Measuring the Quality of Models with Respect to the Underlying System: An Empirical Study

Peer-reviewed author version

JANSSENSWILLEN, Gert; JOUCK, Toon; CREEMERS, Mathijs & DEPAIRE, Benoit (2016) Measuring the Quality of Models with Respect to the Underlying System: An Empirical Study. In: La Rosa, Marcello; Loos, Peter; Pastor, Oscar (Ed.). Business Process Management: 14th International Conference, BPM 2016, Rio de Janeiro, Brazil, September 18-22, 2016. Proceedings, Springer,p. 73-89.

DOI: 10.1007/978-3-319-45348-4_5
Handle: http://hdl.handle.net/1942/22666

Measuring the quality of models with respect to the underlying system: An empirical study

Gert Janssenswillen[1,2], Toon Jouck[1], Mathijs Creemers[1] and Benoît Depaire[1]

[1] Hasselt University, Agoralaan Bldg D, 3590 Diepenbeek, Belgium
[2] Research Foundation Flanders (FWO), Egmontstraat 5, 1060 Brussels, Belgium
{gert.janssenswillen,toon.jouck,mathijs.creemers,benoit.depaire}@uhasselt.be

Abstract Fitness and precision are two widely studied criteria to determine the quality of a discovered process model. These metrics measure how well a model represents the log from which it is learned. However, often the goal of discovery is not to represent the log, but the underlying system. This paper discusses the need to explicitly distinguish between a log and system perspective when interpreting the fitness and precision of a model. An empirical analysis was conducted to investigate whether the existing log-based fitness and precision measures are good estimators for system-based metrics. The analysis reveals that incompleteness and noisiness of event logs significantly impact fitness and precision measures. This makes them biased estimators of a model's ability to represent the true underlying process.

Keywords: Conformance Checking, Evaluation Metrics, Process Model Quality

1 Introduction

Due to the enormous growth of event data during the last decades, organizations are dealing with the challenge of extracting useful knowledge from it, and exploiting it to gain competitive advantages. Process mining provides ways to reach this goal, by getting a better understanding of business processes and improving them [1]. The origin of process mining dates back to the end of the previous century [5,12], and focused on discovering the process control-flow from event logs which contain recorded process behaviour. While the domain has grown much broader, control flow discovery is the most mature research track within process mining. For an overview of existing process discovery algorithms, witness [13]. In order to quantify the quality of a discovered process model, different quality dimensions have been defined [21], i.e. fitness, precision, generalization and simplicity. For each of the dimensions, several metrics have been developed and implemented, of which an overview can be found in [8].

Existing fitness and precision metrics typically measure the quality of a model with respect to the event log it was learned from. They thereby do not take into account that the event log is a limited sample of the real unknown process and

possibly contains measurement errors. Since the underlying process is not known in real-life settings, the quality of a discovered process model as a representation of the underlying process cannot be determined directly. So far, little empirical research has been done to analyse whether the existing fitness and precision measures can be trusted as estimators of the behavioural similarity between the discovered model and the underlying system.

In this paper, the need to explicitly distinguish between a log and system perspective when interpreting the fitness and precision of a model, is highlighted. Experiments are conducted to examine whether the quality of a model with respect to the event log can be used as an unbiased estimator for the quality of the model with respect to the underlying process. Both the metrics their ability to estimated the quality of models unbiasedly, as their ability to unbiasedly rank a given collection of models from worst to best will be investigated.

The next section introduces the evaluation framework. Section 3 describes the experimental set-up. The results of this experiment are discussed in Section 4. Section 5 discusses the role of generalization and its link with the proposed framework. Finally, Section 6 provides an overview or related work and Section 7 concludes the paper.

2 Different perspectives in measuring model quality

The four classical quality dimensions are fitness, precision, generalization and simplicity [1]. Fitness, precision and generalization can be visualized using a Venn diagram[9], as is shown in Fig. 1.[3] In this figure, M, L and S refer to the *process behaviour* which belongs to the model, event log and system, respectively. It therefore abstracts from the representational language of the model. According to the author [9], the system S refers to the context of the process, e.g., the organization, rules, economy, etc.

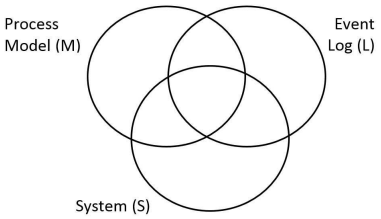

Figure 1. Venn diagram representing the behaviour in the process model (M), event log (L) and system (S) [9].

In this paper, a slightly more tangible definition of *system* will be used. It will refer to the behaviour which is *real*, i.e. the underlying process. This

[3] In this paper, the simplicity dimension will not be taken into account, as it is not directly related to the behaviour of the discovered model.

process, generally unknown, defines the actual way in which work can be done. Note that the system is broader than only a prescriptive model used for the configuration of an information system, but can also include certain unwritten rules or customs. Everything which appears in the event log but is not part of the system is regarded as noise. Examples are measurement errors resulting from system outages. On the other hand, real though infrequent behaviour will not be perceived as noise in this paper.

