All That Glitters Is Not Gold

Towards Process Discovery Techniques with Guarantees

Jan Martijn E. M. van der Werf[1], Artem Polyvyanyy[2], Bart R. van Wensveen[1], Matthieu Brinkhuis[1], and Hajo A. Reijers[1]

[1] Utrecht University, Princetonplein 5, 3584 CC Utrecht, The Netherlands
{j.m.e.m.vanderwerf, m.j.s.brinkhuis, h.a.reijers}@uu.nl
bart@architecturemining.org
[2] The University of Melbourne, Parkville, VIC, 3010, Australia
artem.polyvyany@unimelb.edu.au

Abstract. The aim of a process discovery algorithm is to construct from event data a process model that describes the underlying, real-world process well. Intuitively, the better the quality of the input event data, the better the quality of the resulting discovered model should be. However, existing process discovery algorithms do not guarantee this relationship. We demonstrate this by using a range of quality measures for both event data and discovered process models. This paper is a call to the community of IS engineers to complement their process discovery algorithms with properties that relate qualities of their inputs to those of their outputs. To this end, we distinguish four incremental stages for the development of such algorithms, along with concrete guidelines for the formulation of relevant properties and experimental validation. We use these stages to reflect on the state of the art, which shows the need to move forward in our thinking about algorithmic process discovery.

Keywords: Process mining · process discovery · formal guarantees · properties.

1 Introduction

Process mining focuses on the extraction of process-related information from event logs, a collection of sequences of actions, each encoding a historical process execution [1]. Process discovery is a core area in process mining. It studies algorithms that, given an event log, construct process models that aim to describe the corresponding true process that induced the event log as closely as possible. One of the main challenges in process discovery is that the true process is unknown, and has to be inferred from a *sample* observed and recorded in the event log [10].

An algorithm is a sequence of computational steps that transform a given *input* into some *output* [11]. Different algorithms exhibit different properties, for example, correctness, finiteness, definiteness, effectiveness, and efficiency. Such properties allow us to choose an algorithm that fulfills a certain need, such as performing a guaranteed correct computation within the desired time bounds. A process discovery algorithm transforms a given input event log into an output process model. We usually expect that a process discovery algorithm is finite (terminates after a finite number of computational steps), definite (each computational step is unambiguous), effective (each

computational step can be performed correctly in a finite amount of time), and efficient (the fewer or faster computation steps can be executed the better). However, process discovery algorithms treat quality as a *goal* rather than a guarantee. That is, process discovery algorithms are designed to construct a "good" process model from the input event log [1], where the "goodness" of the model is not established by the internals of the algorithm, but by external measures, e.g., precision and recall.

In this paper, we recommend refining the process discovery goal. Our recommendation is triggered by the observation that a process discovery algorithm can construct a good model from an event log yet discover a worse model from another event log of better quality [23]. We argue that process discovery algorithms should come with guarantees formulated in terms of the relationship between the quality of its inputs and outputs. The present paper makes these contributions:

- We propose measures for the quality of event logs, both in the presence and absence of a true process. In the former case, we use standard conformance checking measures, while in the latter case we rely on sampling techniques and measures as studied in statistics;
- We provide empirical evidence that existing process discovery algorithms can construct good models from event logs and, at the same time, produce poor models from better logs;
- We propose four stages for process discovery algorithms to guarantee the intuitively appealing dependency between the quality of input event logs and the quality of output process models.

We believe that a next step in the evaluation of process discovery algorithms is necessary for the field to advance. Several benchmarks (cf. [5]) have identified process discovery algorithms that "glitter", that is, algorithms that produce high-quality models on a limited collection of event logs. We argue that such benchmarks should be complemented with formal analyses to provide quality guarantees with the algorithms, extending the current state-of-the-art evaluation with statistical methods to establish a relation between log and model quality. We invite the process mining community to contribute to the discussion of the maturity of process discovery algorithms. In addition, we encourage the authors of existing and future discovery techniques to establish the proposed guarantees.

The remainder of the paper is structured as follows. The next section introduces the intuition why process discovery algorithms need to provide guarantees. A statistical approach to establish event log quality is introduced in Section 3. The proposed four stages of process discovery algorithms are presented in Section 4, together with empirical evidence that algorithms do not provide such guarantees yet. Last, Sections 5 and 6 are devoted to related work, and conclusions, respectively.

2 Setting the Stage

2.1 Process Discovery and Conformance Checking

Process mining projects often start by assuming that some underlying process generates an event log that can be observed, recorded, and used for process discovery. We

refer to this underlying entity as the *true process*. The true process is, however, often unknown [10]. Hence, it can only be approximated. Therefore, based on the observed log, process discovery algorithms aim to construct a process model that describes the true process well. Formally, given a set of activities A, an event log L is defined as a multiset over finite sequences, called *traces*, over A. A discovery algorithm disc can be described as a relation $\mathrm{disc} \subseteq \mathcal{L}(A) \times 2^{\mathcal{M}(A)}$, where $\mathcal{L}(A)$ and $\mathcal{M}(A)$ are the universe of all possible logs and the universe of all models over A, respectively. Some algorithms, such as the ILP-miner [31], are non-deterministic, i.e., applying a process discovery algorithm may yield different results for the same input log.

To measure how well the discovered process models describes the behavior recorded in the event log, different conformance measures have been proposed [28]. *Precision* is a function $\mathrm{prec} : \mathcal{L}(A) \times \mathcal{M}(A) \to [0, 1]$ that quantifies the fraction of behavior allowed by the model that was actually observed. *Recall* is a function $\mathrm{rec} : \mathcal{L}(A) \times \mathcal{M}(A) \to [0, 1]$ that quantifies the observed behavior allowed by the model. For both measures, the value of one denotes perfect conformance between the log and model. As shown in [23, 26], the entropy-based precision and recall measures satisfy all the requirements for this class of measures proposed in [23, 26–28].

