# Statistical Tests and Association Measures for Business Processes

Sander J.J. Leemans*, James M. McGree†, Artem Polyvyanyy‡, Arthur H.M. ter Hofstede†
*RWTH Aachen †QUT Brisbane ‡University of Melbourne

**Abstract**—Through the application of process mining, organisations can improve their business processes by leveraging data recorded as a result of the performance of these processes. Over the past two decades, the field of process mining evolved considerably, offering a rich collection of analysis techniques with different objectives and characteristics. Despite the advances in this field, a solid statistical foundation is still lacking. Such a foundation would allow analysis outcomes to be found or judged using the notion of statistical significance, thus providing a more objective way to assess these outcomes. This paper contributes several statistical tests and association measures that treat process behaviour as a variable. The sensitivity of these tests to their parameters is evaluated and their applicability is illustrated through the use of real-life event logs. The presented tests and measures constitute a key contribution to a statistical foundation for process mining.

**Index Terms**—Process mining, statistical tests, association, correlation.

## 1 Introduction

Organisations improve their business processes using recorded historical behaviour: each case in the process – a claim, order, shipment – traverses the process, and each step – deciding, processing, packing – is recorded by an information system, from which an event log can be extracted that describes this process behaviour. Process mining aims to derive actionable insights from these event logs. Process mining consists of a broad range of techniques, for instance to automatically derive a process model from an event log (*process discovery*), to compare a process model with an event log for deviations (*conformance checking*) and to obtain global or detailed process-based performance measures from process models and event logs [1]. These techniques are often applied sequentially by analysts to derive insights into the business process. For instance, to derive performance measures, first a process model is discovered, after which its quality is verified using a conformance technique, after which the performance measures can be computed.

Existing process mining techniques often work on a best-effort basis. That is, very few existing techniques offer guarantees or have solid statistical foundations. Consequently, results obtained from process mining techniques need to be verified by analysts, which is complicated by the sequential accumulation of weaknesses and complex interplay of the techniques [1]. Several approaches have been proposed to address this. For instance, in medical research, statistical significance of results is a necessity. Some techniques address this problem for the performance of single activities [2]. However, to enable the application of process mining in such fields, comprehensive statistical procedures that operate on the behaviour of entire processes are necessary.

In this paper, we introduce statistical methods for process mining. In particular, we introduce statistical tests for process behaviour, and association measures to assess relationships between process behaviour and trace data.

The methods of this paper enable several new types of analyses and hypotheses, of which some examples are:

**Ex1**: From an event log, we have discovered two process models using two different automated discovery techniques that we want to compare. Which of the two process models best fits the log, and is there a statistically significant difference in behaviour between the models? While existing conformance checking techniques or benchmarks could answer the first question, the measures they provide do not bear statistical meaning [3], [4].

**Ex2**: Several city councils are executing a similar process that is prescribed by law, but implemented slightly differently in each city council [5]. We take a sample event log of two city councils. Can we reject the null-hypothesis that the event logs were derived from the same underlying unknown model, and thus conclude that there are statistically significant differences between the way the councils execute the process?

**Ex3**: A company recognises several types of customers: silver, gold and platinum. Is there a statistically significant difference or relationship between the processing of these types of customers?

**Ex4**: A company strategically aims to improve customer satisfaction. Using data obtained from customer surveys, is there a statistically significant difference in processing of satisfied customers versus dissatisfied customers [6], or is there a relationship between the satisfaction of the customers and the process followed? Is there a relationship between conformance to the process and the overall duration?

**Ex5**: A company is optimising a business process and several redesigned scenarios have been proposed, to preserve the customer-focused behaviour as much as possible. Using what-if analyses and simulation, these redesigns can be evaluated, while a statistical test expresses the absence of significant differences in the process of customer interactions between the redesigned alternatives and the current situation.

**Ex6**: Changing processes pose a challenge to process mining techniques, as models, conformance information and performance can only be reported on a single process [7]. Concept drift detection splits an event log into sub-logs according to process changes in the log.While [8] considers the stochastic perspective of behaviour, that is, a drift point might be related to how often parts of the process are executed, ideally such a point should imply a statistically significant difference between the processes before and after the point.

From these examples, two generic problems can be derived. The first one covers Ex1, Ex2, Ex3 and Ex5, which involve a decision on whether observed processes are different "enough":

**Problem statement #1**: given two representations of two processes (that is, logs and/or process models), what is the probability that they were derived from the same underlying process?

The second one covers Ex4 and Ex6, which involve an assessment of the extent of a relation between process behaviour and other data:

**Problem statement #2**: given a process and a trace attribute, what is the association between the process and the attribute?

Consequently, to address these problems and the motivational examples, we introduce several statistical tests and association measures grounded in the bootstrap method, and we provide their open-source implementation in the ProM framework [9]. We evaluate the sensitivity of the introduced methods to their varying parameters, and illustrate their applicability on real-life logs.

The remainder of this paper is structured as follows. Section 2 introduces existing concepts. Section 3 discusses requirements for statistical tests in process mining and elaborates on why existing statistical methods cannot be used. Section 4 introduces statistical tests, Section 5 introduces association measures, Section 6 evaluates the introduced methods, Section 7 discusses related work, and Section 8 concludes the paper.

## 2 Preliminaries

In this section, we introduce key concepts from process mining and statistical methods.

### 2.1 Stochastic Languages & Models

An *event* represents the execution of a process step, and a *trace* is a sequence of events, which represent the process steps being executed for a particular case in a process. A trace can be annotated with *trace attributes*, denoting properties of the trace. Given an alphabet of process steps (*activities*) $\Sigma$, $\mathcal{T}$ denotes the set of all possible traces over $\Sigma$. An *event log* is a collection of traces $\subseteq \mathcal{T}$. We refer to the set of projections of traces onto their sequences of activities as the set of *trace variants*. For instance, the trace $\langle a, b, c\rangle^{\text{amount}=10}$ consists of three events and has one attribute (amount).

***Definition 1 (stochastic language).*** A *stochastic language* is a function $L$ that maps trace variants onto probabilities:

$$L\colon \mathcal{T} \to [0,1] \text{ such that } \sum_{t\in\mathcal{T}} L(t) = 1$$

In this paper, we assume that $L$ is known and can be queried $L(t)$ to obtain the particular probability of a trace $t$, as well as be queried for all traces for which a non-zero probability is expressed: $\mathcal{L}(L) = \{t \mid L(t) > 0\}$. An *(in)finite stochastic language* $L$ has an (un)bounded number of traces $\mathcal{L}(L)$.

An event log expresses a finite stochastic language, which can be obtained by dividing the occurrence of each trace variant by the total number of traces in the event log. Thus, the techniques proposed in this paper can be applied to any standard event log. Please note that support for event logs with lifecycle information depends on the chosen trace and process distance functions $\delta$ and $\Delta$; EMSC supports lifecycle information implicitly, if it is contained in the activity notion.