Figure 1 points out that a discrepancy exists between the system and the event log. Process discovery tasks can thus be conducted using different objectives. Firstly, business users might be interested in the relation between the discovered model and the event log. In such a case, the objective will be to find a model which perfectly mimics the behaviour in the event log. Secondly, one might be interested in the relationship between the discovered model and the true underlying system. To understand the way work is actually be done, the objective will be to learn a model that exactly represents the system behaviour.

Following the well-known definitions of fitness and precision [10], it is evident that both dimensions try to tell something about the relationship between event log and model. Generalization, being defined in [9] as *the likelihood of previously unseen but allowed behaviour being supported by the process model*, appears to aim at assessing the relationship between the model and the system. However, the effectiveness of existing measures in achieving these goals remains unexplored.

In the remainder of this section, we articulate four alternative quality dimensions based on Fig. 1 and the work in [9]. Two of these dimensions measure the *distance* between a model and an event log, and correspond to the classical dimensions of fitness and precision. The other two quantify the distance between a model and a system. The four dimensions will be defined conceptually in terms of L, M and S, after some preliminary notations have been introduced.

2.1 Preliminaries

Definition 1 (Activity sequences). *We define $\mathcal{A}$ as the activity alphabet. $\mathcal{A}^*$ is the set of all finite sequences over $\mathcal{A}$, representing the universe of traces. A trace $\sigma_j \in \mathcal{A}^*$ is a finite sequence of activities.*

Definition 2 (Event log, model, system). *An event log L is a multiset of traces, i.e. $L \in \mathbb{B}(\mathcal{A}^*)$, where $\mathbb{B}(\mathcal{A}^*)$ is the set of all mutlisets of $\mathcal{A}$.*

Definition 3 (Model, system). *A model M and a system S are subsets of the universe of traces, i.e. $M \in \mathbb{P}(\mathcal{A}^*)$ and $S \in \mathbb{P}(\mathcal{A}^*)$, where $\mathbb{P}(\mathcal{A}^*)$ is the power set of $\mathcal{A}^*$. $\mathbf{M}$ and $\mathbf{S}$ represent the domain of all possible models and systems, respectively, whereby $\mathbf{M} = \mathbf{S} = \mathbb{P}(\mathcal{A}^*)$. $\mathbf{L}$ represents the domain of all possible logs, whereby $\mathbf{L} = \mathbb{B}(\mathcal{A}^*)$.*

2.2 Model-log distance

The fit between an event log and a process model is monitored by two ratios [9], corresponding to the known concepts of fitness and precision. Given event log L, the log-fitness and log-precision of a model M can be defined as follows.

Definition 4 (Log-fitness). *Log-fitness is a function* $F : \mathbf{M} \times \mathbf{L} \rightarrow [0, 1]$*, which quantifies how much of the behaviour in the event log is captured by the model. This can be defined as [9]:*

$$F(M, L) = \frac{|L \cap M|}{|L|} \tag{1}$$

Definition 5 (Log-precision). *Log-precision is a function* $P : \mathbf{M} \times \mathbf{L} \rightarrow [0, 1]$*, which quantifies how much of the behaviour in the model was observed in the event log. This can be defined as [9]:*

$$P(M, L) = \frac{|L \cap M|}{|M|} \tag{2}$$

Only when both log-fitness and log-precision are equal to 1, then $L = M$, i.e. the event log and the model represent exactly the same behaviour. These metrics are orthogonal to each other, making it possible to construct models which score poorly on one criterion and excellent on the other. Acting as complementary forces, maximizing log-fitness and log-precision simultaneously maximizes the *fit* between the model and the event log.

2.3 Model-system distance

By drawing the analogy, it is evident that two similar dimensions are needed to quantify the match between the model and the system. Firstly, there is a need for a metric that ensures the selection of models that contain all possible real behaviour. Secondly, a metric that favors the selection of models that only contain real behaviour is needed. Therefore, given the system S, the system-fitness and system-precision of a model M can be defined as:

Definition 6 (System-fitness). *System-fitness is a function* $F : \mathbf{M} \times \mathbf{S} \rightarrow [0, 1]$*, which quantifies how much of the behaviour in the system is captured by the model. This can be defined as [9]:*

$$F(M, S) = \frac{|S \cap M|}{|S|} \tag{3}$$

Definition 7 (System-precision). *System-precision is a function* $P : \mathbf{M} \times \mathbf{S} \rightarrow [0, 1]$*, which quantifies how much of the behaviour in the model is part of the system. This can be defined as [9]:*

$$P(M, S) = \frac{|S \cap M|}{|M|} \tag{4}$$

When event logs are incomplete and contain noise, log-based and system-based metrics will diverge. Depending on the goal, business users should then direct their attention to one pair of metrics. Note that the above formulas are rather coarse-grained. While in reality more fine-grained measures are preferred, these formulas suffice to distinguish the different concepts.

3 Experimental analysis

3.1 Goal of the experiments

The goal of the experiments conducted in this paper is twofold. Firstly, the goal is to analyse whether existing metrics are unbiased estimators of the true system fitness and precision. Secondly, the goal is to analyse whether the ranking of a set of models, based on existing metrics, also represents the true ranking in representing the underlying system.

Note that both abilities - estimation and ranking - have different impacts: when the quality of models with respect to the system is consistently overestimated, the ranking of the models will remain valid. However, when the biases are highly variable among models, also the ranking of models will be perturbed.