Process discovery algorithms are often designed with a specific quality goal in mind. Several algorithms have *rediscoverability* as their goal: if the unknown, true process that generated the event log has specific properties, and the event log satisfies certain criteria, then the algorithm discovers the true process. For example, the α-miner has the rediscoverability property for structured workflow nets, imposing log completeness as criterion [3]. Similarly, the Inductive Miner [17] can rediscover process trees under the assumption of activity completeness, i.e., every leaf in the tree should occur at least once in the event log. Other algorithms take different approaches, e.g., to return a model that scores best on one or more conformance measures (e.g., [13, 29, 31]).

2.2 Relating Log Quality and Model Quality

Event logs used as inputs to process discovery algorithms are often assumed to be faithful representations of the true processes. Let us reflect on the consequences of this assumption. Consider Fig. 1. Assume some event log L is a faithful representation of some true process TP. In other words, L has a high model quality $\mathcal{P}^T$, measured in terms of precision and recall between L and TP. The true process TP is executed continuously, thus generating a stream of events, from which L is a snapshot [16, 28]. Therefore, L can be seen as a sample from this stream. Potentially, samples of L can be faithful representations of TP as well. Let S be a sample of L. As it is a sample, the field of statistics provides methods to assess the quality e of the sample with respect to L. And, because S is an event log itself, it can be used to discover some model M, which has quality $\mathcal{P}^S$, again measured in terms of precision and recall, but this time between S and M. Then, if S is a good representation of log L, a process discovery algorithm should construct a model with a quality that approaches $\mathcal{P}^T$.

Now, draw two samples from L, say S_1 and S_2. For S_1, model M_1 is discovered, with model quality $\mathcal{P}^{S_1}$, and for S_2 a model M_2 is discovered, with model quality $\mathcal{P}^{S_2}$. Suppose S_1 has a higher sample quality than S_2. In other words, S_1 is a better representation for L than S_2. Intuitively, the quality of M_1 should then also be closer

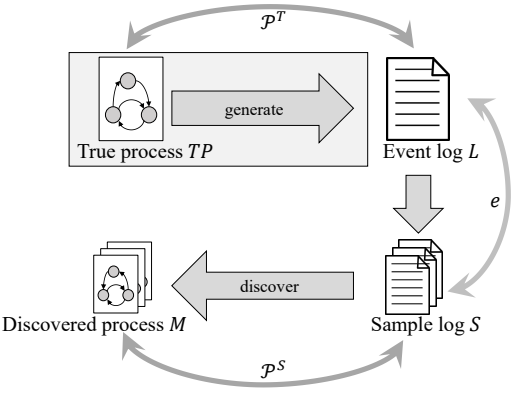

Fig. 1. A true process TP generates an event log L with unkown quality $\mathcal{P}^T$. Any sample S drawn from L has some error e. Discovering a model from S results in a model with quality $\mathcal{P}^S$.

to $\mathcal{P}^T$ than the quality of M_2. In other words, if $e(S_1) \geq e(S_2)$ then one should expect that $\mathcal{P}^{S_1} \geq \mathcal{P}^{S_2}$. Hence, it is desirable that the process discovery algorithm guarantees that better quality logs result in better quality models.

In real-life situations, the true process that generated the event log is unknown. In most process mining methods (cf., [9, 14]) the event log is prepared, and then process discovery techniques are applied to unravel a process model. An important concern that these methods do not address relates to the internal validity of process mining projects: if the process is repeated on a new observation, i.e., a new event log, to what degree do the results agree between the analyses? For this property, i.e., test-retest reliability, the guarantees of a process discovery algorithm come into play. If the different samples are of similar quality, then the constructed models should be of similar quality. However, current process discovery algorithms do not explicitly claim to provide such guarantees.

3 Sampling to Measure Log Quality

A necessary step in providing guarantees on the results of process discovery algorithms is to establish measures for log quality. We argue that any event log can be studied as a random sample of traces generated by the true process. Similar to [28], the true process can be represented as a set of traces with some trace likelihood function that assigns a probability to each trace. Consequently, any sample of an event log is again a sample of the true process, as proposed in [16]. We consider a sample log S of an event log L to be a subset of the traces observed in the event log, i.e., $S(\sigma) \leq L(\sigma)$, for all traces $\sigma \in L$, and $S(\sigma) = 0$ if $\sigma \notin L$. This allows drawing different samples from a given event log, and then comparing these samples with the event log to analyze the quality of these samples. Currently, little is known about the representativeness or quality of random samples in process mining [16,30]. In the remainder of this section, we propose random sampling techniques to be used in process mining and provide measures to analyze the quality of a sample with respect to the original event log.

3.1 Sampling Techniques

In this section, we propose three probability sampling techniques that can be used to draw a sample from an event log, where each trace in the event log has equal probability of being sampled. Consequently, samples from these techniques can be used to estimate characteristics of the event log, and, thus, of the true process.

The first technique is *simple random sampling*, where a sample is created by randomly including traces with a predetermined sampling ratio. The second technique is *stratified sampling*, where the data is divided into unique groups, called strata. For process discovery, these groups can be formed based on unique traces. Then, a simple random sample is taken from each group. In theory, this sampling technique would give more representative samples because of stratification on unique traces. However, one has to be careful when applying stratified sampling: as only a natural number of traces can be added to a sample, a trace can only be added fully or not at all. Hence, a problem occurs if a stratum contains fewer traces than there are expected to be sampled. To solve this, rounding using the half to even rule (cf. IEEE 754) can be used, which rounds halves to the nearest even integer, while still rounding other decimal numbers to the nearest integer. No literature exists on the topic of using stratified sampling in the area of process discovery [30].

