the proposed α-precision, consider the following application scenario:

> A data scientist wants to discover a process model from an event log that contains the significant (typical) behavior. The goal is to understand the standard way of working within the department and not depict exceptional process executions in the process model. First, they set the α threshold at 1 percent, which means that any trace that has a probability less than 1 percent is considered non-significant. Next, the discovered stochastic model appears to have an $P_\alpha = 0.8$. This value tells her that 80 percent of the traces generated by this process model are expected to be significant, i.e., 80 percent of the behavior produced by the model has a system probability $\pi_S(\sigma_i)$ greater than 1 percent.

Note how both the α-threshold and the α-precision have natural interpretations that allow users to use context and domain-expertise to set a proper threshold and interpret and evaluate precision levels found for their discovered models. Also, note that in the case of a deterministic model, the interpretation of the precision measure would only slightly change to the conclusion that 80 percent of the traces contained in the model are significant.

3.2 Estimation Method

Calculating the α-precision is straightforward when the system $\mathcal{S}$ and the its probability distribution π_S are known. However, in real-life, one does not know the system. The only available information is typically an event log — a limited sample of the system's behavior. Therefore, we introduce a method to estimate the true α-precision from the available information in the log.

The general idea behind the method is to estimate the system probabilities $\widehat{\pi_S}$ from the event log. Next, these estimates are used to estimate the indicator function (cfr. Eq. 1), which subsequently is used to estimate the α-precision $\widehat{P_\alpha}$. Thus, the estimation problem reduces to the estimation of the system probabilities from the event log. The proposed method is based on Bayesian Inference and inspired by the work of [10].

To infer knowledge about the system from the log, we need additional assumptions about the system.

Assumption 1 *The system contains a finite amount of behavior.*

This assumption implies that the system has some mechanism that prevents a process from being executed indefinitely. For business processes with humans involved, this is a fair assumption.

Assumption 2 *The system support is correctly defined.*

This assumption restricts the modeling of uncertainty to the system probability density. The assumption that the system support is correctly defined corresponds to the common assumption in statistics of correct model specification. While it is hard to prove that the system support is defined correctly, it is essential to realize that any theoretically possible trace is part of the system support, no matter how small the probability of occurrence.

Assumption 3 *The log is a representative sample from the system's behavior*

As the proposed method will rely on statistical inference, this assumption is required to draw proper conclusions from the data for the underlying system. Considering these assumptions, the α-precision can be estimated in four steps.

Step 1: Define the system support. First, the system support S of the system $\mathcal{S}$ needs to be specified. Under the assumption that the system behavior is finite, the system support can be defined as a set of traces σ_i for which the system probability $\pi_S(\sigma_i) > 0$.

Step 2: Define a prior distribution over $\boldsymbol{S}$. As indicated before, a system $\mathcal{S} = (S, \pi_S)$ consists of two components: its support and a probability function. The latter can be defined as the vector $\pi_S = (\pi_S(\sigma_1), \ldots, \pi_S(\sigma_K))$ of system probabilities, where K is the size of the system.

However, because the actual system is unknown, we do not know the true categorical probability function. In order to model this uncertainty, we consider all theoretically possible categorical probability functions for K possible outcomes and assign a probability to each one of them. This is modeled as a Dirichlet distribution.[12]

From the perspective of Bayesian Inference, the first step is to encode the prior belief about the system as the prior distribution. In this paper's context, the prior belief refers to the knowledge about the system probability function π_S before observing the data. Assuming that there is no specific information to favor one probability function over the other, a flat Dirichlet distribution is chosen as the prior. This distribution is equivalent to a uniform distribution over all possible system probability distributions π_S and achieved by setting all the Dirichlet parameters θ_i to 1.

Step 3: Determine Posterior Distributions. The flat Dirichlet distribution from the previous step represents our prior belief that all possible probability functions π_S are equally likely. However, once we have observed an event log, we notice that some traces are more common than others, indicating that some probability functions π_S must be more likely than others.