Unbiased estimation For the first goal of the experiment the quality of a set of models was measured both with respect to the event log it was learned from and the underlying system that generated the event log. The fitness value obtained from replaying event log L onto model M is represented as $F(M, L)$ for any fitness-measure. Conversely the fitness value obtained when comparing the system S with model M is represented as $F(M, S)$. Equivalently, we can compute $P(M, L)$ and $P(M, S)$ for any precision metric. Note that $F(M, L)$ and $P(M, L)$ represent log-fitness and log-precision, as defined in Equation 1 and 2, respectively, while $F(M, S)$ and $P(M, S)$ represent system-fitness and system-precision as expressed in Equation 3 and 4, respectively.

To investigate whether $F(M, L)$ and $P(M, L)$ are unbaised estimates of $F(M, S)$ and $P(M, S)$, respectively, we define the difference between these values for both fitness and precision as follows.

$$\Delta F(M, L, S) = F(M, L) - F(M, S) \tag{5}$$

$$\Delta P(M, L, S) = P(M, L) - P(M, S) \tag{6}$$

For example, $F^{ab}(M, L)$, the log-fitness as measured by the Alignment Based Fitness metric will be an unbiased estimator of $F^{ab}(M, S)$, if $E[\Delta F^{ab}] = 0$. When this value is positive, $F^{ab}(M, L)$ is said to overestimate $F^{ab}(M, S)$. Both ΔF and ΔP will be analysed for logs with and without noise, and with varying levels of completeness.

Unbiased ranking In order to examine whether the ranking which log-based metrics define on a set of models represent the true ranking of these models with respect to the underlying system, a second analysis will be done. Even when existing metrics are biased estimators, if they still rank different models accurately, they can still be used to compare the quality of models. In order to investigate this, a limited set of discovered models will be compared with a collection of event logs generated by the same system. These event logs will have different levels of completeness and noise. The actual set up of the experiments will be discussed in the following paragraph.

Table 1. Experimental set up.

Characteristic	Value
Number of systems	10
Completeness Levels	100%, 75%, 50%, 25%
Noise levels	0%, 5%, 10%, 15%
Number of event logs for each combination	5
Discovery algorithms	Heuristics[24]
	Inductive[18]
	ILP[10]
Fitness Metrics	Alignment Based Fitness[3](ab)
	Negative Event Recall[7](ne)
	Token-Based Fitness[20](tb)
Precision Metrics	Alignment Based Precision[3](ab)
	Best Align Etc Precision[4](ba)
	Negative Event Precision[7](ne)
	One Align Etc Precision[4](oa)
Generalization Metrics	Alignment Based Generalization[3](ab)
	Negative Event Generalization[7](ne)
Total number of event logs	800 logs
Total number of models	2400 models

3.2 Set up

The set up of the experiments was based on the framework for comparing process mining algorithms in [22]. The design is explained in more detail below. Table 1 shows an overview of the key-characteristics of the experiment, including the overall scale of the experiment.

1. Generate 10 systems, which will act as ground truth process models.
2. Estimate the number of different traces which can be generated by the systems, in order to target the completeness of the logs to be simulated.
3. Simulate the enactment of each system to produce artificial event logs with different levels of noise and completeness. Furthermore, simulate a ground truth event log for each system.
4. For each log, mine a set of process models using discovery algorithms.
5. Compute the quality of the models using the selected metrics
   a. For each model, compute process quality metrics both in relation to the log it was discovered from, and in relation to the ground truth event log.
   b. For a set of randomly selected models for each system, compute the quality metrics in relation to all the event logs generated by that system.

Generation of systems In total 10 random models were artificially generated, in order to be used as systems, using the methodology in [17]. The generated systems were process trees. Each system has been generated according to its own characteristics, i.e. the expected number of activities, the different types of process operators (sequence, choice, parallel, etc.), the occurrence of silent activities and the occurrence of duplicate tasks.

Determine number of paths In order to target the completeness of event logs during log simulation, the number of unique activity sequences in each model was computed using the algorithm in [16]. In order to cope with loops, a maximum number of iterations was taken into account for this calculation. This can be justified by adapting a so-called *fairness assumption*, which states that a task of a process cannot be postponed indefinitely. The assumption therefore rules out infinite behaviours that are considered unrealistic [6].

Simulation of event logs Each of the artificial systems was used to simulate event logs with different levels of completeness and noise. Both the completeness and noise level of the logs were controlled explicitly during the simulation. The amount of noise as well as the amount of completeness has been defined at the level of activity sequences. Four different levels were defined for each characteristic, resulting in 16 different types of logs. For each type, 5 different logs were simulated, amounting to a total of 80 logs per system. Next to these, a ground truth event log, with perfect completeness and without noise is created for each system.

The levels of completeness are 100%, 75%, 50% and 25%. A log of 100% completeness is obtained by simulating the system until the simulated event log contains the same number of unique paths as calculated in the previous step, say n. A log of 75% is obtained by simulating the system until $0.75n$ different paths have been seen, etc.