An extension of stratified sampling is an approach we call *stratified squared* sampling. First, a stratified sample is drawn. Then the number of sampled traces is compared to the number of expected traces based on the sampling ratio. Due to rounding, the number of expected traces can be greater than the number of actually added traces. If this happens, the uncovered strata are sorted based on their frequency, and a trace of each of these strata is added, until the number of sampled traces matches the expected number of traces, or all strata are covered.

3.2 Towards Sample Quality Measures for Process Mining

Event logs describe the behavior of a system in terms of traces of events. As in [16], we define behavior as the directly-follows relation induced from the event log L. The directly-follows relation $>_L$ is defined on pairs of events a and b, such that $a >_L b$ iff the event log L contains a trace in which the two activities a and b occur consecutively. A first measure to compare a sample to its original event log is existential completeness, i.e., the extent to which all possible directly-follows relations are present. This results in the first sample quality measure: *coverage*, which is defined by the proportion of unique directly-follows relations present in the sample and the number of unique directly-follows relations in the event log.

Coverage does not take the occurrence frequency of behavior into account. Different methods exist to measure frequency representativeness. In statistics, error measures are used to quantify the error between the expected values and the real occurrences. We propose to adapt these error measures to quantify the error between the behavior observed in a sample, and the expected behavior from the event log based on the sampling ratio. This results in several measures for sample quality, where $\mathbf{e}$ denotes the expected behavior, and $\mathbf{s}$ denotes the sampled behavior as vectors of length n:

The Normalised Mean Absolute Error (NMAE) calculates the normalized absolute deviation (i.e. error) of the number of occurrences of each unique directly-follows relation of the sample from their respective expected frequency:

$$\text{NMAE} = \frac{\text{MAE}}{\text{avg}\,\mathbf{e}} = \frac{\sum_{i=1}^{n} |\mathbf{s}_i - \mathbf{e}_i|}{\sum_{i=1}^{n} \mathbf{e}_i} \tag{1}$$

Normalised Root Mean Square Error (NRMSE) is similar to NMAE, but uses the root of the squared values, instead of the absolute values, thus penalising large deviations more heavily:

$$\text{NRMSE} = \frac{\text{RMSE}}{\text{avg}\,\mathbf{e}} = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n} (\mathbf{s}_i - \mathbf{e}_i)^2}}{\frac{1}{n}\sum_{i=1}^{n} \mathbf{e}_i} \tag{2}$$

The Symmetric Mean Absolute Percentage Error (sMAPE) is a symmetric variation of the NMAE, expressed as a percentage error, with the advantage that the under-sampling of behavior is penalised more heavily:

$$\text{sMAPE} = \frac{1}{n} \sum_{i=1}^{n} \frac{|\mathbf{e}_i - \mathbf{s}_i|}{\mathbf{e}_i + \mathbf{s}_i} \tag{3}$$

The Symmetric Root Mean Square Percentage Error (sRMSPE) is similar to sMAPE, using the root mean square error instead of the mean absolute error, thus penalising large deviations more heavily:

$$\text{sRMSPE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left(\frac{\mathbf{e}_i - \mathbf{s}_i}{\mathbf{e}_i + \mathbf{s}_i} \right)^2} \tag{4}$$

For a detailed evaluation of the above measures, we refer the reader to [30]. These measures assess the behavioral quality of a sample with respect to the event log it is drawn from. In other words, these measures provide ways to establish the quality of the input of process discovery algorithms.

4 Designing Process Discovery Algorithms with Guarantees

As observed in a study on the quality of conformance measures [23], some process discovery algorithms have a large variability in the quality of the constructed process models, though the used measures satisfy the properties proposed in [27, 28]. In particular, given different samples of a single event log, the same algorithm sometimes provides good results on small samples, while on larger samples, the algorithm discovers worse models. On further inspection, these algorithms are state of the art, and do not perform any major "process mining crimes" [24]. In addition, they "glitter" in the benchmark study reported in [5].

We consider this observation a threat to the application of process mining, particularly for its repeatability and, hence, the reliability of its results. Suppose for a true

process several event logs are captured and analyzed, and the results do not agree, i.e., they differ largely in quality. Several explanations for this phenomenon are possible. A first explanation could be the quality of the input, i.e., the quality of the event logs differed significantly. However, as the observation highlights, another plausible – yet undesirable – explanation lies in the process discovery algorithm itself. In other words, if the process discovery algorithm does not provide any guarantees on the quality of the resulting models, it is impossible to exclude the algorithm as a root cause.

Consequently, we advocate process discovery algorithms to provide guarantees on the quality of the produced results. To this end, we propose to distinguish four stages during the introduction of a process discovery algorithm:

1. The algorithm is well designed;
2. The algorithm is validated on real-life examples;
3. The algorithm has an established relationship between the log and model quality;
4. The algorithm is effective.

Though the first two stages are basic, not all algorithms make it to the second stage, as illustrated later. Arguably, algorithms that are shown not to pass the second stage should not be used in empirical studies. The third and fourth stages are entirely novel for process discovery. Once the algorithm is shown to be applicable on real-life examples, the authors should study which guarantees their algorithm provides in a controlled setting where the true process is known. To pass the last stage, the algorithm should provide evidence that in settings where the true process is unknown, the algorithm provides the guarantees stated at stage 3.

4.1 Stage 1: The Algorithm is Well Designed

In the first stage, the developers of a process discovery algorithm should properly introduce their algorithm. For this, the developers need to provide the following:

- The class of process models the algorithm constructs;
- Evidence for meeting the quality goals of the algorithm;
- Criteria on the logs, e.g., requirements on the true process that generates the logs;
- An initial evaluation on artificial data sets.