Bayesian inference uses Bayes' theorem to update our prior beliefs with the evidence in the log, which results in a posterior distribution [6]. In most situations, the posterior distribution is not obtainable analytically unless the prior distribution is conjugate to the likelihood distribution of the data. If this is the case, the posterior distribution can be analytically calculated from the prior distribution and the data.

Because the Dirichlet distribution is a conjugate prior to the multinomial distribution and the event log is a multinomial distribution, the posterior distribution is also a Dirichlet distribution. More specifically, the posterior distribution will be a Dirichlet distribution with parameters $\theta'_i = \theta_i + n_i$, where θ_i represents the i-th parameter of the prior distribution and n_i represents how often outcome i was observed in the data.

Given a flat Dirichlet distribution as the prior distribution and our event log L being a multi-set of traces σ_i, this results in the following posterior distribution:

$$\pi_S | L \sim \mathrm{Dir}\left((1 + n_{\sigma_1}, \ldots, 1 + n_{\sigma_K})\right) \tag{4}$$

Note that the posterior distribution assigns a probability to each possible system probability distribution π_S based on the evidence in the log.

Step 4: Estimate $\boldsymbol{\alpha}$-Precision. Now that the posterior distribution over all possible system probability functions is known, the true system probability function can be estimated by taking the expected value of the Dirichlet posterior, which is defined as follows:

Definition 5. *Let $X = (X_1, \dots X_K) \sim \mathrm{Dir}(\theta')$, then the expected value of X_i is*

$$E[X_i] = \frac{\theta'_i}{\sum_{k=1}^{K} \theta_k} \tag{5}$$

Given that the parameters of the posterior distribution are $\theta'_i = 1 + n_{\sigma_i}$ for $1 \leq i \leq K$, we can estimate the system probabilities $\pi_S(\sigma_i)$ as follows:

$$\forall \sigma_i \in S : \widehat{\pi_S(\sigma_i)} = \frac{1 + n_{\sigma_i}}{\sum_{k=1}^{K}(1 + n_{\sigma_k})} = \frac{1 + n_{\sigma_i}}{K + N} \tag{6}$$

Based on this estimator for system probability, we can subsequently estimate the α-significance indicator function as follows:

$$\widehat{I_\alpha(\sigma_i)} = \begin{cases} 1, & \text{if } \widehat{\pi_S(\sigma_i)} \geq \alpha \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

Using this indicator function in Equations 2 and 3, for stochastic and non-probabilistic models respectively, we can then estimate the α-precision. We will denote this estimated precision as $\widehat{P_\alpha}$.

3.3 Estimating the System Support Size

Analyzing the final equations of the proposed estimation method reveals that it requires two parameters: the α-significance level and the theoretical system support size K. While the former can be freely chosen and should reflect the domain expert's interpretation of significance, the latter should preferably match its theoretical value. Various approaches to estimate K can be devised. This paper proposes two approaches — the *unrestricted* and *restricted* approach — and motivates them both based on underlying assumptions.

The unrestricted approach is so-called because it does not strongly limit the behavior that is included in the system support, except for the alphabet of activity labels observed in the event log $\mathscr{A}$ and a user-defined maximum trace length γ. It then defines the system support S as the set of all possible sequences σ_i over $\mathscr{A}$ with a length $|\sigma_i| \leq \gamma$. This approach assumes that the entire activity alphabet has been observed and a maximum trace length exists. The rationale behind the maximum trace length assumption is that the system would not allow a process instance to keep ongoing indefinitely. Given the alphabet $\mathscr{A}$ and the maximum trace length γ, the size of the system can be calculated as follows: $K = \sum_{i=1}^{\gamma} \mathscr{A}^i$.

The restricted approach can be seen as taking the system support S from the first approach as its starting point but removing all traces that contain a directly-follows relation not observed in the log. The assumption thus is that all possible directly-follows relations have been observed in the log.