A *stochastic process model* expresses a possibly infinite stochastic language, typically in human-readable form. Several stochastic process modelling formalisms have been proposed, such as Stochastic Petri Nets, Generalised Stochastic Petri Nets, and Generalised Stochastic Labelled Petri nets; a detailed introduction of these formalisms is beyond the scope of this paper. Please note that the techniques presented in this paper are defined over stochastic languages of stochastic process models – time and other additional modelled information is ignored – and are therefore flexible in the stochastic modelling formalisms that can be used.

### 2.2 Process & Trace Distances

The *Levenshtein distance* expresses the minimum number of edit operations (remove event, add event, substitute event) to transform one trace into another [10]. The *normalised Levenshtein distance* is the Levenshtein distance divided by the length of the longest of the two traces, and is a number between 0 and 1.

The *Earth Movers' Stochastic Conformance* (EMSC) technique measures a normalised distance between two stochastic languages, that is, between any combination of event logs and stochastic models [11]. To this end, a minimum-cost reallocation matrix is computed that transforms the first stochastic language into the second. Process models with loops are sampled. The cost of this matrix $\in [0, 1]$ is the distance between the stochastic languages.

### 2.3 Statistical Methods

*Covariance.* The *covariance* of two numerical variables expresses the direction of a relationship between the two variables: a positive covariance indicates that the larger values of one variable correspond to the larger values of the other variable, while a negative covariance indicates that the lower values of one variable correspond to the higher values of the other variable. For two random variables $X$ and $Y$, covariance can be computed as: $\text{cov}(X, Y) = E((X - \bar{X})(Y - \bar{Y}))$.

*Bootstrap.* Suppose we are interested in some property $\hat{\theta} = s(x)$ for some function $s$, where $x \sim F(x)$, $F(x)$ is an unknown distribution function, and $x = (x_1, \ldots, x_n)$. In such cases, the sampling distribution of $\hat{\theta}$ is unknown, so finding, for example, the distribution of $\hat{\theta}$ poses some difficulties. Fortunately, the *bootstrap* method can be used to form an approximation to this sampling distribution, thus allowing inference to be performed [12].

The bootstrap method was proposed as the basis for some of the methods in this paper as it is a general approach for approximating (bootstrap) the sampling distribution of a given statistic without assuming a parametric form for the distribution of the data. This is particularly useful when analysing, for example, distances as they are generally non-negative and can be highly skewed, meaning that assuming normality is not appropriate. Having the sampling distribution of a given statistic readily available means (bootstrap) confidence intervals can be formed straightforwardly, which provides an assessment of uncertainty, and can be used to draw inference about the population value.

To construct an approximation to the probability distribution $F(x)$, place probability mass of $1/n$ at each point of $x$. Denote this as $\hat{F}(x)$; a non-parametric estimate of $F(x)$. With $\hat{F}(x)$ fixed, draw a random sample of size $n$ with replacement from $\hat{F}(x)$ i.e. $x^{(b)} \sim \hat{F}(x)$. This is called the *bootstrap sample*. Based on this sample, $\hat{\theta}^{(b)} = s(x^{(b)})$ can be evaluated, and this is known as a sample from the bootstrap distribution of $\hat{\theta}$, which forms an approximation to the sampling distribution of $\hat{\theta}$. Thus, one can then approximate the sampling distribution of $\hat{\theta}$ by drawing many samples (of size $n$) from $\hat{F}(x)$ and evaluating $s(x^{(b)})$ for each sample. The collection of these draws is known as the bootstrap distribution. Algorithm 1 provides pseudo code for drawing from the bootstrap distribution.

**Algorithm 1** One-sample bootstrap algorithm

```
Initialise $x, s(x), \hat{F}(x)$
for $b \in [1 : B]$ do
    Sample $x^{(b)} \sim \hat{F}(x)$ of size $n$ with replacement with probability $1/n$
    Evaluate $\hat{\theta}^{(b)} = s(x^{(b)})$
end for
Approximate sampling distribution of $\hat{\theta}$ with $\{\hat{\theta}^{(1)}, \ldots, \hat{\theta}^{(B)}\}$
```

$\chi^2$ *test* Given two categorical variables, the $\chi^2$ (e.g. Pearson [13]) test establishes whether there is a statistically significant difference between frequency distributions of the values of the two categorical variables. That is, whether the distribution of values of one variable differs from the value of the other variable. The Pearson test assumes that both variables are normally distributed and independent. Let $k$ be the number of categories (in our case: number of trace variants); $n$ the number of observations (in our case: number of traces in the log); $x_i$ the number of times $i$ was observed (in our case: number of times a trace variant ($i$) is in the log); $p_i$ the probability of trace variant ($i$) in the model; and $m_i = np_i$ number of traces of a trace variant ($i$) expected in the draw. Then, $\chi^2 = \sum_{i=1}^{k} \frac{(x_i - m_i)^2}{m_i} = \sum_{i=1}^{k} \frac{x_i^2}{m_i} - n$. Some observations on this test are that (1) if $\exists x_i = 0$ (in our case: if a model does not support a trace), then there is a division by zero, and (2) if the language of the model is infinite, then $m_i \to 0$, which violates the test assumption that all traces are sufficiently often seen.

*Simulation-based approaches.* Simulation can also be used to assess whether a model could have generated some given data by assessing the closeness of generated data to some given observed data. An example of such an approach is approximate Bayesian computation which aims to find a posterior distribution based on samples of parameters that yield simulated data that are close to the initial sample [14]. Such methods can also be used to compare models, i.e. determine which model is more likely to generate data that are close to the observed data. In this paper, we use this approach as the basis for choosing between different logs, i.e. to determine which log is more likely to have generated a given trace.

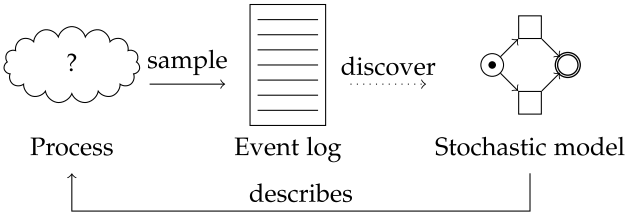


Fig. 1: Context of process mining.

## 3 Context & Requirements for Process-Based Statistical Methods

In this section, we sketch the context of process mining, define the problem we are addressing, gather requirements for process-based statistical techniques, and show that typical statistical techniques do not satisfy these requirements.

As illustrated in Figure 1, the *process* is executed in an organisation. A process is unknown, but a sample of traces, an *event log*, can typically be extracted from information systems that support the process. Using an event log, process mining techniques can automatically discover a stochastic process model [15], which aims to describe the process well and to be interpretable by human analysts.

The input of a process-based statistical test is either (i) two event logs, (ii) a stochastic process model and an event log, or (iii) two stochastic process models, where the output of a process-based statistical test is a likelihood that the two stochastic languages are derived from the same underlying and unknown business process. The input of a process-based correlation measure is an event log, in which the traces have been annotated with an attribute.

Process-based tests and correlation measures inherently differ from standard statistical methods; as such, we have identified the following requirements.