Most process discovery algorithms satisfy the requirements of this stage. For example, the ILP-miner [31] is designed for the class of classical Petri nets with interleaving semantics. It is proven to always return a Petri net with a perfect recall score. It imposes no requirements on the input event logs and is tested on artificial logs. Also, the α-miner [3] algorithm is at least in this stage. It is designed for well-structured Workflow nets with rediscoverability as a goal. It imposes two requirements on an input event log: it should contain all directly-follows relations present in the true process, and the true process should be block-structured. A similar argument holds for the Inductive Miner [17].

4.2 Stage 2: The Algorithm is Validated

Even though an algorithm may be well designed, i.e., it passes stage 1, it is not guaranteed that it works in practice. The second stage in introducing the algorithm is, therefore, the validation of the algorithm on a collection of real-life event logs, such as used in the

Algorithm 1: Establish Relation

```
while True do
    TP ← GenerateModel(M, A);
    foreach i ∈ [1..N] do
        L ← GenerateLog(TP, T);
        P^T ← calcModelQuality(L, TP);
        foreach r ∈ ratios do
            foreach j ∈ [1..K] do
                S ← DrawSample(L, r);
                e ← calcSampleQuality(L, S);
                M ← DiscoverModel(S);
                P^S ← calcModelQuality(S, M);
```

Algorithm 2: Test Effectiveness

```
foreach L ∈ Benchmark do
    foreach r ∈ ratios do
        foreach j ∈ [1..K] do
            S ← DrawSample(L, r);
            e ← calcSampleQuality(L, S);
            M ← DiscoverModel(S);
            P^S ← calcModelQuality(S, M);
```

benchmark reported in [5]. Several algorithms fail to reach this stage. For example, the α-miner is theoretically a robust algorithm, but the requirements it imposes on the true process are too strong for application in real-life situations. Similarly, the ILP-miner is designed from a theoretical point of view and has limitations for practical use, primarily because of its guaranteed recall and runtime performance. Other algorithms, such as the Inductive Miner [17], the Declare Miner [19] and the Split Miner [4] have been applied successfully on several real-life event logs, and thus pass this stage.

4.3 Stage 3: An Established Relationship Between Log and Model Quality

Although passing stage two shows the algorithm's capabilities, this does not provide any guarantees on the quality of the algorithm's output. As a first step in establishing a relationship between the log and model quality, it needs to be shown to what degree the algorithm satisfies the guarantees as sketched in Fig. 1. In other words, the designers need to show that if an event log is a faithful representation of a true process, as per measure $\mathcal{P}^T$, then the algorithm should satisfy properties similar to those listed below:

P1. For a sample log S that approaches the perfect quality, the quality $\mathcal{P}^S$ of the discovered model from S approaches $\mathcal{P}^T$;

P2. For two samples S_1 and S_2, if sample S_1 has a higher quality than S_2, then the model quality $\mathcal{P}^{S_1}$ is higher than $\mathcal{P}^{S_2}$.

Algorithm designers can choose different strategies to provide evidence for these properties. The most potent form of evidence is a formal proof that the algorithm satisfies these properties for specific instantiations of log and model quality measures. In that way, a relationship between an input log quality and the resulting model quality can be established. We also encourage algorithm designers to define algorithm-specific log quality measures. If a formal proof is not feasible, instead, statistical evidence of these properties can be provided. For this, we propose a controlled experiment as outlined in Algorithm 1. Such a controlled experiment follows the approach shown in Fig. 1. It requires the algorithm designers to have a model generator for the class of true processes the algorithm accepts. The algorithm then generates repeatedly for a true process one or more event logs, and for each event log a set of samples.

We propose to use statistical tests to evaluate the two properties. Property P1 needs an analysis of the relation between the expected $\mathcal{P}^T$ and the observed $\mathcal{P}^S$. For property

Table 1. Results of the controlled experiment. The last 10 columns show the Spearman rank correlation between the error measures, and precision and recall. All bold values are statistically significant ($p < 0.001$).

	True Process		Precision					Recall				
Model	prec.	recall	Cov.	sMAPE	sRMSPE	NRMSE	NMAE	Cov.	sMAPE	sRMSPE	NRMSE	NMAE
1	0.538	1.000	**0.658**	**-0.988**	**-0.986**	**-0.988**	**-0.989**	**0.338**	**-0.356**	**-0.354**	**-0.354**	**-0.356**
2	0.797	1.000	**0.470**	**-0.986**	**-0.985**	**-0.901**	**-0.954**	0.154	-0.051	-0.052	0.012	-0.004
3	0.935	1.000	**0.781**	**-0.990**	**-0.989**	**-0.975**	**-0.984**	**0.637**	**-0.406**	**-0.417**	**-0.410**	**-0.412**
4	0.953	1.000	**0.705**	**-0.991**	**-0.992**	**-0.984**	**-0.987**	-0.103	0.105	0.108	0.081	0.090
5	0.988	1.000	**0.540**	**-0.983**	**-0.981**	**-0.980**	**-0.986**	**0.437**	-0.201	-0.206	-0.207	-0.201
6	0.871	1.000	**0.532**	**-0.934**	**-0.938**	**-0.917**	**-0.926**	**-0.529**	**0.973**	**0.962**	**0.963**	**0.968**
7	0.943	1.000	**0.511**	**-0.991**	**-0.989**	**-0.986**	**-0.989**	**0.456**	-0.242	-0.240	-0.228	-0.231
8	0.616	1.000	**0.773**	**-0.992**	**-0.991**	**-0.989**	**-0.990**	0.114	-0.148	-0.154	-0.156	-0.157
9	0.710	1.000	**0.519**	**-0.981**	**-0.978**	**-0.970**	**-0.973**	**0.518**	**-0.327**	**-0.330**	**-0.340**	**-0.341**
10	0.883	1.000	**0.703**	**-0.982**	**-0.982**	**-0.977**	**-0.976**	0.116	-0.022	-0.027	-0.016	-0.023

P2, the Spearman rank correlation can be used to test whether there is a strong correlation between the sample quality and the model quality. If this is the case, then statistical evidence has been provided for the relationship between log and model quality.