Suppose that the directly-follows relations are represented by a matrix $\mathbf{D}$ of size $|\mathscr{A}| \times |\mathscr{A}|$ where D_{ij} equals 1 if and only if it was observed in the log that activity i of the alphabet was directly followed by the activity j, and 0 otherwise. It follows naturally that the number of allowed sequences of length two is equal

Table 1. Parameters.

	Parameter	values
System	`Alphabet length`	[4, 6, 8]
	`Max trace length`	[4, 6]
Log	`Log size`	[100, 500, 1000 , 5000, 10000]
Model	`Discovery threshold`	[0.3, 0.4, 0.5, 0.6, 0.7]
	`# iterations`	25

to the sum over this matrix, i.e., $\sum_{\mathbf{D}} D_{ij}$. Furthermore, assume a vector $\mathbf{o}$ of size $|\mathscr{A}|$, where o_i is 1 if and only if activity i in the alphabet is a valid start activity, and 0 otherwise. The number of sequences of length one with a valid start activity is then equal to the sum over vector $\mathbf{o}$, i.e., $\sum_{\mathbf{o}} o_i$. The number of sequences of length two with a valid start activity is equal to the sum over $\mathbf{o^T D}$. This can be generalized to $\mathbf{o^T D}^{\gamma-1}$ for sequences of length γ. In order to limit the number of sequences to specific final activities, a vector $\mathbf{f}$ of size $|\mathscr{A}|$ can be defined where f_i is 1 if and only if activity i is a valid end activity, and 0 otherwise. The scalar $\mathbf{o^T D}^{\gamma-1}\mathbf{f}$ then equals the total number of sequences of length γ with valid start and end points.[3] As a result, for the restricted approach, $K = \sum_{i=1}^{\gamma} \mathbf{o^T D}^{i-1}\mathbf{f}$, which is computationally easy to calculate. Note that the restriction of valid start and end activities can be omitted without difficulty depending on the specific context.

4 Empirical Evaluation

This Section provides an empirical evaluation of the α-precision by means of a controlled experiment and a application on real-life data. The goal is to provide insights and knowledge claims about the behavior and performance of this newly developed measure.

4.1 Unbiasedness Estimator

In this section, we describe a controlled experiment to analyze the unbiasedness of the α-precision estimator. The experiment exists of the following six steps: (1) Generate systems, (2) Generate logs, (3) Generate models, (4) Calculate actual α-precision, (5) Calculate estimated α-precision, and (6) Analyze bias.

Generate systems. First, different systems were generated using the alphabet length and maximum trace lengths in Table 1 as input parameters. Each of the system-traces is assigned a probability $\pi_S(\sigma_i)$.

Because we want define the full system including probability distribution in this supervised experiment, both the size of the alphabet and the maximum possible trace length are kept relatively low. An alphabet of length 10 with a maximum trace length of 8 leads to 1.23×10^9 possible systems sequences.

[3] In the specific case that $\gamma = 1$, $\mathbf{D}^0$ equals the identity matrix $\mathbf{I}$, and thus $\mathbf{o^T D}^{\gamma-1}\mathbf{f} = \mathbf{o^T f}$, which is the number of activities that are both valid start and end activities. This is indeed equal to the number of valid sequences of length one.

Generate logs. For each of the systems, logs of different sizes (cfr. Table 1) are generated, using the defined system probabilities for each trace.

Generate models. Subsequently, a model is generated based on each system. A discovery threshold is set to steer the amount of the system that is captured by the model. A discovery threshold of 0.7 means that each trace has a 70% probability to be included in the model.

To each of the traces that is included in the model, a probability is assigned in order to turn it into a stochastic model as defined in Section 2. These probabilities are independent from the system probabilities created in step 1, thereby making sure no algorithm bias is introduced.

For each of the combinations of parameters listed in Table 1, we repeat this process 25 times.

Calculate actual α-precision. Given that we know the actual system probability distribution π_S, the actual α-precision can be computed using Equation 2. Because there is no domain expertise in this artificial setting to define the level of α, a rule of thumb was used to set α equal to $\frac{1}{K}$ where K is the size of the system support.