**Req1**: *Process-based statistical methods that support stochastic process models should support infinite behaviour, as any process model with a loop contains an infinite number of trace variants.*

This limits the applicability of standard statistical techniques, as categorical data is typically not assumed to be infinite. Some measures, such as EMSC [11], can, in some cases, be computed algebraically, however this challenges the practical applicability of these measures.

**Req2**: *Process-based statistical methods should be forgiving against non-occurring behaviour.*

In process mining, it is desirable that models do not include infrequent behaviour: human analysts interpret the models, so simplicity is an important quality dimension. Consequently, it is likely for a log $L$ and a model $M$ that for some $t \in L$ it holds that $M(t) = 0$.

TABLE 1: Illustrations of requirements for process-based statistical methods.

(a) Methods should be forgiving towards non-occurring behaviour (Req2).

| trace | $L$ | $M_1$ | $M_2$ |
|---|---|---|---|
| $\langle a, b\rangle$ | 1000 | 1 | 0.01 |
| $\langle c\rangle$ | 1 | 0 | 0.99 |

(b) Trace distance should be taken into account (Req3).

| trace | $L$ | $M_1$ | $M_2$ |
|---|---|---|---|
| $\langle a, b, c, d, e, f\rangle$ | 10 | 0 | 0 |
| $\langle a, b, c, d, e, g\rangle$ | 0 | 0.99 | 0.01 |
| $\langle a\rangle$ | 0 | 0.01 | 0.99 |

For instance, the example shown in Table 1a contains a log $L$ and two models $M_1$ $M_2$ such that $L$ contains a trace $\langle c\rangle$ that is not supported by model $M_1$, but is supported by $M_2$ with a high probability. In this example, we argue that $M_1$ is a more likely explanation for $L$ than $M_2$, despite this one trace $\langle c\rangle$. In other words: $M_1$ is preferred over $M_2$, because data generated from $M_1$ will be more similar to $L$ than data generated from $M_2$.

Standard statistical tests would consider that $L$ has no chance of being derived from $M_1$, as it contains an observation ($\langle c\rangle$) that is not in $M_1$. Consequently, these tests would return 0 or be undefined, and thus not yield information.

**Req3**: *Process-based statistical tests should take distances between traces into account.*

Process mining aims to analyse the behaviour logged in an event log. In many real-life processes, deviations or exceptions may occur. However, such a deviating event entails only a small portion of all steps executed in a trace, thus it would be a waste of information to disqualify an entire trace just for a single deviating event to, e.g., measure performance or frequencies of the "correct" parts of the trace. Therefore, state-of-the-art process mining techniques perform approximate matching using distance measures between traces.

For instance, consider the example shown in Table 1b, consisting of a log $L$ and two models $M_1$ and $M_2$. The trace $\langle a, b, c, d, e, f\rangle \in L$ does not appear in $M_1$ or $M_2$. For this example, we would argue that $L$ is more likely to be derived from $M_1$ than from $M_2$, as the trace $\langle a, b, c, d, e, f\rangle$ is arguably closer to the trace $\langle a, b, c, d, e, g\rangle$ than to the trace $\langle a\rangle$.

Standard statistical tests would consider that $L$ has no chance of being derived from either $M_1$ or $M_2$, as it contains an observation ($\langle a, b, c, d, e, f\rangle$) that is not in $M_1$ or $M_2$. Consequently, these tests would return 0, or be undefined, rather than considering what the most likely model for $L$ is.

**Req4**: *If a process-based statistical test has a parameter $s$ indicating the resample size or $n$ indicating the number of repetitions, then the test should be deterministic and converge to the deterministic correct answer for $s \to \infty$ and $n \to \infty$.*

In the tests and measures introduced in the remainder of this paper, two auxiliary measures might be used: a language distance measure and a trace distance measure. The introduced tests and measures are agnostic to these auxiliary measures, though some pose additional requirements.

For the trace distance measure, we choose the normalised Levenshtein distance: it matches intuition of insertion and deletion, is normalised, and is relatively fast to compute. Alternatives include a strict equivalence test (which would not satisfy Req3), or any other string edit distance function, which could even take other attributes (such as cost or time) into account, or partial orders and concurrency. As such, the techniques presented in this paper constitute frameworks, that can be instantiated with other such functions and henceforth may possess different properties. A detailed study of these properties is beyond the scope of this paper.

TABLE 2: Overview of tests.

| | unknown process | known process | |
|---|---|---|---|
| | | log | model |
| 2 processes (log/log) | P-P-UP | P-P-L | P-P-L |
| 2 processes (log/model) | P-P-UP | P-P-L | P-P-L |
| 2 processes (model/model) | P-P-UP | P-P-L | P-P-L |
| multiple processes (log/categorical trace attribute) | L-CA | - | - |

For the language distance measure, we choose EMSC: it is one of the few measures that takes the frequency of behaviour of both languages into account, which is necessary to satisfy Req2. Currently, the only other alternative would be Entropy-based stochastic fitness and precision [16], however due to its dual nature would introduce unwanted asymmetry in the tests and measures.

## 4 Statistical Tests

In this section, we introduce statistical tests for process behaviour; Table 2 provides an overview. First, there's a distinction between whether the underlying process is known (horizontal direction in Table 2). If the underlying process is unknown, a test establishes whether logs or models could have been derived from the same unknown underlying process. If the underlying process is known (and given in the shape of a log or a model), a test establishes which of two given logs or models is closest to the known underlying process. In the vertical direction, there are various combinations in which the processes can be provided. Please note that even though most of our tests apply to any combination of logs and models, this is due to the construction of these tests and not inherent. Comparing multiple processes to a known process remains future work.

In the remainder of this section, we discuss the three tests.

### 4.1 Process vs. Process - Unknown Process Test

Given two logs or process models $P_1$ and $P_2$, we would like to establish whether the logs show significantly different behaviour. For instance, $P_1$ could represent satisfied customers while $P_2$ could represent dissatisfied customers, and we would like to know whether there is a significant difference in the process of serving these groups of customers.

**Hypothesis**: *$P_1$ and $P_2$ were derived from the same underlying (unknown) process.*

To test this hypothesis, we evaluate how likely $P_2$ is if we assume the model is $P_1$. To do so, we apply the bootstrap method to approximate the sampling distribution of distances of logs generated under the model given by $P_1$, then determine how likely $P_2$ (or a more extreme log) would be observed under this model.

That is, we repeatedly take resamples from $P_1$ and compare these resamples with $P_1$ using a process distance measure $\Delta$. Next, we compare $P_1$ with $P_2$ using $\Delta$, and if this distance is higher than $1-\alpha$ of the sample distances, we reject the hypothesis. For instance, for $\alpha = 0.05$, at least 95% of the $P_1$ resamples need to have a smaller distance to $P_1$ than $P_2$ to reject the hypothesis. Algorithm 4.1 formalises this procedure.