The levels of noise have been defined at 0%, 5%, 10% and 15%. A log with 5% noise is created by taking $0.05/0.95 = 5.26\%$ of the traces of a noise-free log. To this subset of traces, different types of noise are added: missing head, missing body, missing tail or missing activity [19]. These noisy traces are then added to the original event log. The resulting event log consequently has $n(1 + 0.0526)$ traces, of which $0.0526n$ are noisy. The noise level will thus be 5%.

Model discovery For each log, several process models are discovered using process discovery algorithms. ProM 6.5 was used for the discovery of the process models. Default values were used for all parameters.

Conformance checking Regarding the first goal of the experiment, the quality of each of the discovered models was examined both with respect to the event log from which the model was discovered, and with respect to the ground truth event log. Furthermore, concerning the second goal, for a randomly selected set of 10 discovered models for each system, the quality with respect to all event logs generated by that system was measured. Table 1 shows the measures that were used in the analysis. Each measure was given a short label for simplicity. All calculations were performed using the benchmarking framework CoBeFra [8].

4 Results

4.1 Estimation biases

In order to visually analyse estimation biases of fitness metrics, Fig. 2 show the distribution of ΔF as boxplots for the different fitness measures, conditioned on the completeness and noisiness of the event logs. Note that, next to the ideal levels of noise and completeness, only the 15% noise level and 50% completeness level are depicted due to space limitations. Nevertheless, these levels of noise and completeness seem to be representative for real-life event logs [19,25]. Note that the asterisks represent the mean difference, while the middle horizontal lines of the boxplots represent the median values.

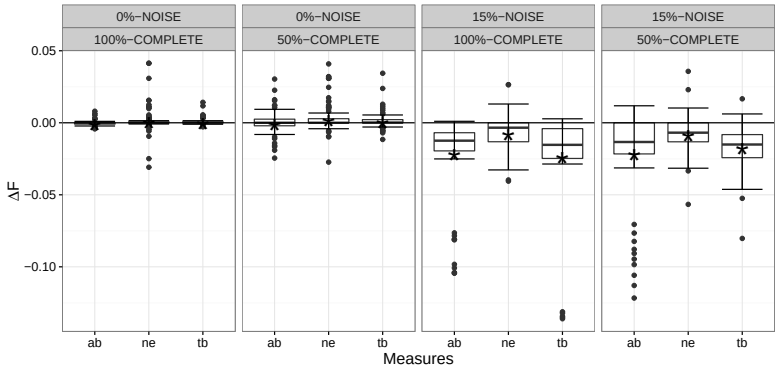

Figure 2. Distribution of ΔF for fitness measures

It can be observed that noise causes the average difference between log and system-based measurement of fitness metrics to be negative, which means that the fitness-measures generally underestimate the real system-fitness. In these cases, models are presumably being punished because they cannot replay behaviour that is not even real. On the other hand, the incompleteness of event logs does not seem to bias the fitness measures significantly, although for some measures it clearly decreases its precision as an estimator. Note that variability in the case when completeness is 100% and noise is 0% is the mere result of sampling variability in the composition of the event logs.

Fig. 3 shows the same graph for precision metrics. In contrast to system-fitness, system-precision appears to be slightly underestimated by most precision metrics when the event log is incomplete. Indeed, when event logs contain less behaviour, each model's log-precision will decrease, while the system-precision is independent of log completeness. Furthermore, the presence of noise seems to have an adverse effect.

In addition to the visual analysis, a Kruskal-Wallis test was done for each fitness and precision metric, to see whether there are statistically significant

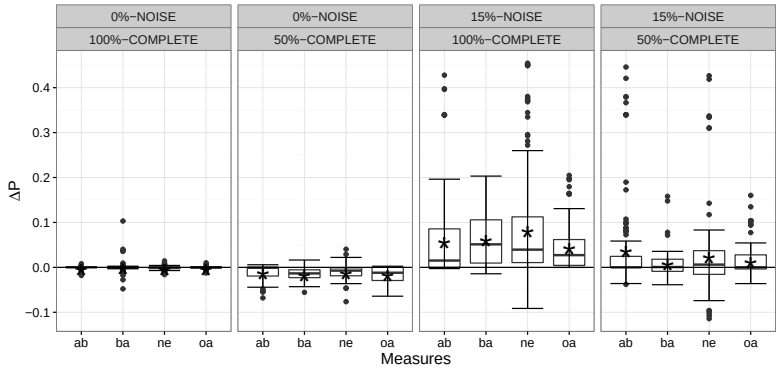

Figure 3. Distribution of ΔP^z for precision measures

differences between ΔF or ΔP for different levels of noise and completeness. It can be observed in Table 2 that for all the metrics, the hypothesis that there are no significant differences among the groups, is rejected.

Table 2. Results of Kruskal-Wallis rank sum test by completeness and noise

Fitness Metrics	Kruskal-Wallis χ^2	Precision Metrics	Kruskal-Wallis χ^2
Alignment Based Fitness	120.7946^{***}	Alignment Based Precision	1114.2523^{***}
Negative Event Recall	352.5966^{***}	Negative Event Precision	1006.2333^{***}
Token-based Fitness	320.5694^{***}	Best Align Precision	587.7634^{***}
		One Align Precision	1276.3387^{***}

Note: *p<0.1; **p<0.05; ***p<0.01

In order to further understand these differences, Table 3 and 4 show the average difference between log and system measures, for fitness and precision metrics, respectively. The levels of statistical significance indicated in these table reflect whether this mean is different from zero. Since the data suffers from non-normality, the non-parametric Wilcoxon signed rank test was used.