Calculate estimated α-precision. Given the model and log, we can then estimate the α-precision by using Equation 2 in combination with the estimated system probabilities as defined in Equation 6.

Next to the information provided in the model and the log, we need to define the value of K and α. For K, the size of the system, we take the *unrestricted* approach described in Step 1 of Section 3.2, where we consider all possible sequence of the alphabet. Subsequently, α is also set to $\frac{1}{K}$ for the estimation.

Measure bias. Given both the actual α-precision and the estimated α-precision, we define the difference between the two as follows.

$$\beta = \widehat{P_\alpha} - P_\alpha \tag{8}$$

When the estimated α-precision is greater than the actual α-precision, the bias as measured by β will be positive, and there is thus an overestimation. Otherwise, β will be negative and the actual α-precision will thus be underestimated.

Results. In Figure 1, it can be seen that the estimator underestimates the real precision when the size of the log is relatively small compared to the size of the system. For the system with size 340 (i.e. alphabet length of 4 and maximum trace length of 4), the estimator becomes unbiased when approximately 1000 cases have been observed in the log. For larger systems, the biases only approaches zero for logs of 10000 cases, while for the largest systems in the experiment the estimators still shows a large bias at logs of size 10000.

Figure 2 shows the extent of bias specifically in relation to the ratio between the log size and the system size. The vertical line indicates the where the ratio is 1, i.e. the number of different sequences in the system support equals the number of observed traces in the log. It can be seen that the biases quickly decreases when the ratio approaches 1, and then decreases more gradually toward zero for ratios greater than 1.

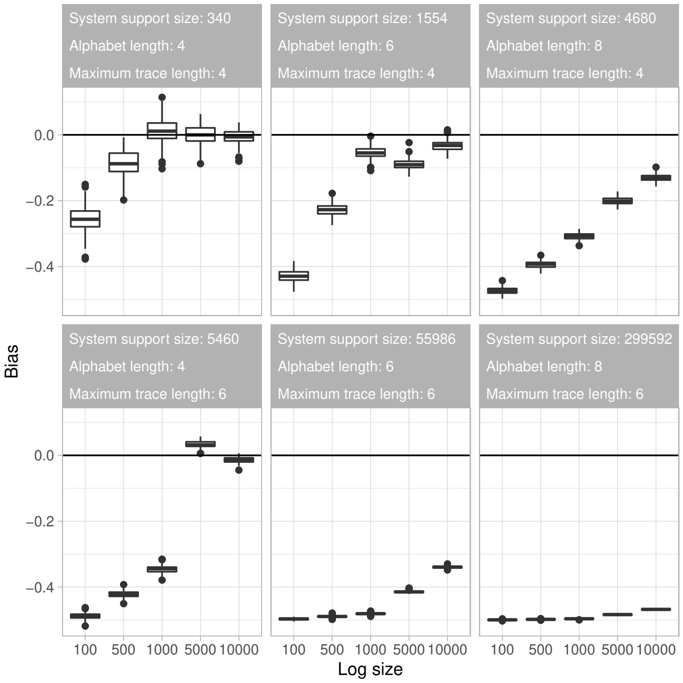

Fig. 1. Bias $\widehat{P_\alpha} - P_\alpha$ for different sizes of the log and the system.

Table 2. BPIC'12 log - descriptive statistics.

Metric	value	metric	value
Number of activities	10	Number of traces	17
Number of events	60849	Max. trace length	8
Number of cases	13087	Avg. trace length	4.65

4.2 Demonstration on real-life event logs

In this and the next section, the proposed method will be applied on real-life event data. For this, data from the Business Process Intelligence Challenge 2012 is used [5]. Descriptive statistics for this event log can be found in Table 2.

Based on the log, a stochastic model has been discovered using the frequency estimator [4] on a model discovered by the Directly Follows Model Miner [18]. The discovered model contains 6 different activity sequences, of which the probability varies between 0.029 and 0.509. In the analyses, we will both approach the estimation from the starting point of a stochastic model (Eq. 2), as from the starting point of a deterministic model (Eg. 3). In the latter case, we will ignore the obtained probabilities and replace them with $\frac{1}{|M|}$.