**Algorithm 2** Log vs. Log - Unknown Process Test

```
function P-P-UP TEST(Process $P_1, P_2$, process difference $\Delta$, repetitions $n$, threshold $\alpha$)
    $D \leftarrow []$
    for $n$ times do
        $P_1' \leftarrow$ random sample of $P_1$ of size $|s|$ with replacement
        $D \leftarrow D \uplus [\Delta(P_1, P_1')]$
    end for
    $p \leftarrow |[d \mid d \in D \wedge d \leq \Delta(P_1, P_2)]|n$
    if $p \geq 1 - \alpha$ then
        reject hypothesis
    else
        do not reject hypothesis
    end if
end function
```

For this test, requirements Req2 and Req3 are satisfied if $\Delta$ satisfies them. The test supports infinite behaviour (Req1) if $\Delta$ is able to perform log-log, log-model and model-model comparisons where appropriate.

If EMSC is used for $\Delta$, then the run time complexity of Algorithm is linear in $n$, quadratic in the maximum trace length and polynomial in the number of trace variants.

If $P_1$ is a log, then the resample size $s$ is fixed at $|L_1|$ by the bootstrap method. However, if $P_1$ is a model, the resample size must be chosen carefully, as the resample size determines the extent to which the test can determine a difference. Big differences should be easy to detect and relatively small resample sizes would suffice, while minor differences would require larger resample sizes to detect the differences. In Section 6.3.2, we demonstrate an application of this test, while in Section 6.2.1, we study the behaviour of this test under different resample sizes.

#### *4.1.1 Further applications*

The same test can be applied to model-model settings as well: if we are given two models $M_1$ and $M_2$, and we would like to establish whether these models describe significantly different behaviour. For instance, both an as-is process model and a proposed to-be redesign are provided, abstracted to the most important customer interactions. We would like to know whether there are significant customer-facing changes. Notice that language-equivalence establishing algorithms cannot provide this information.
**Hypothesis**: *$M_1$ and $M_2$ describe equivalent stochastic behaviour.*

Given a process distance measure $\Delta$, one could measure the distance between $M_1$ and $M_2$ directly, and use a threshold on the measure to reject or sustain the hypothesis. However, this requires (1) $\Delta$ to satisfy Req2 (forgiving to absent behaviour) and Req3 (considering trace distance); (2) $\Delta$ to fully support infinite behaviour (Req1); (3) $\Delta$ to be consistent over different types of differences: the weighing of types of differences directly influences the result; and (4) $\Delta$ to provide well-defined semantics over small differences such that a well-defined threshold can be chosen, e.g., a difference of 0.05 vs. 0.04 might be important. Even then, statistical significance could not be concluded. To the best of our knowledge, no such $\Delta$ exists. Instead, the test can be conducted to test whether $M_1$ and $M_2$ describe a different process. Such a test satisfies Req1, Req2 and Req3 if $\Delta$ satisfies these requirements. Please note that apart from consistency, the actual values returned by $\Delta$ are unimportant, as the test only compares values of $\Delta$ with one another.

### 4.2 Process vs. Process - Log Test

Given an event log $L$ and two processes $P_1$, $P_2$, we would like to establish whether there is a significant difference in how well the processes represent the event log. For instance, two process discovery techniques were applied to a given event log, and we would like to establish which of the two models is the preferred model, that is, represents the log better than the other.
**Hypothesis**: *$P_1$ and $P_2$ represent $L$ equally well.* Alternative hypothesis: $P_1$ represents $L$ better. Alternative hypothesis: $M_P$ represents $L$ better.

Given a process distance measure $\Delta$, one could measure the distances $\Delta(L, P_1)$ and $\Delta(L, P_2)$ directly, and choose the least distance with some threshold. However, similar to the reasons mentioned in Section 4.1.1, such a $\Delta$ does not exist currently and would not provide statistical significance.

To test this hypothesis, we apply the Approximate Bayesian Computation method [14, Alg. B], where we repeatedly perform an experiment of taking a sample from both processes, and accepting the experiment if the sample of $P_1$ is closer to $L$ than the sample of $P_2$, according to a process distance function $\Delta$. If one of the processes wins this comparison often enough, based on a threshold $\alpha$, this provides evidence of statistically significant differences. Algorithm 3 provides a formal algorithm.

This procedure satisfies Req2 and Req3 if $\Delta$ satisfies these requirements, satisfies Req1, and requires a suitable sample size to be chosen. For $s \rightarrow \infty$, $p$ tends to 0 or 1 yielding the correct answer, while for $n \rightarrow \infty$, $p$ will stabilise (Req4).

### 4.3 Log vs. Categorical Attribute Test

Given an event log $L$, in which the traces are annotated with a categorical attribute $\varphi$, we would like to establish whether the sub-logs defined by the attribute were derived from different processes. For instance, an event log was derived from a traffic fine collections process, and we would like to establish whether the process is different for different types of traffic violations.
**Hypothesis**: *the sub-logs defined by $\varphi$ are derived from identical processes.*

To test this hypothesis, we apply the bootstrap method to establish whether knowledge of $\varphi$ decreases the average trace distance. That is, whether the average trace distance with knowledge of $\varphi$ – thus only for traces with the same

**Algorithm 3** Process vs. Process - Log Test

```
function P-P-L TEST(Log $L$, Process $P_1$, $P_2$, sample size $s$, repetitions $n$, process difference $\Delta$, threshold $\alpha$)
    $a \leftarrow 0$
    for $n$ times do
        $L_1 \leftarrow$ random sample of $P_1$ of size $s$ with replacement
        $L_2 \leftarrow$ random sample of $P_2$ of size $s$ with replacement
        if $\Delta(L, L_1) \leq \Delta(L, L_2)$ then $a \leftarrow a + 1$ end if
    end for
    $p \leftarrow a/n$
    if $p \leq 0.5\alpha$ then
        reject null hypothesis; $P_2$ represents $L$ better
    else if $p \geq 1 - 0.5\alpha$ then
        reject null hypothesis; $P_1$ represents $L$ better
    else
        do not reject null hypothesis
    end if
end function
```

value for $\varphi$ – is lower than the average trace distance without knowledge of $\varphi$, according to a trace distance measure $\delta$.

Intuitively, we take $n$ bootstrap samples of $L$, and measure the average trace distance in the sample between ($r$) all traces, and ($a$) all traces with equivalent $\varphi$. If $a$ is smaller than $r$, we have seen evidence against the null hypothesis, and we record this in $e$. If there is enough evidence, we reject the null hypothesis. For practical feasibility – the method is quadratic in the size of the resample – we parameterise the test with resample size $s$.

Formally, let $L \subseteq \mathcal{R}$ be a collection of traces, let $\varphi\colon \mathcal{R} \to \mathbb{A}$ be a categorical attribute of the traces in $L$ – we assume that all traces have this attribute – and let $\delta\colon \mathcal{R} \times \mathcal{R} \to \mathbb{R}$ be a trace distance function, then Algorithm 4 shows the computation.

If $a$ yields an empty domain, the sample is discarded. If all samples are discarded, then the test is undefined. To avoid a bias in the average, a drawn trace is not compared with itself, unless drawn multiple times.