Example Evaluation. As an example, the controlled experiment has been implemented in ProM [3] for the Inductive Miner [17]. To calculate precision and recall, an implementation of exact matching entropy-based measures in Entropia is used [21]. For each true process, a single event log with 5,000 traces has been generated. The event logs were 10 times sampled for 12 sampling ratios: 0.01, 0.02, 0.05, and 0.1 up to 0.9.

The results are shown in Tbl. 1 and Fig. 2. From Fig. 2 we conclude that property P1 holds for precision and recall. For each model that describes the true process, the Spearman rank correlation is calculated between each of the log quality measures and precision, and similarly for recall. As for the measures sMAPE, sRMSPE, NRMSE, and NMAE, 0 is the best quality, a negative correlation indicates the required guarantee that samples of higher quality result in better discovered models, whereas for coverage, a positive correlation indicates this result. As can be seen in the table, the experiment

[3] The source code is available on: https://github.com/ArchitectureMining/SamplingFramework

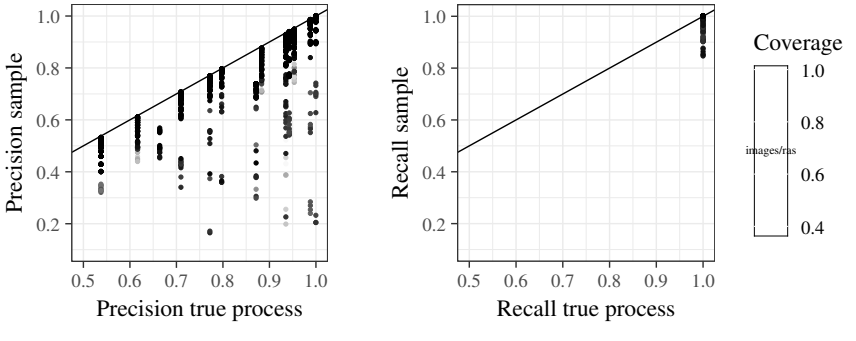

Fig. 2. Relation between the quality of the true process and the quality of the discovered models, for precision (left) and recall (right). Darker points represent a higher coverage.

generates mixed results. Though property P2 holds for precision, it is not satisfied for recall. Hence, we can conclude that the Inductive Miner satisfies the two properties for precision, but fails to do so for recall on the second property.

4.4 Stage 4: The Algorithm is Effective

An established relationship between log and model quality, the essence of stage 3, does not guarantee the algorithm to be effective in real-life situations. The main caveat in the controlled environment of the previous stage is that the true process is known. Each event log is generated from the known true processes. In real-life situations, the true process is unknown, and, hence, may invalidate assumptions of the discovery algorithm. For example, the Inductive Miner assumes event logs to be generated from process trees. However, no criteria are given to test whether an event log is generated by a process tree, nor does the algorithm provide any details on the model quality if the assumption is invalid.

In this stage, the algorithm designer has to validate how effective the algorithm is in real-life situations. One way to obtain insights into the effectiveness of the algorithm is to apply sampling on a benchmark. This benchmark can be a set of well-known real-life event logs as used in [5], or can be generated automatically, if the designers ensure that the class of generated models is larger than the class of true processes studied in the previous stage. The algorithm designers need to analyze property P2 in the absence of a true process. In other words, even if the true process is unknown, event logs of better quality should return better quality models. This may result in an experiment as outlined in Algorithm 2.

The analysis of property P2 in the absence of a true process can have two possible outcomes. Either it is shown that the algorithm has the desired property, or, if this is not possible, the algorithm should be further improved, or provide additional log quality measures, that guarantee that an event log satisfies the assumptions of the process discovery algorithm.

Example Evaluation. As an example of the analysis in stage 4, we conducted the proposed experiment on the Inductive Miner [17]. Two real-life event logs have been selected, the Road Traffic Fine event log [12] and the Sepsis event log [20]. The Road Traffic Fine log has in total 150,370 traces and 561,470 events. There are 231 unique traces and 11 unique event types. The Sepsis log consists of 1,049 traces, of which 845 are unique, and 15,190 events with 16 unique event types. Sampling was done at the same sampling ratios as before: 0.01, 0.02, 0.05, and 0.1 up to 0.9. For each ratio, ten samples were drawn.

The sample quality measures for the Road Traffic Fine log are shown on the left in Fig. 3. As the plot shows, the larger the sampling ratio, and thus the log size, the better the quality is (error measures: $\rho < -0.9$, $p < 0.001$, coverage: $\rho = 0.96$, $p < 0.001$). Sample size and the conformance measure on precision (Fig. 4) show a moderate positive correlation ($\rho = 0.56$, $p < 0.001$), while there is no correlation between sampling ratio and recall ($\rho = 0.03$, $p = 0.72$). Analyzing the quality measures with the conformance measures shows a different story. In Fig. 4, the coverage is plotted against the precision, indicating there is no correlation between coverage and precision.