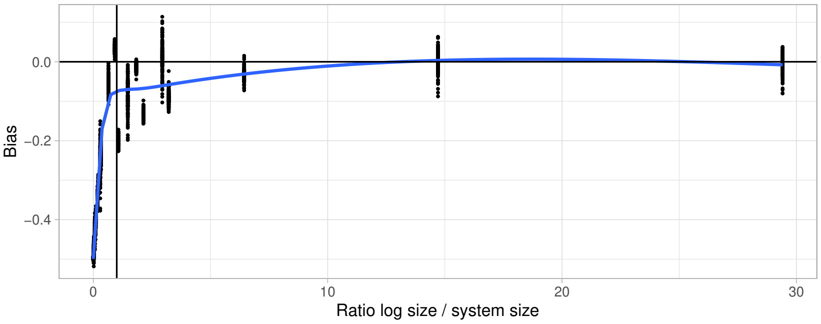

Fig. 2. Extent of bias in estimated *alpha*-precision in relation to the ratio between log size N and system size K.

4.3 Impact of K and α.

As discussed before, there are different approaches to define the size of the system support K. In the unrestricted approach, where we look only at the length of the alphabet and the maximum trace length, we get $K = \sum_{i=1}^{8} 10^i = 1.11 \times 10^8$. Given the fairly high structuredness of the data in question, this seems an exuberant amount. If we therefore take the restricted approach instead, where we only take into account sequences that adhere to the observed directly-follows relations, start activities and end activities, the system support K' is only 32. When we drop the start and end activity requirement, K'' equals 1539.

Fig. 3 shows the value of $\widehat{P_\alpha}$ for different values of K and different values of α for both the stochastic and deterministic approach. It can be seen that the estimated α-precision is relatively stable with respect to the value of K. Only when K is increased to 50000 can noticeable differences in $\widehat{P_\alpha}$ be seen.

Note that as the model only contains 6 different activity sequences, we can see apparent jumps in the measured precision when α changes such that a traces moved from insignificant to significant. For the deterministic model all jumps are equal in size, while this is naturally not the case for the stochastic model. The biggest jump in the latter case happens when α drops below 0.05 (approximately), and the trace with the highest probability according to the model (0.509) becomes significant.

Impact of significance filtering. Figure 4 shows the values of $\widehat{P_\alpha}$ for models discovery by the Inductive Miner infrequent [16] and Directly-follows miner [18], with different setting for significance filtering. The higher the significance parameter, the more significant behavior must be to make it into the model.

It can be seen that, for different values of alpha, when the significance parameters increases, so does the estimated precision. This provides implicit validation that the proposed measure behaves as expected, as models with a stronger significance filtering gets higher precision scores.

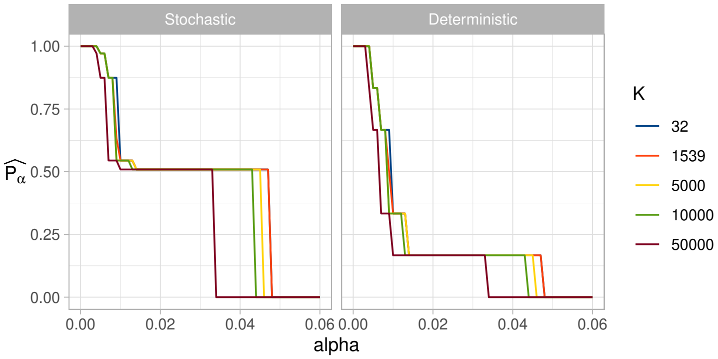

Fig. 3. $\widehat{P_\alpha}$ for different values of K and alpha.

Figure 4 also shows the potential of the α-precision to analyze and compare various algorithms. The visual analysis shows that for the DFM-algorithm, the evolution of precision with respect to the significance parameter is smoother than for the Inductive Miner algorithm. This implies better control for significance filtering in the former algorithm. This derives from the fact that Inductive miner often results in the same model for various significance filtering levels. In the extreme, the measured precision drops to zero for the DFM algorithm with the filtering parameter set to 100, as this results in an empty model.