**Algorithm 4** Log vs. Categorical Attribute Test

```
function L-CA TEST(Log $L$, attribute $\varphi$, trace difference $\delta$, repetitions $n$)
    $e \leftarrow 0$
    for $n$ times do
        $L' = (t_1 \ldots t_s) \leftarrow$ random sample of $L$ of size $|L|$ with replacement
        $a \leftarrow \text{avg}[\delta(t_i, t_j) \mid t_i, t_j \in L' \wedge i \neq j \wedge \varphi(t_i) = \varphi(t_j)]$
        $r \leftarrow \text{avg}[\delta(t_i, t_j) \mid t_i, t_j \in L' \wedge i \neq j]$
        if $a < r$ then $e \leftarrow e + 1$ end if
    end for
    if $1 - e/n < \alpha$ then
        reject null hypothesis: at least one value of $\mathcal{A}$ associates with a difference in process
    else
        do not reject null hypothesis
    end if
end function
```

TABLE 3: Overview of associations.

| Associations | numerical trace attribute | categorical trace attribute |
|---|---|---|
| process behaviour (log) | P-NA (5.1) | P-CA (5.2) |
| process conformance (log & model) | C-NA (5.3) | C-NA (5.3) |

For this procedure, Req1 (infinite behaviour support) is irrelevant. The procedure satisfies Req2, and Req3 if $\delta$ considers events in the traces. For $n \to \infty$ and $s \to \infty$, the measure $1 - e/n$ will stabilise (Req4).

## 5 Association Measures

Association measures describe the relationship between the behaviour of traces in a log and other data annotated to the traces. Notice that correlation is a special kind of association (linear). As summarised in Table 3, we introduce association measures to describe the relation between process behaviour or process conformance vs. numerical or categorical trace attributes. As standard process models do not emit trace attributes, only processes described in logs are considered.

### 5.1 Process vs. Numerical Attribute Association

In this section, we introduce an association technique between behaviour in an event log and a numerical trace attribute:

| Process | loan amount |
|---|---|
| $\langle h \rangle$ | 30 000 |
| $\langle h, i \rangle$ | 70 000 |
| $\langle h, i \rangle$ | 100 000 |
| $\langle h, j \rangle$ | 90 000 |
| ... | |

For instance: what is the dependence between the requested loan amount and the process followed? Should a model of the process take the requested loan amount into account? Does a model that includes the loan amount explain significantly more variability than a model that does not include the loan amount?

We aim to measure association. However, as traces have neither an inherent order nor an expected value or mean, we cannot measure association directly. Noting that correlation is covariance normalised with variance, we aim to measure the covariance of two variables $X$ and $Y$, of which $Y$ cannot be numerically observed. Instead, we use the following derivation, which shows that we can also take the covariance of the difference between pairs of measures, in which $X_i$ and $X_j$ denote copies of $X$ (similar for $Y$):

$$\begin{aligned} \text{cov}(\Delta X, \Delta Y) &= \text{cov}(X_i - X_j, Y_i - Y_j) \\ &= \text{cov}(X_i, Y_i) + \text{cov}(X_j, Y_j) \\ &\quad - \text{cov}(X_i, Y_j) - \text{cov}(X_j, Y_i) \\ &= 2\,\text{cov}(X_i, Y_i) \end{aligned} \tag{1}$$

To compute (1), one could take all pairs of traces of the log, and measures the distance between these traces of (1) the chosen numerical trace attribute $\varphi$ and (2) their described process paths using a trace distance function $\delta$. There are $\frac{|L|(|L|-1)}{2}$ of such trace pairs. However, for larger real-life logs this is infeasible, and, as we will show in Section 6.2.4, considering all possible pairs may not be necessary to get a close approximation of the measure.

Therefore, our measure takes a randomly chosen trace pair $n$ times, and from the measured difference tuples, a standard association plot and measure (such as correlation) can be computed. Notice that this procedure assumes that all traces have been annotated with trace attribute $\varphi$; the implementation removes traces without $\varphi$.

Formally, let $L \subseteq \mathcal{R}$ be a collection of traces, let $\varphi\colon \mathcal{R} \to \mathbb{R}$ denote a numerical attribute of the traces in $L$ – we assume that all traces have this attribute – and let $\delta\colon \mathcal{R} \times \mathcal{R} \to \mathbb{R}$ be a trace distance function. Then, Algorithm 5 shows the computation.

**Algorithm 5** Process-Numerical Attribute Association

```
function P-NA ASSOCIATION(traces $L$, attribute $\varphi$, repetitions $n$)
    $m \leftarrow \max_{t \in L} \varphi(t)$
    for $n$ times do
        $t_a \leftarrow$ random trace of $L$
        $t_b \leftarrow$ random trace of $L$
        $d_\varphi = |\varphi(t_a)/m - \varphi(t_b)/m|$
        $d_\delta = \delta(t_a, t_b)$
        record $(d_\varphi, d_\delta)$
    end for
    return plot or correlation measure of recorded tuples
end function
```

Figure 10 shows several examples of correlation density plots, and in Section 6.3.3 we discuss their application to real-life event logs.

Variable $d_\varphi$ is normalised by our method; if $d_\delta$ is chosen to be normalised (e.g. normalised Levenshtein), then all recorded tuples will be $\in [0, 1] \times [0, 1]$, and thus standard correlation measures can be used without further normalisation.

For this measure, as event logs are inherently finite, infinite behaviour is irrelevant and Req1 does not apply. Requirements Req2 and Req3 are  satisfied if the trace distance function $\delta$ satisfies these requirements. For $n \to \infty$, the method converges to the correct value as if all pairs of traces had been taken (Req4).

### 5.2 Process vs. Categorical Attribute Association

In this section, we introduce an association measure between behaviour in an event log and a categorical trace attribute $\varphi\colon \mathcal{T} \to \mathcal{A}$, where $\mathcal{A}$ is the set of possible values of the attribute. For instance: assume that every trace is annotated with the level of customer that the trace represents. Then, what is the relationship between the level of customer (silver, gold, platinum) and process behaviour?

If we consider the traces belonging to each customer level as a separate sub-log, we obtain three stochastic languages:

| | silver | gold | platinum | full log |
|---|---|---|---|---|
| $\langle h \rangle$ | 30 | 20 | 10 | 60 |
| $\langle h, i \rangle$ | 50 | 30 | 4 | 84 |
| $\langle h, j \rangle$ | 15 | 25 | 0 | 40 |
| ... | | | | |

A standard correlation measure such as $\chi^2$ could then be applied as the log, and the set of values of $\varphi$ are finite, however that would not capture the similarity between traces (Req3) and would not be defined if any trace does not appear for an attribute value (Req2).