Note that for each metric, 16 different tests were done. In order to limit the family-wise error to 5%,, the Bonferonni correction was applied. Therefore, aach individual test has to be at a signficance level of $1 - 0.95^{(1/16)} = 0.0032$, annotated with the $^{+}$ symbol in Table 3 and 4. It can be seen that for fitness metrics, there is indeed a bias for noisy logs. The bias for noise-free but incomplete logs is less prevalent and not significant at a 0.32% significance level. However, it must be observed that this corrected significance level is very restrictive, as the Kruskall-Wallis test already pointed out that biases do exist. Notwithstanding the statistical significance, the real impact of the bias is limited.

For precision metrics, it is clear that both noisiness and incompleteness of event log creates significant biases for the quality metrics. It is remarkable that ΔP was found to be significantly different from zero for the Negative Event

Table 3. Mean ΔF for fitness metrics under differing noise and completeness levels.

Metric	Completeness	Noise 0%	5%	10%	15%
Alignment Based Fitness	100%	-0.0003**	-0.0061+	-0.0124+	-0.0179+
	75%	-0.0017***	-0.0072+	-0.0134+	-0.0185+
	50%	-0.0007	-0.0062+	-0.012+	-0.018+
	25%	0.0000	-0.0059+	-0.013+	-0.0179+
Negative Event Recall	100%	0.0005	-0.0016+	-0.0039+	-0.0058+
	75%	0.0000	-0.0018+	-0.0039+	-0.0058+
	50%	0.0011**	-0.0017+	-0.0035+	-0.0059+
	25%	0.0017**	-0.0001**	-0.0043+	-0.0048+
Token-based Fitness	100%	0.0004	-0.0048+	-0.0109+	-0.0161+
	75%	0.0007*	-0.0035+	-0.0078+	-0.0135+
	50%	0.0009**	-0.0028+	-0.0079+	-0.0115+
	25%	0.0016	-0.0012+	-0.0044+	-0.0056+

Note: *p<0.1; **p<0.05; ***p<0.01;+p<0.0032
Based on Wilcoxon signed rank test with continuity correction

Precision and One-Align Precision metrics under the condition of noise-free and complete logs. Further experiments have to be conducted to see whether this result is reproducible.

Table 4. Mean ΔP for precision metrics under differing noise and completeness levels.

Metric	Completeness	Noise 0%	5%	10%	15%
Alignment Based Precision	100%	-0.0004	0.0726+	0.093+	0.0989+
	75%	-0.0025+	0.0539+	0.0729+	0.0895+
	50%	-0.008+	0.0343+	0.0617+	0.0653+
	25%	-0.0195+	0.0069	0.0162+	0.0239+
Best Align Precision	100%	0.0004	0.064+	0.0827+	0.1008+
	75%	-0.004+	0.0401+	0.0508+	0.0609+
	50%	-0.0098+	0.0233+	0.04+	0.0446+
	25%	-0.0261+	-0.0042	0.0137	0.01
Negative Event Precision	100%	-0.0009+	0.0786+	0.0887+	0.1076+
	75%	-0.0056+	0.0398+	0.0637+	0.0803+
	50%	-0.0102+	0.0228+	0.0392+	0.0518+
	25%	-0.0255+	-0.0062+	-0.0074**	0.0082
One Align Precision	100%	-0.0002**	0.0637+	0.0873+	0.0872+
	75%	-0.0036+	0.0456+	0.0615+	0.0752+
	50%	-0.0111+	0.0224+	0.0485+	0.0506+
	25%	-0.0286+	-0.0014	0.0079*	0.0174+

Note: *p<0.1; **p<0.05; ***p<0.01;+p<0.0032
Based on Wilcoxon signed rank test with continuity correction

It is clear that precision measures, and to a lesser extent fitness measures, are not always unbiased nor reliable estimators of the system-alignment, as they fail to adequately estimate a model's system-fitness and system-precision. However, if these estimation errors are consistent among all models, the correct ranking of models will be preserved. It is therefore essential to investigate how the effects of noise and completeness differ among models.

4.2 Ranking biases

In order to investigate ranking biases, 10 of the discovered models for each system were randomly chosen. Subsequently, the quality of these models with respect to all event logs generated from this system was measured. The relationship between the value of these metrics and the level of noise and completeness were subsequently analyzed. Fig. 4 shows the relationship between the level of completeness and the fitness and precision values. Note that only noise-free logs were considered in this graph, in order to isolate the effect of (in)completeness.