4.4 Discussion

Based on the design and empirical evaluation of the α-precision measure and its estimation method, various knowledge insights can be constructed about the precision measure.

The controlled experiment showed that the measure is unbiased when the size of the log is sufficiently large in comparison with the system sizes. When there are fewer cases in the log than there are different activity sequences in the system, a substantial underestimation is present. In those circumstances, the estimated α-precision acts as a lower bound of the actual measure.

At the same time, the demonstration of the measure on real-life data shows that the proposed measures behave as expected for different values of K, α and the significance filter parameters of process discovery algorithms and holds potential to evaluate process discovery algorithms aimed at discovering significant behavior. While defining the system support size K is an important step towards estimating the α-precision, it has been shown that the estimator is relatively robust for changes in K.

An ongoing discussion in the field of process mining is that of (desirable) properties (or axioms) of conformance measures [21]. However, the properties studied in past research are not applicable to α-precision as most properties are defined with respect to non-stochastic process models. The exception are the eight properties for stochastic conformance checking defined in [17]. Unfor-

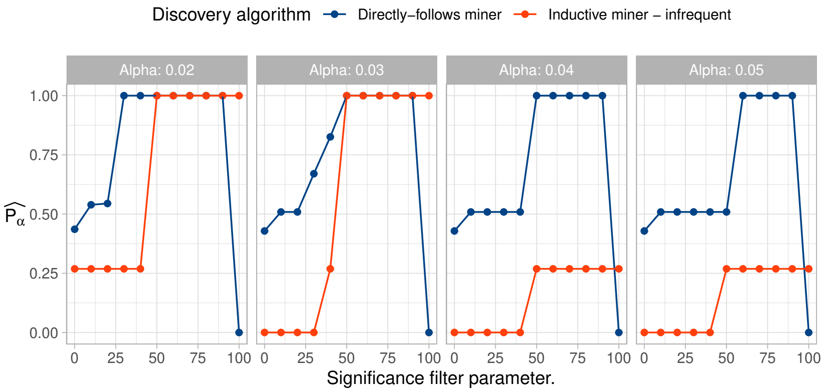

Fig. 4. $\widehat{P_\alpha}$ for different values in the significance filtering parameter of [16] and [18], and different values of α. Note that the values of the significance parameters for the Directly-follows miner has been inverted for the sake of comparison. A small significance parameter value means that only very insignificant behavior is removed by the discovery algorithm, a large value means that only very significant behavior is retained by the discovery algorithms.

tunately, even these properties are not directly applicable because they relate to log-model measures, while α-precision is a system-model measure. Nevertheless, we can derive four properties for the α-precision which are inspired by the properties discussed in [17].

Property 1. The α-precision measure is deterministic.

This property relates directly to property P1 in [17] and holds as α-precision is a function of the assumed system support size K, the α-significance level, the log size N and the trace frequencies n_{σ_i}, which are all fixed at the start.

Property 2. The α-precision measure depends only on the stochastic language of the log and model and not on their representation.

This property refers directly to property P2 in [17] and holds naturally, as the α-precision is calculated directly from the stochastic language of the log and model.

Property 3. The α-precision measure returns values between 0 and 1.

This property relates directly to property P3 in [17]. Since the α-indicator function $\widehat{I_\alpha(x)}$ is either 0 or 1 (cf. Eq. 7), it follows that the minimum and maximum value of the α-precision (cf. Def. 3 and 4) is also 0 and 1 and can only be achieved when all α-indicator functions evaluate to 0 or 1 respectively.

Property 4. The α-precision measure asymptotically goes to 1 if (i) the model only contains the α-significant system behavior, (ii) the log has the same stochastic language as the system and (iii) the log size increases towards infinity.