To the best of our knowledge, association measures for categorical-numerical variable combinations have not been defined, thus a distance-based version like in our P-NA association measure (Section 5.1) would not be possible. Instead, we use a proxy for the desired association measure, by measuring the correlation between the average trace distance with knowledge of $\varphi$ vs. without knowledge of $\varphi$. That is, the average trace distance for traces that have equal $\varphi$ ($a$) vs. the average trace distance for all traces ($r$). The standard correlation between the normalised $a$ and $r$ is 1 when they coincide and 0 if they are unrelated. As we aim to express the opposite – 1 means that $\varphi$ determines the process completely – we reverse the standard correlation by taking 1 - the standard correlation (negative correlations cannot occur).

To obtain a correlation between $a$ and $r$, we apply the bootstrap method: $n$ samples are taken, and for each an $a$ and an $r$ is computed. Special cases such as an empty domain for $a$ are handled in the implementation. As this measure is quadratic in the size of the sample, which for the bootstrap method is the number of traces in the log, we parameterise the resample size $s$.

Formally, let $L \subseteq \mathcal{R}$ be a collection of traces, let $\varphi\colon \mathcal{R} \to \mathcal{A}$ denote a categorical attribute of the traces in $L$ – we assume that all traces have this attribute – and let $\delta\colon \mathcal{R} \times \mathcal{R} \to \mathbb{R}$ be a trace distance function. Then, Algorithm 6 shows the computation.

**Algorithm 6** Process-Categorical Attribute Association

```
function P-CA ASSOCIATION(Process $P$, attribute $\varphi$, trace difference $\delta$, repetitions $n$)
    for $n$ times do
        $L' \leftarrow$ random sample of $P$ of size $|L|$ with replacement
        $a \leftarrow \text{avg}[\delta(t, t') \mid t, t' \in L' \wedge \varphi(t) = \varphi(t')]$
        $r \leftarrow \text{avg}[\delta(t, t') \mid t, t' \in L']$
        report $(a, r)$
    end for
    return plot or 1 - correlation measure of recorded tuples
end function
```

For this association measure, infinite behaviour is irrelevant, as event logs are inherently finite, thus Req1 does not apply. Req2 applies by sampling and Req3 is satisfied if the trace distance function $\delta$ takes events into account. As for Req4, the method stabilises with $n \to \infty$ and $s \to \infty$.

The run time complexity is $O(n|L|\delta)$; if normalised Levenshtein is used for $\delta$, then the run time is $O(nsl^2)$, where $l$ is the maximum trace length.

The P-CA association measures can also be used to describe the association between several process models $M_1 \ldots M_n$, by annotating all behaviour of model $M_i$ with $\varphi$ being $i$.

### 5.3 Conformance vs. Numerical Attribute Association

In this section, we introduce an association measure between the conformance of traces in an event log with respect to a process model and a numerical trace attribute $\varphi \colon \mathcal{T} \to \mathbb{R}$. For instance: assume that every trace is annotated with the claim amount that the trace represents. Then, what is the relationship between the claim amount and the conformance of the trace?

Formally, let $L \subseteq \mathcal{R}$ be a collection of traces, let $\varphi \colon \mathcal{R} \to \mathbb{R}$ denote a numerical attribute of the traces in $L$ – we assume that all traces have this attribute – and let $M \subseteq \mathcal{R}$ be a process model and let $\theta \colon \mathcal{R} \times \mathcal{R}^* \to \mathbb{R}$ be a trace-model distance function. Then, Algorithm 7 shows the computation.

**Algorithm 7** Conformance-Numerical Attribute Association

```
function C-NA ASSOCIATION(log $L$, attribute $\varphi$, model $M$, distance function $\theta$, number of samples $n$)
    for $n$ times do
        $t \leftarrow$ random trace from $L$
        record $(\varphi(t), \theta(t, M))$
    end for
    return plot or association measure of recorded tuples
end function
```

An example of a trace-model distance function $\theta$ is trace-fitness of alignments, which returns a number between 0, indicating the trace does not fit the model at all, and 1, indicating the trace is fully represented by the model [17]. Notice that this measure does not take the stochastic perspective of the model into account.

The C-NA association measure can also be applied to categorical attributes, which will result in a list of numerical-categorical tuples, which can be compared using e.g. a Kruskal-Wallis test.

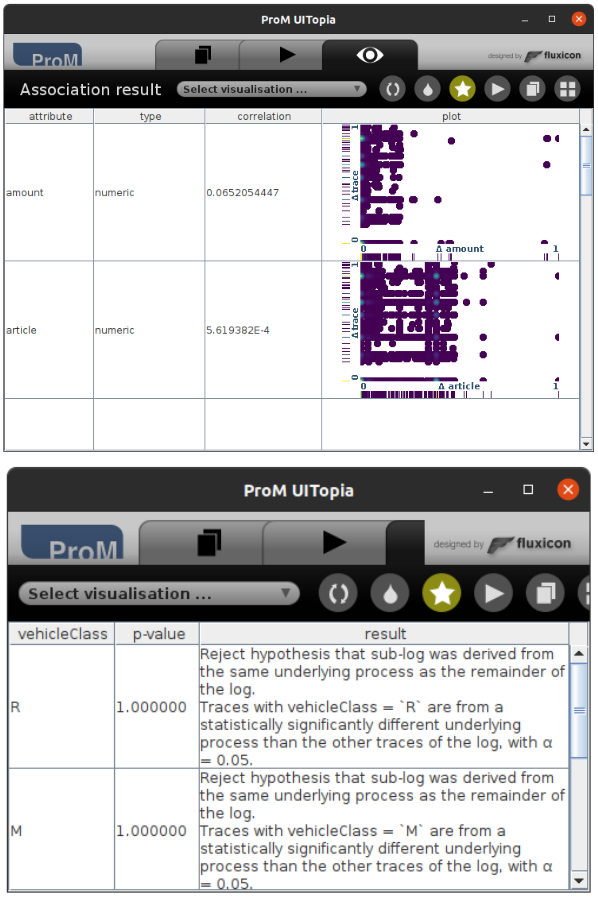
The association could be computed without sampling, as there are a finite number of traces to compute. However, alignments are exponential in the length of the traces, and a close approximation of the association measure might be obtained from a sample.

## 6 Evaluation

In this section, we evaluate the techniques introduced in this paper by (1) describing their implementation, (2) evaluating the influence of parameters, feasibility and validity, and (3) demonstrating their applicability on example applications.

### 6.1 Implementation

The methods introduced in this paper have been implemented as plug-ins of the ProM framework [9][1]. A single plug-in ("Compute association/correlation between the process and trace attributes") computes the association for each appropriate attribute (P-CA & P-NA association) in the log and visualises the results. The statistical tests each have their own plug-in. Additionally, "Log vs. categorical attribute test (pairwise comparison)" applies a P-P-UP test to each pair of sub-logs defined by a categorical attribute value; the results are corrected for the multiple tests being performed using the Benjamini-Hochberg method [18] (see Figure 2).



| attribute | type | correlation | plot |
|---|---|---|---|
| amount | numeric | 0.0652054447 | |
| article | numeric | 5.619382E-4 | |

| vehicleClass | p-value | result |
|---|---|---|
| R | 1.000000 | Reject hypothesis that sub-log was derived from the same underlying process as the remainder of the log.<br>Traces with vehicleClass = `R` are from a statistically significantly different underlying process than the other traces of the log, with α = 0.05. |
| M | 1.000000 | Reject hypothesis that sub-log was derived from the same underlying process as the remainder of the log.<br>Traces with vehicleClass = `M` are from a statistically significantly different underlying process than the other traces of the log, with α = 0.05. |

Fig. 2: Plug-ins in ProM: associations and log-categorical pairwise test.