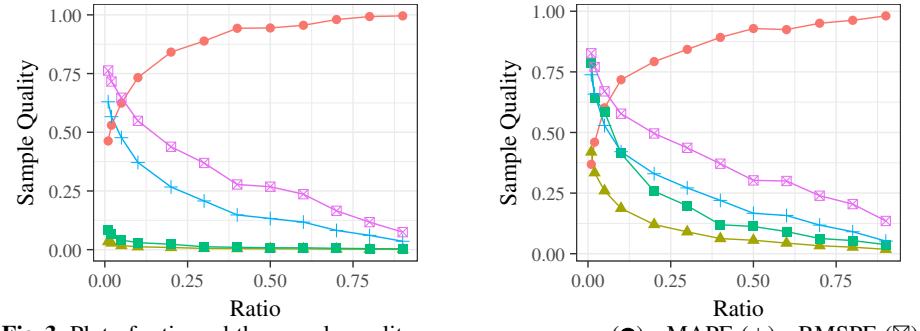

Fig. 3. Plot of ratio and the sample quality measures coverage (●), sMAPE (+), sRMSPE (☒), NRMSE (■) and NMAE (▲) for the Road Traffic Fine log (left) and the Sepsis log (right).

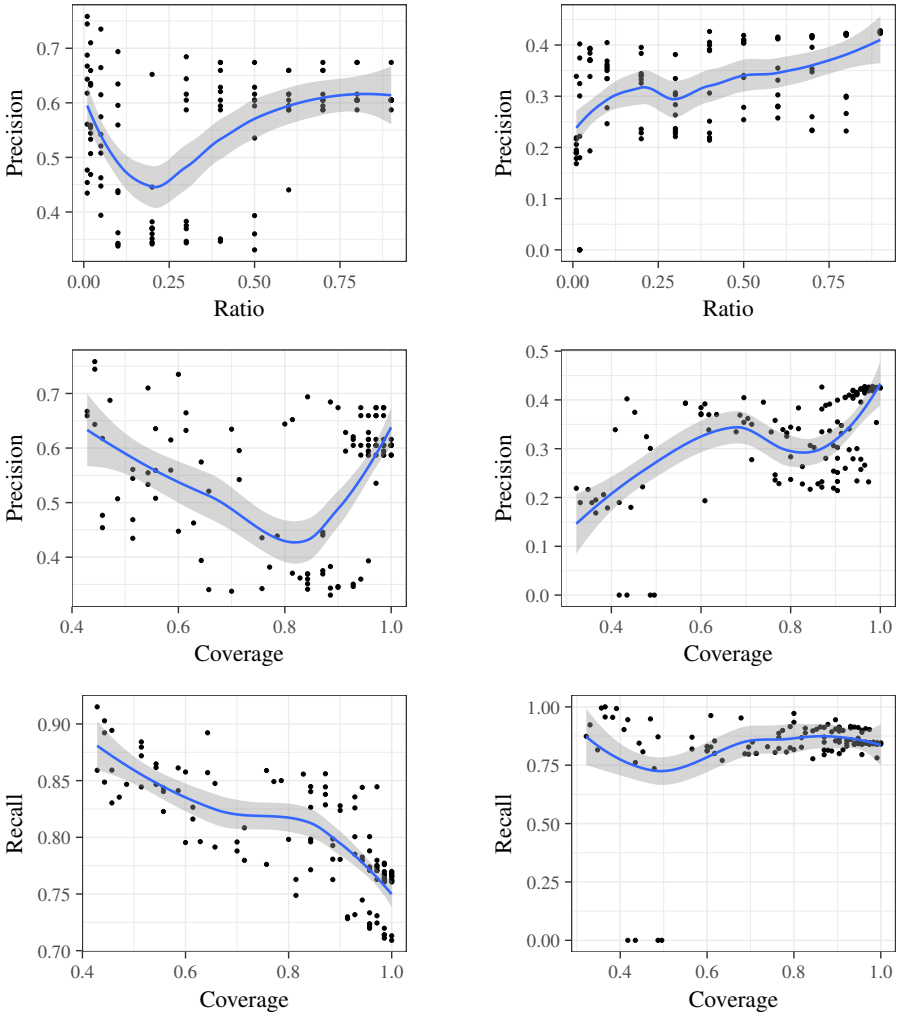

Fig. 4. Plots of ratio and precision, and coverage with precision and recall for the Road Traffic Fine log (left) and the Sepsis Log (right).

Further analysis revealed no correlations between the sample quality measures and precision (sMAPE: $\rho = -0.19$, $p = 0.03$, sRMSPE: $\rho = -0.18$, $p = 0.051$, NRMSE: $\rho = -0.21$, $p = 0.02$, NMAE: $\rho = -0.20$, $p = 0.03$, coverage: $\rho = 0.17$, $p = 0.06$). The correlations found for recall show that samples of worse quality result in better models (sMAPE: $\rho = 0.80$, $p < 0.001$, sRMSPE: $\rho = 0.79$, $p < 0.001$, NRMSE: $\rho = 0.77$, $p < 0.001$, NMAE: $\rho = 0.78$, $p < 0.001$, coverage: $\rho = -0.79$, $p < 0.001$).

For the Sepsis log, similar results are found. As indicated by the plots at the right hand side of Fig. 3, a correlation is found between the sampling ratio and the log quality measures (for all error measures: $\rho < -0.9$, $p < 0.001$, coverage: $\rho = 0.59$, $p < 0.001$). The larger the sampling ratio, the higher the precision is ($\rho = 0.57$, $p < 0.001$), but no correlation was found between sampling ratio and recall ($\rho = 0.03$, $p = 0.72$). A moderate negative correlation was found between the log quality measures and precision (for the error measures: $-0.60 < \rho < -0.50$, $p < 0.001$, coverage: $\rho = 0.59$, $p < 0.001$), while the log quality measures did not show any correlation with recall (for all measures: $-0.04 < \rho < 0.02$, $p > 0.70$).