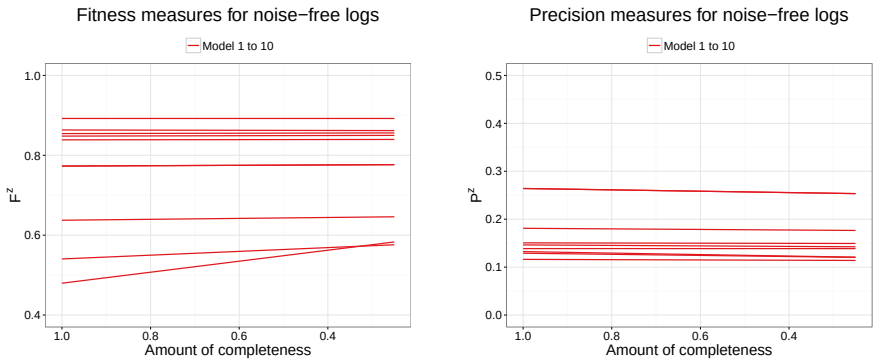

Figure 4. Relating fitness and precision measures to the completeness of logs

Each line in this Figure represents one of the 10 models, and its height represents the average fitness (precision) value at a given level of completeness. These values were averaged over all logs at the given completeness level and over all fitness (precision) metrics. Since fitness (precision) metrics were largely correlated, they are not longer individually distinguished from each other. Moreover, the graphs for individual metrics were found to be similar. Note that the mapping between each of the lines and the models is irrelevant for our purpose. Also observe that this figure only shows the results for one of the systems, though the results for other systems were found to be similar.

When one would draw a vertical line at a certain level of completeness, the intersections with the lines of the graph define a ranking on the models. The intersection with the highest line refers to the model which is perceived the best, while the lowest intersection will point out which model is the worst. Consequently, when the lines of two different models cross one another, the ranking between these two models will change. The intersections with the y-axis reflect to correct ranking of the models with respect to the underlying system, as logs are complete and noise-free at this point. Under such circumstances, metrics are unbiased estimators of system-quality, as was demonstrated in Section 4.1. The more cross-overs that take place between lines when moving rightwards, the more the true ranking is distorted.

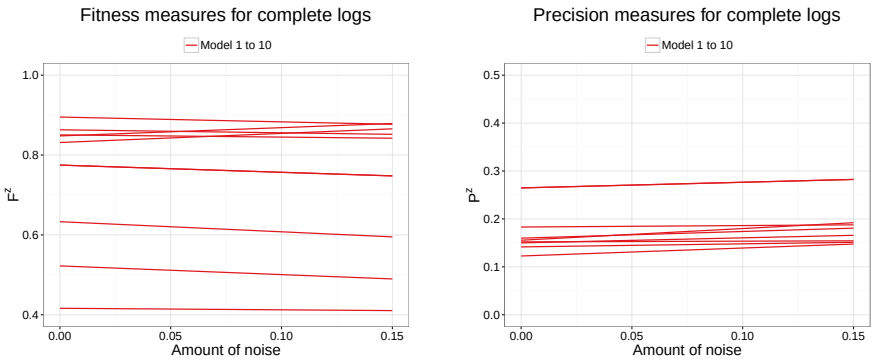

Figure 5. Relating fitness and precision measures to the amount of noise

In can thus be seen that only a very limited number of cross-overs occur concerning the ranking of both fitness and precision under increasing levels of incompleteness. Thus, incomplete logs seem to induce only a minimal ranking bias. The same analysis is done for the impact of noise in Fig. 5. Here it can be seen that for both fitness and precision, several cross-overs occur, distorting the ranking of models. Moreover, note that the fact that the *best* model is not always impacted by the perturbations in the ranking, does not solve the problem. After all, there is no guarantee that this *best* model will always be found.

It can be concluded that the existence of noise impacts the measurement of fitness and precision. Not only do they fail at estimating the quality of models correctly under these circumstances, the impact varies greatly among different models, which in turn significantly confuses their ranking. On the other hand, incompleteness of event logs only slightly biases the existing fitness and precision metrics when assessing a models quality with respect to the underlying system.

5 The role of generalization

It can be argued that the generalization quality dimension somewhat matches the system-based perspective, in particular system-fitness. In order to conduct a thorough analysis, we therefore also investigate the distance between the generalization measures *ab* and *ne* on the one hand, and both system-fitness and system-precision on the other hand.

This distance, for each log and model is defined as $\Delta G_F(M, L, S) = G(M, L) - F^{ab}(M, S)$ for system-fitness, and $\Delta G_P(M, L, S) = G(M, L) - P^{ab}(M, S)$ for system-precision. For the sake of clarity, generalization measures were only compared with one system-fitness and one system-precision measure, i.e. F^{ab} and P^{ab}. These measures where chosen because of their relatively intuitive interpretations. The distribution of ΔG_F and ΔG_P can then be analyzed as before, for both the *alignment based* and *negative event* generalization metrics. This was done in Figure 6.

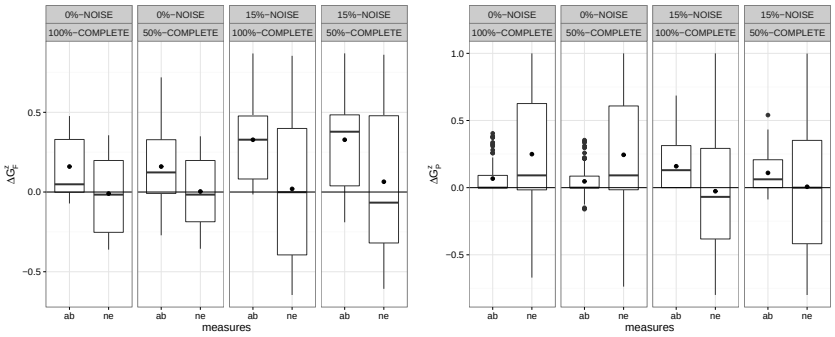

Figure 6. Distribution of ΔG^z_F and ΔG^z_P

It can be observed that G^{ne} is a relatively good estimator of system-fitness, although it is biased when logs are both noisy and incomplete. On the other hand, there does not seem to be any relationship with system-precision, as one would expect based on the definition of generalization. However, G^{ab} is not a good predictor of system-fitness, nor system-precision. Moreover, it should be noted that the generalization measures were hardly correlated (0.176), which confirms the fact that generalization remains a vague and ambiguous concept, both regarding its definition and its implementations.