In these implementations, the sample measures are parallelised, and sampling is sped up using the Alias method [19] and memory techniques: a sample is an array of double-precision numbers representing the likelihood of the trace variants.

As the trace distance function $\delta$, we use the normalised Levenshtein distance, while as the process distance function $\Delta$, we used the Earth Movers' Stochastic Conformance (EMSC) measure [20]. To the best of our knowledge, EMSC is currently the only measure of stochastic behavioural difference that is (1) symmetric, (2) considers trace distance using Levenshtein (Req3), and (3) penalises non-occurring behaviour moderately (Req2). EMSC does not support infinite behaviour; instead, it unfolds loops in process models up to a user-chosen threshold. Still, Req1 is satisfied by the statistical methods in this paper.

### 6.2 Parameters, Feasibility & Validation

In this section, we evaluate the sensitivity of our methods to their parameters, assess their feasibility on real-life logs and validate their results. All experiments were performed once on a range of standard computers (unless indicated otherwise), and thus can only show general feasibility trends.

We used 8 real-life publicly available event logs of the IEEE task force on process mining[2]. For each log, we chose

[1] The source code of the plug-ins is available at https://svn.win.tue.nl/repos/prom/Packages/StatisticalTests.

[2] https://data.4tu.nl/search?q=:keyword: IEEETaskForceonProcessMining

TABLE 4: Adjusted logs.

| Log | Probabilities |
|---|---|
| LL | Log itself: $\forall_{t \in L} LL(t) = L(t)$. |
| TE | Average two least-occurring trace variants: $t$ least-occurring trace, $t'$ second-least-occurring trace, such that $L(t) < L(t')$. Then $TE(t) = \lfloor (L(t) + L(t'))/2 \rfloor$ and $TE(t') = \lceil (L(t) + L(t'))/2 \rceil$. |
| TS | Swap two least-occurring trace variants: $t$ least-occurring trace, $t'$ second-least-occurring trace, such that $L(t) < L(t')$. Then $TS(t) = L(t')$ and $TS(t') = L(t)$. |
| MS | Swap two most-occurring trace variants: $t$ most-occurring trace, $t'$ second-most-occurring trace, such that $L(t) > L(t')$. Then $MS(t) = L(t')$ and $MS(t') = L(t)$. |
| LE | All trace variants having the same probability: $\forall_{t,t' \in LE} LE(t) = LE(t')$. |

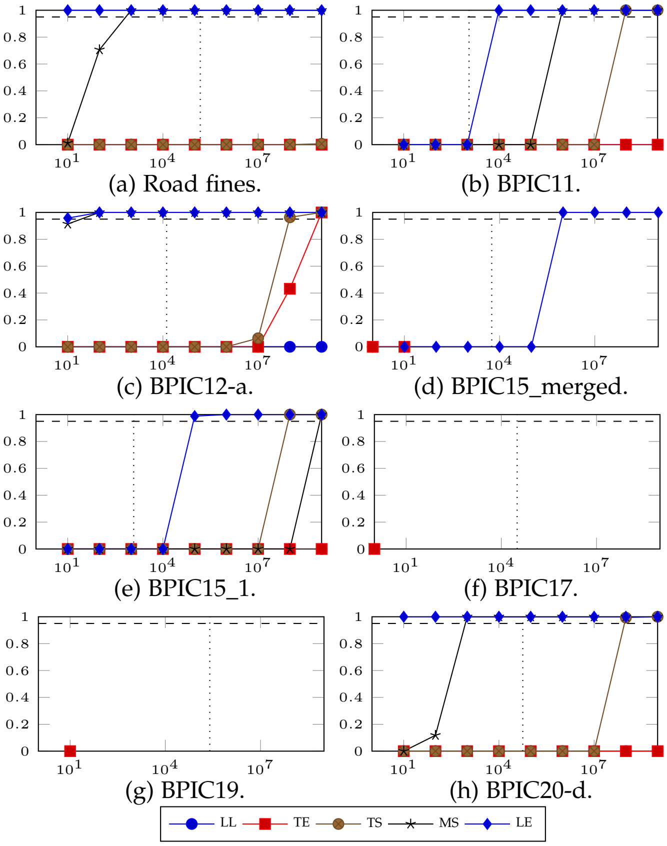
an example numerical attribute as to obtain a mix of continuous, discrete and date typed attributes. Furthermore, for each log that had categorical trace attributes (6/8) we chose such an attribute to likely have an influence on the followed process (just as an analyst would select these using domain knowledge). One of the eight logs (BPIC15_merged) was derived by including all traces from five BPIC15 logs, where an artificial categorical trace attribute indicating the log the trace came from.

### 6.2.1 P-P-UP Test: resample size

For our logs, the resample size of the Process vs. Process - Unknown Process test (P-P-UP, Section 4.1) is fixed by the bootstrap method. Nevertheless, in this section, we make this resample size a parameter ($s$) and evaluate the sensitivity of the P-P-UP test to the resample size. To this end, we apply the test to combinations of real-life event logs and slightly adjusted synthetic copies of these logs, with exponentially increasing resample size $s$. These adjusted logs are summarised in Table 4.

Figure 3 shows the results; BPIC15_merged, BPIC17 and BPIC19 could not be obtained due to the application of EMSC taking more than 1 000 hours on 100 CPUs. Other stochastic language difference functions ($\Delta$) may address this. Intuitively, the test compares the differences of log $L_2$ and resamples of $L_1$ to $L_1$: the test is sensitive to differences that are "larger" than the differences introduced by resampling. Hence, given a large enough resample size, it is sensitive to any difference. The results confirm that very small differences in TS become significant only at $s = 10^7$ for BPIC12-a, even though this log has only 17 trace variants. Other logs show similar patterns: with increasing $s$, first LE becomes significant, followed by MS, TS and TE, as would be expected from the construction of these adjusted logs. For some logs, the $10^9$ samples we used was not sufficient to distinguish all adjusted logs. We observe that this test did not falsely classify equal logs as non-equal, which indicates a very low chance of type-1 errors. We conjecture that type-2 errors can be avoided by taking a large enough sample size, which for some logs might exceed $10^9$.

We conclude that the P-P-UP test is sensitive to the resample size, which confirms the design decision to stick with the standard resample size suggested by the bootstrap method.



(a) Road fines.

(b) BPIC11.

(c) BPIC12-a.

(d) BPIC15_merged.

(e) BPIC15_1.

(f) BPIC17.

(g) BPIC19.