This property is an adaptation of property P4 in [17] to the context of our measure. If the log size N goes towards infinity, then the system probability estimates (cf. Eq. 6) will go towards $\frac{n_{\sigma_i}}{N}$. The latter equates to the true system probabilities since the log and the system express the same stochastic language. Consequently, if the model only contains α-significant system behavior, the indicator function will evaluate to 1 for all traces in the model and the α-precision measure will equate to 1.

The other properties P5, P7 and P8 in [17] are not directly applicable to our measure. Properties P5, P6 and P8 are related to recall measures rather than precision measures.While property P6 does relate to precision measures, it focuses on a log-model relation which doesn't have a clear analog counterpart in the system-model context we are operating.

5 Related Work

In typical process mining projects, the system is unknown; thus, quality measures (conformance checking techniques) have focused on the relationship between model and log, rather than system and model. The quality of (non-stochastic) models with respect to logs is typically measured using fitness, precision, generalization, and simplicity, where fitness is the fraction of behavior of the log that is in the system, precision is the fraction of behavior in the model that was observed in the log, generalization is the predicted fraction of future behavior of the system that is in the model, and simplicity expresses the size or complexity of the model to express its behavior [3]. Recently, the concept of precision (and to a lesser extent generalization) has seen discussion in terms of desirable properties such measures should possess [21]; however, this discussion has not yet included unbiasedness with respect to unknown systems. Of these quality dimensions, generalization aims to describe the system and could be seen as a system-fitness measure [20]. Such log-based measures are not unbiased estimators of system properties empirically [9]. Compared to these approaches, our proposed measure explicitly and understandably takes the significant behavior of an unknown system into account.

For stochastic process models, quality measures include stochastic distance [13], stochastic precision and recall [17], and entropic relevance [19], however these do not aim to compare a model with an unknown system. While not intended for the system-model context, it would be interesting to study the bias of these techniques when applied in a system-model context, like [9, 11].

The system has been the subject of study in process mining, as the ultimate goal of process mining is to obtain insights into the system to improve it. Some process discovery techniques guarantee to return a model that is the language equivalent to the system, under some assumptions, such as the log being noise-free or complete with respect to a particular abstraction of the system [14, 23, 1]. However, such techniques do not offer any guarantees when these assumptions are not met; thus, it is a valuable exercise to have an unbiased estimator of the relation between system and model.

6 Conclusions

In process mining, organizations aim to gain insights into their business processes, which we refer to as systems, by discovering process models from event logs. Typically, the quality of a process model is assessed with respect to an event log, however we argue that it might be useful to compare a model to the unknown system, based on its significant behavior. In this paper, we presented a new precision metric that expresses the extent to which the model contains significant system behavior, based on an α-significance level. We empirically evaluated the new measure by showing that it can be unbiased under certain assumptions and demonstrated its applicability and value on real-life event logs.

While the initial results indicate that this precision measure supports the analysis of discovery algorithms aimed at discovering significant behavior, the empirical analysis also shows that the construction of unbiased system estimators is particularly challenging and requires future research to better understand and remove this apparent bias. Important aspects to consider are a more realistic definition of ground-truth systems in controlled experiments, as well as the proper estimation of the system size when using the estimator. The impact of prior configurations, which are currently uninformative, is another aspects that requires further analyses.

Overall, we hope this work provides an initial yet solid foundation for further research into system measures supporting the use case of discovering significant behavior.