As the results suggest, there is no clear relation between log and model quality. Hence, it is with the current measures not possible to conclude that the Inductive Miner is guaranteed to be effective in real-life situations. As a next step, new log quality measures should be developed that do establish the required relationship between log and model quality. The process can then be repeated until sufficient guarantees can be provided on the effectiveness of the algorithm.

5 Related Work

The statistical approach we propose to establish a relation between log and model quality relates to event data quality in general, builds upon established properties of conformance measures, and requires sampling techniques on event logs. This section reviews literature on these topics, and shows how our approach relates to them.

Measuring log quality. As the process mining manifesto articulates, process mining treats data as first-class citizens [2], and defines four data qualities, of which *completeness* is studied mostly. For example, [8] identifies four categories of process characteristics and 27 classes of event log quality issues. Most studies on event log quality focus on the incompleteness of the data. Examples include not having enough information recorded in the event log (e.g., missing cases or events) [1, 8], not having recorded enough behavior in the event log [15], or the traces not being representative of the process [15], and noise. Different notions of noise are studied, such as infrequent behavior that is either incorrect or rare [13]. However, event logs are studied in isolation in these studies. Instead, we argue to assess the quality of event logs relative to other event logs, using statistical techniques based on sampling.

Properties of conformance measures. The process mining community has recently initiated a discussion on which formal properties should "good" conformance measures satisfy. In [27], the authors proposed five properties for precision measures. For instance, one property states that for two process models that describe all the traces in the

log, a less permissive model should not be qualified as less precise. By demonstrating that a measure fulfills such properties, one establishes its usefulness. In [23], the authors strengthened the properties from [27]. For example, according to these properties, the less permissive model from the example above should be classified as more precise. In [28], the precision properties from [27] were refined, and further desired properties for recall and generalization measures were introduced, resulting in 21 conformance propositions. Finally, in [22], properties for precision and recall measures that account for the partial matching of traces, i.e., traces that are not the same but share some subsequences of activities, were introduced. The precision and recall measures used in our evaluations satisfy all the introduced desired properties for the corresponding measures [23, 26–28].

Sampling in process mining. Sampling has been studied before in process mining, but never as a systematic approach to evaluate process discovery techniques. A first set of measures for the representativeness of samples have been proposed in [16]. Their results show the need for a systematic approach as proposed in this paper.

In [7], a sampling technique specific for the Heuristics Miner is described, claiming that only 3% of the original log is sufficient to discover 95% of the dependency relations. However, a proper evaluation of this claim has not been provided, nor are the results generalizable to other process discovery techniques.

A statistical framework based on *information saturation* is proposed in [6]. Their approach differs from the probability sampling techniques we propose. Instead of generating samples that estimate the event log, their approach focuses on creating a sufficiently small sample that contains as much information from the event log as possible. Consequently, this approach cannot be used to measure sample quality with respect to the event log.

Several biased sampling techniques are described in [25]. These techniques have been evaluated on six real-life event logs and three discovery techniques. The evaluation showed that sampling sometimes improves the F-measure for some of the models. A similar result on the F-measure was obtained in [18]. Their study applied the Google PageRank algorithm on event logs to create a representative sample, which reduced the execution time of the Inductive Miner by half without decreasing the F-measure. As the F-measure harmonizes precision and recall, and no analysis was performed on the reasons behind the improvements, it is unclear how sampling influenced the process discovery results of both studies. Instead of using sampling to improve the quality of the output, we propose to use probability sampling to analyze the input of algorithms, and to establish a relationship between log and model quality. This relationship then allows one to explore why some samples give better models than other samples.

6 Conclusion

This paper identifies the need for process discovery algorithms with guarantees that characterize the dependency between the quality of input event logs and the quality of the process models constructed from these event logs. In particular, we argue that process discovery algorithms should produce better models from better input logs. Currently, process discovery algorithms have never provided such guarantees, since, so far,

we, as a community, lacked a theoretical foundation to establish such a relationship. In this paper, for the first time, measures for the statistical sample quality for ranking the quality of event logs are proposed. We recommend using grounded conformance checking measures for assessing the quality of the discovered models. Combining log quality measures with conformance measures provides a framework to formally define properties that express the desired guarantee that better event logs result in better models. These properties can be instantiated with various measures for quality of event logs and process models and be less or more pronounced, for example, imposing a strictly increasing or non-decreasing relation, or requiring a statistical association of a certain degree between the qualities of the corresponding logs and models. To overcome this problem, we propose four stages in the design of an algorithm. Each design comes with additional properties and obligations to establish effective algorithms with guarantees.

We invite the process mining community to further contribute to the discussion of desired qualities for process discovery algorithms to ensure that state-of-the-art algorithms fulfill them, and in this way, advance the field of process discovery as well as the design and evaluation of such algorithms.

Acknowledgments. Artem Polyvyanyy was in part supported by the Australian Research Council project DP180102839.