6 Related work

Over time, many challenges within the field of process discovery, such as dealing with duplicate tasks and non-free choice constructs have been tackled [13]. Consequently, new process discovery algorithms increasingly focus on outperforming existing algorithms rather than tackling new challenges. This shift in research requires an agreed-upon and scientifically sound evaluation framework to compare different process mining algorithms. Recently, first attempts towards the development of an evaluation framework have been made [8,10,21]. The set of evaluation measures is the area which has received most attention so far.

In [2], the author states that "process discovery and conformance checking aim to tell something about the unknown real process rather than the example traces in the event log". The author therefore claims that, the one and only goal of process discovery would be to represent the true underlying process. However, existing quality metrics are mostly focussed on the relationship between the model and the event log. Furthermore, little empirical evidence exists so far.

The problem of log incompleteness is well acknowledged in literature. In [25], the authors defined different estimators of log completeness. The application of these on several real-life data sets showed that the estimated coverage of event logs is only about 50%, bearing in mind that the estimators were even found to be over-estimating the coverage when tested on artificial event logs.

Noise has been defined less unambiguously. According to [15], noise covers the occurrence of errors, the incompleteness of the event log, as well as exceptional behaviour. Other authors have equated noise only with exceptional behaviour [1]. Finally, some authors have defined noise as measurement errors [23]. The latter definition has been adopted in this paper, as it matches the classical definition of noise in the field of data mining.

Dealing with incomplete as well as noisy event logs has been tackled by several process discovery algorithms, notably the Heuristics Miner [24] and the Inductive Miner [18]. In the field of declarative process models, [14] systematically analysed the sensitivity of mined declarative constrains to noise. Using similar types of noise as in this paper, the authors empirically confirmed which types of declarative constraints are (not) resilient to certain types of noise.

Another approach towards handling noise in process discovery was proposed in [11]. Here, handling of noise is regarded as a preprocessing step, i.e. *cleaning* the log, before discovery algorithms are applied. While this view on managing noise in event logs definitely has its merits, more research is needed on how to distinguish noise in an event log, and how to do this in an automatic way.

Despite the efforts towards handling noise in process discovery, the concepts of noise and log-incompleteness are not incorporated in most process quality measures. As a result, one must be cautious to use the same quality measures when the goal is rather to describe the underlying process. A notable exception has been the work on artificial negative events [7]. The induction of negative events explicitly supports the fact that event logs are not complete. The negative events aim at delineating the system by defining its complement S^c. The related quality metrics are thus expected to be more suitable for measuring a process model's alignment with the system rather than the event log, although this is not exactly clear from the analyses.

7 Conclusions and future work

This papers suggest that there are different objectives within process discovery. To gain information on how work is done in a business process, the objective of process discovery is to learn a model which provides a good representation of the underlying process, i.e. the system. However, when process discovery is used for auditing purposes, the mere objective might be to discover a model which is limited to the behaviour described in the event log.

Although discovery algorithms have been able to tackle noisiness and incompleteness of event logs, existing quality measures are predominantly focused on the event log as the unmistaken truth. In order to examine whether these measures still perform well in case of noisy and incomplete event logs, their sensitivity to these issues was investigate. The results show that both fitness and precision measures are very sensitive to noise, which makes them biased estimators of system-precision. Moreover, when event logs are incomplete, the variability of the measurements increase. Under these circumstances, there is no guarantee that the metrics will be able to correctly assess a models quality with respect

to the underlying system, and rank different models accordingly. Furthermore, it is unclear what existing generalization measures quantify. Moreover, the two generalization measures under consideration were found to be hardly correlated.

Ranking biases clearly need to be further investigated from a more statistical point of view. Further research concerning the existing generalization measures is also needed to uncover their added value. Moreover, it should be investigated how existing measures can be corrected in order to remove their bias as estimator for system-fitness and system-precision in the presence of noise and incomplete event logs, which is especially needed for precision measures. Also, the need for confidence intervals for quality metrics should be further investigated, in order to assess the reliability of their results.

Acknowledgments. The computational resources and services used in this work for both process discovery and process conformance tasks were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation - Flanders (FWO) and the Flemish Government.