References

1. Badouel, É.: On the α-reconstructibility of workflow nets. In: Petri nets. LNCS, vol. 7347, pp. 128–147 (2012)
2. Brons, D., Scheepens, R., Fahland, D.: Striking a new balance in accuracy and simplicity with the probabilistic inductive miner. In: ICPM. pp. 32–39. IEEE (2021)
3. Buijs, J.C.A.M., van Dongen, B.F., van der Aalst, W.M.P.: Quality dimensions in process discovery: The importance of fitness, precision, generalization and simplicity. Int. J. Cooperative Inf. Syst. **23**(1) (2014)
4. Burke, A., Leemans, S.J.J., Wynn, M.T.: Stochastic process discovery by weight estimation. In: ICPM Workshops. LNBIP, vol. 406, pp. 260–272 (2020)
5. van Dongen, B.: Bpi challenge 2012 (Apr 2012). https://doi.org/10.4121/uuid:3926db30-f712-4394-aebc-75976070e91f
6. Gelman, A., Carlin, J.B., Stern, H.S., Dunson, D.B., Vehtari, A., Rubin, D.B.: Bayesian Data Analysis. Taylor & Francis Ltd, Boca Raton (Nov 2013)
7. Günther, C.W., van der Aalst, W.M.P.: Fuzzy mining - adaptive process simplification based on multi-perspective metrics. In: BPM. LNCS, vol. 4714, pp. 328–343 (2007)
8. Günther, C.W., Rozinat, A.: Disco: Discover your processes. In: BPM. CEUR Workshop Proceedings, vol. 940, pp. 40–44. CEUR-WS.org (2012)
9. Janssenswillen, G., Depaire, B.: Towards confirmatory process discovery: Making assertions about the underlying system. Bus. Inf. Syst. Eng. **61**(6), 713–728 (2019)
10. Janssenswillen, G., Depaire, B., Faes, C.: Enhancing discovered process models using bayesian inference and mcmc. In: BPM. pp. 295–307 (2020)

11. Janssenswillen, G., Jouck, T., Creemers, M., Depaire, B.: Measuring the quality of models with respect to the underlying system: An empirical study. In: BPM. LNCS, vol. 9850, pp. 73–89 (2016)
12. Kotz, S., Balakrishnan, N., Johnson, N.L.: Continuous multivariate distributions, Volume 1: Models and applications, vol. 1. John Wiley & Sons (2004)
13. Leemans, S.J.J., van der Aalst, W.M.P., Brockhoff, T., Polyvyanyy, A.: Stochastic process mining: Earth movers' stochastic conformance. Inf. Syst. **102**, 101724 (2021)
14. Leemans, S.J.J., Fahland, D.: Information-preserving abstractions of event data in process mining. Knowl. Inf. Syst. **62**(3), 1143–1197 (2020)
15. Leemans, S.J.J., Fahland, D., van der Aalst, W.M.P.: Discovering block-structured process models from event logs containing infrequent behaviour. In: BPM Workshops. LNBIP, vol. 171, pp. 66–78 (2013)
16. Leemans, S.J.J., Fahland, D., van der Aalst, W.M.P.: Discovering block-structured process models from incomplete event logs. In: Petri nets. LNCS, vol. 8489, pp. 91–110 (2014)
17. Leemans, S.J.J., Polyvyanyy, A.: Stochastic-aware conformance checking: An entropy-based approach. In: CAiSE. LNCS, vol. 12127, pp. 217–233 (2020)
18. Leemans, S.J.J., Poppe, E., Wynn, M.T.: Directly follows-based process mining: Exploration & a case study. In: ICPM. pp. 25–32. IEEE (2019)
19. Polyvyanyy, A., Moffat, A., García-Bañuelos, L.: An entropic relevance measure for stochastic conformance checking in process mining. In: ICPM. pp. 97–104. IEEE (2020)
20. Syring, A.F., Tax, N., van der Aalst, W.M.P.: Evaluating conformance measures in process mining using conformance propositions. Trans. Petri Nets Other Model. Concurr. **14**, 192–221 (2019)
21. Tax, N., Lu, X., Sidorova, N., Fahland, D., van der Aalst, W.M.P.: The imprecisions of precision measures in process mining. Inf. Process. Lett. **135**, 1–8 (2018)
22. Weijters, A., van Der Aalst, W.M., De Medeiros, A.A.: Process mining with the heuristics miner-algorithm. Technische Universiteit Eindhoven, Tech. Rep. WP **166**, 1–34 (2006)
23. van Zelst, S.J., van Dongen, B.F., van der Aalst, W.M.P.: Avoiding over-fitting in ilp-based process discovery. In: BPM. LNCS, vol. 9253, pp. 163–171 (2015)