References

1. W. M. P. van der Aalst. *Process Mining—Data Science in Action, Second Edition*. Springer Berlin Heidelberg, 2016.
2. W. M. P. van der Aalst et al. Process mining manifesto. In *BPM Workshops*, volume 99 of *LNBIP*, pages 169–194. Springer, 2011.
3. W. M. P. van der Aalst, A. J. M. M. Weijters, and L. Maruster. Workflow Mining: Discovering Process Models from Event Logs. *Knowledge & Data Engineering*, 16(9):1128–1142, 2004.
4. A. Augusto, R. Conforti, M. Dumas, and M. La Rosa. Split miner: Discovering accurate and simple business process models from event logs. In *ICDM 2017*, pages 1–10. IEEE, 2017.
5. 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. *IEEE Trans. Knowl. Data Eng.*, 31(4):686–705, 2019.
6. M. Bauer, A. Senderovich, A. Gal, L. Grunske, and M. Weidlich. How much event data is enough? a statistical framework for process discovery. In *CAiSE 2018*, volume 10816 of *LNCS*, pages 239–256. Springer, 2018.
7. A. Berti. Statistical sampling in process mining discovery. In *eKNOW 2017*, pages 41–43. IARIA, 2017.
8. J. C. Bose, R. S. Mans, and W. M. P. van der Aalst. Wanna improve process mining results? In *CIDM 2013*, pages 127–134. IEEE, 2013.
9. M. Bozkaya, J. M. A. M. Gabriels, and J. M. E. M. van der Werf. Process diagnostics : a method based on process mining. In *eKNOW 2009*, pages 22–27. IEEE, 2009.
10. J. C. A. M. Buijs, B. F. van Dongen, and W. M. P. van der Aalst. Quality dimensions in process discovery: The importance of fitness, precision, generalization and simplicity. *International Journal of Cooperative Information Systems*, 23(1), 2014.
11. Th. H. Cormen, Ch. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press Ltd, 2009.
12. M. de Leoni and F. Mannhardt. Road Traffic Fine Management Process, 2 2015. doi:10.4121/uuid:270fd440-1057-4fb9-89a9-b699b47990f5.

13. A. K. A. de Medeiros, A. J. M. M. Weijters, and W. M. P. van der Aalst. Genetic process mining: an experimental evaluation. *Data Min. Knowl. Discov.*, 14(2):245–304, 2007.
14. M. L. van Eck, X. Lu, S. J. J. Leemans, and W. M. P van der Aalst. PM2: A process mining project methodology. In *CAiSE 2015*, volume 9097 of *LNCS*. Springer, 2015.
15. C. Günther. *Process mining in flexible environments*. PhD thesis, Eindhoven University of Technology, 2009.
16. B. Knols and J. M. E. M. van der Werf. Measuring the behavioral quality of log sampling. In *ICPM 2019*, pages 97–104. IEEE, 2019.
17. S. J. J. Leemans, D. Fahland, and W. M. P. van der Aalst. Scalable process discovery with guarantees. In *EMMSAD 2015*, volume 214 of *LNBIP*, pages 85–101. Springer, 2015.
18. C. Liu, Y. Pei, Q. Zeng, and H. Duan. Logrank: An approach to sample business process event log for efficient discovery. In *Knowledge Science, Engineering and Management*, volume 11061 of *LNCS*, pages 415–425. Springer, 2018.
19. F. M. Maggi, J. C. Bose, and W. M. P. van der Aalst. Efficient discovery of understandable declarative process models from event logs. In *CAiSE 2012*, volume 7328 of *LNCS*, pages 270–285. Springer, 2012.
20. F. Mannhardt. Sepsis Cases - Event Log, 12 2016. doi:10.4121/uuid:915d2bfb-7e84-49ad-a286-dc35f063a460.
21. A. Polyvyanyy, H. Alkhammash, C. Di Ciccio, L. García-Bañuelos, A. A. Kalenkova, S. J. J. Leemans, J. Mendling, A. Moffat, and M. Weidlich. Entropia: A family of entropy-based conformance checking measures for process mining. In *ICPM Doctoral Consortium and Tool Demonstration*, volume 2703 of *CEUR*, pages 39–42. CEUR-WS.org, 2020.
22. A. Polyvyanyy and A. A. Kalenkova. Monotone conformance checking for partially matching designed and observed processes. In *ICPM 2019*, pages 81–88, 2019.
23. A. Polyvyanyy, A. Solti, M. Weidlich, C. Di Ciccio, and J. Mendling. Monotone precision and recall measures for comparing executions and specifications of dynamic systems. *ACM Trans. Softw. Eng. Methodol.*, 29(3):17:1–17:41, 2020.
24. J. Rehse and P. Fettke. Process mining crimes - A threat to the validity of process discovery evaluations. In *BPM Forum 2018*, volume 329 of *LNBIP*, pages 3–19. Springer, 2018.
25. M. Fani Sani, S. J. van Zelst, and W. M. P. van der Aalst. Improving the performance of process discovery algorithms by instance selection. *Comput. Sci. Inf. Syst.*, 17(3):927–958, 2020.
26. A. F. Syring, N. Tax, and W. M. P. van der Aalst. Evaluating conformance measures in process mining using conformance propositions. *ToPNOC*, pages 192–221, 2019.
27. N. Tax, X. Lu, N. Sidorova, D. Fahland, and W. M. P. van der Aalst. The imprecisions of precision measures in process mining. *Inf. Process. Lett.*, 135:1–8, 2018.
28. Wil M. P. van der Aalst. Relating process models and event logs—21 conformance propositions. In *ATAED*, volume 2115 of *CEUR Workshop Proceedings*, pages 56–74. CEUR-WS.org, 2018.
29. A. J. M. M. Weijters and J. T. S. Ribeiro. Flexible heuristics miner (FHM). In *CIDM 2011*, pages 310–317. IEEE, 2011.
30. B. R. van Wensveen. Estimation and analysis of the quality of event log samples for process discovery. Master's thesis, Utrecht University, 2020. https://dspace.library.uu.nl/handle/1874/400143.
31. J. M. E. M. van der Werf, B. F. van Dongen, C. A. J. Hurkens, and A. Serebrenik. Process discovery using integer linear programming. *Fundamenta Informaticae*, 94(3-4):387 – 412, 2009.