References

1. van der Aalst, W.M.P.: Process mining: discovery, conformance and enhancement of business processes. Springer, Heidelberg (2011)
2. van der Aalst, W.M.P.: Mediating between modeled and observed behavior: the quest for the Right process. In: IEEE Computing Society. pp. 31–43 (2013)
3. van der Aalst, W.M.P., Adriansyah, A., van Dongen, B.: Replaying history on process models for conformance checking and performance analysis. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 2(2), 182–192 (2012)
4. Adriansyah, A., Munoz-Gama, J., Carmona, J., van Dongen, B.F., van der Aalst, W.M.P.: Alignment based precision checking. In: Business Process Management Workshops. pp. 137–149. Springer (2013)
5. Agrawal, R., Gunopulos, D., Leymann, F.: Mining process models from workflow logs. In: Schek, H.J., Saltor, F., Ramos, I., Alonso, G. (eds.) Advances in Database Technology - EDBT '98. vol. 1377, pp. 467–483. Springer-Verlag Berlin Heidelberg (1998)
6. Baier, C., Katoen, J.P., et al.: Principles of model checking, vol. 26202649. MIT press Cambridge (2008)
7. vanden Broucke, S.K.L.M., De Weerdt, J., Vanthienen, Jan, B., Baesens, B.: Determining process model precision and generalization with weighted artificial negative events. Knowledge and Data Engineering, IEEE Transactions on 26(8), 1877–1889 (2014)
8. vanden Broucke, S.K.L.M., De Weerdt, J., Vanthienen, J., Baesens, B.: A Comprehensive Benchmarking Framework (CoBeFra) for conformance analysis between procedural process models and event logs in ProM. In: Computational Intelligence and Data Mining (CIDM), 2013 IEEE Symposium on. pp. 254–261. IEEE (2013)
9. Buijs, J.C.A.M.: Flexible Evolutionary Algorithms for Mining Structured Process Models. Ph.D. thesis, Technische Universiteit Eindhoven, Eindhoven (2014)
10. Buijs, J.C., van Dongen, B.F., van der Aalst, W.M.P.: On the role of fitness, precision, generalization and simplicity in process discovery. In: On the Move to Meaningful Internet Systems: OTM 2012, pp. 305–322. Springer (2012)

11. Cheng, H.J., Kumar, A.: Process mining on noisy logs can log sanitization help to improve performance? Decision Support Systems 79, 138 – 149 (2015)
12. Cook, J.E., Wolf, A.L.: Software process validation: quantitatively measuring the correspondence of a process to a model. ACM Transactions on Software Engineering and Methodology (TOSEM) 8(2), 147–176 (1999)
13. De Weerdt, J., De Backer, M., Vanthienen, J., Baesens, B.: A multi-dimensional quality assessment of state-of-the-art process discovery algorithms using real-life event logs. Information Systems 37(7), 654–676 (2012)
14. Di Ciccio, C., Mecella, M., Mendling, J.: Data-Driven Process Discovery and Analysis: Third IFIP WG 2.6, 2.12 International Symposium, SIMPDA 2013, Riva del Garda, Italy, August 30, 2013, Revised Selected Papers, chap. The Effect of Noise on Mined Declarative Constraints, pp. 1–24. Springer (2015)
15. Folino, F., Greco, G., Guzzo, A., Pontieri, L.: Discovering expressive process models from noised log data. In: Proceedings of the 2009 international database engineering & applications symposium. pp. 162–172. ACM (2009)
16. Janssenswillen, G., Depaire, B., Jouck, T.: Calculating the number of unique paths in a block-structured process model. In: Algorithms & Theories for the Analysis of Event Data (2016)
17. Jouck, T., Depaire, B.: Generating Artificial Data for Empirical Analysis of Process Discovery Algorithms: a Process Tree and Log Generator. Technical Report, Universiteit Hasselt, Universiteit Hasselt (Mar 2016)
18. Leemans, S.J., Fahland, D., van der Aalst, W.M.: Discovering block-structured process models from event logs containing infrequent behaviour. In: Business Process Management Workshops. pp. 66–78. Springer (2013)
19. de Medeiros, A.K.A.: Genetic process mining. Ph.D. thesis, Technische Universiteit Eindhoven, Eindhoven (2006)
20. Rozinat, A., van der Aalst, W.M.P.: Conformance checking of processes based on monitoring real behavior. Information Systems 33(1), 64–95 (2008)
21. Rozinat, A., De Medeiros, A.K.A., Günther, C.W., Weijters, A.J.M.M., van der Aalst, W.M.P.: Towards an evaluation framework for process mining algorithms. Beta, Research School for Operations Management and Logistics (2007)
22. Weber, P., Bordbar, B., Tiňo, P., Majeed, B.: A framework for comparing process mining algorithms. In: GCC Conference and Exhibition (GCC), 2011 IEEE. pp. 625–628. IEEE (2011)
23. Weijters, A.J.M.M., van der Aalst, W.M.P.: Rediscovering workflow models from event-based data. In: Proceedings of the 11th Dutch-Belgian Conference on Machine Learning (Benelearn 2001). pp. 93–100. Citeseer (2001)
24. Weijters, A.J.M.M., van der Aalst, W.M.P., De Medeiros, A.K.A.: Process mining with the heuristics miner-algorithm. Technische Universiteit Eindhoven, Tech. Rep. WP 166, 1–34 (2006)
25. Yang, H., van Dongen, B., ter Hofstede, A., Wynn, M., Wang, J.: Estimating completeness of event logs. BPM Center Report, 12-04-2012 (2012)