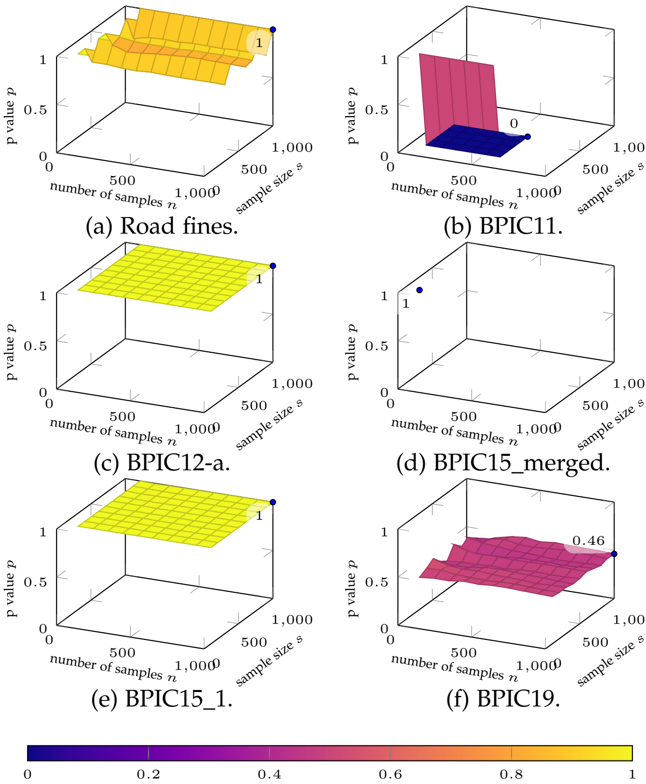
(h) BPIC20-d.

Fig. 3: Sensitivity of the P-P-UP test. Vertical: $p$-value; horizontal: resample size; $n = 10\,000$. Dashed line: $\alpha = 0.05$. Dotted line: number of traces in the log.

### 6.2.2 P-P-L test

In this section, we evaluate the sensitivity of the P-P-L test to its parameters the number of samples ($n$) and the sample size ($s$). We vary both from 100 to 1 000. For the most realistic scenario, we needed quite similar but slightly different stochastic processes, for which we used a process model discovered by the Directly Follows Model Miner [21], with a noise parameter setting of 0.5 and 0.6, annotated to a stochastic model by the Frequency Estimator [22]. As the P-P-L test heavily uses a language-difference function $\Delta$, using EMSC was not an option. Instead, we opted for the uEMSC variant [11], which does not consider differences between traces (i.e. traces are either equivalent or different), which does not satisfy Req3.

The results are shown in Figure 4. Results could not be obtained for BPIC11 (some) and BPIC15_merged (all except one), due to the size of the models: it can require 891 000 steps to traverse the model of BPIC11 before an end state is reached, for instance. Maximum run time varied from 7 seconds (Road fines) to 18 hours (BPIC11). As the critical values of the test are at 0.025 and 0.975 for $\alpha = 0.05$, most of the logs seem stable (even BPIC15_merged), with the exception of BPIC11, which clearly needs a higher sample size $s$ before potentially stabilising (the value at $s = 900$, $n = 900$ is 0.79).



(a) Road fines. (b) BPIC11. (c) BPIC12-a. (d) BPIC15_merged. (e) BPIC15_1. (f) BPIC19.

Fig. 4: Sensitivity of the P-P-L test to its parameters.

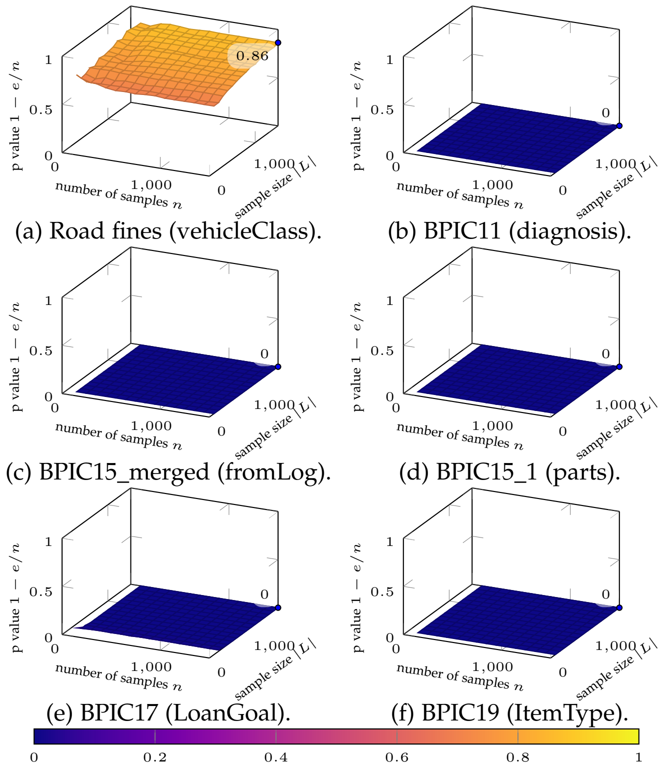
### 6.2.3 L-CA test

The Log-Categorical Attribute test (L-CA test) has one parameter: the number of repetitions $n$. Here, we evaluate the sensitivity of the L-CA test to $n$ by varying $n$ from 100 to 1 500 on our logs that have categorical trace attributes. Furthermore, the L-CA test contains a sampling step of the input log $L$ with size $|L|$, as is the default in the bootstrap method. To test the sensitivity of the L-CA test to the size of the input log, we vary this sampling size similarly.

The results are shown in Figure 5. The Road fines log starst wiggly but stabilises quickly at 0.86, which is well above any typically used $\alpha$, thus the test *suggests* that there is no association between the vehicle class and the followed process. We confirmed this with a domain expert, and performed a manual analysis, which showed that out of the 4 such classes, 3 indeed have largely similar processes. However, the last class (“R”) has a difference in process compared to the other classes: for the R class, the fine is never paid but always sent, while for the other vehicle classes around 33% of the fines is paid and not sent. The L-CA test is insensitive to this difference as there are only 4 traces with with vehicleClass R, out of 150 370 traces in total, due to sampling the test performs. Thus, the L-CA test may be sensitive to the balancedness of the categorical attribute.

For BPIC17, the metric stabilises on 0 at $|L| = 500$, and for the 4 remaining logs, the test metric is never above 0 (in double precision). For these last 5 logs, independence has been disproven for any value of $\alpha$: at least one value of the categorical attribute associates with a difference in process.

The feasibility is illustrated by the maximum run time of a single test on a 10-year old laptop ranging from 12 minutes (Road fines) to around 5 hours (BPIC11).



(a) Road fines (vehicleClass). (b) BPIC11 (diagnosis). (c) BPIC15_merged (fromLog). (d) BPIC15_1 (parts). (e) BPIC17 (LoanGoal). (f) BPIC19 (ItemType).

Fig. 5: Sensitivity of the L-CA test.

### 6.2.4 P-NA association

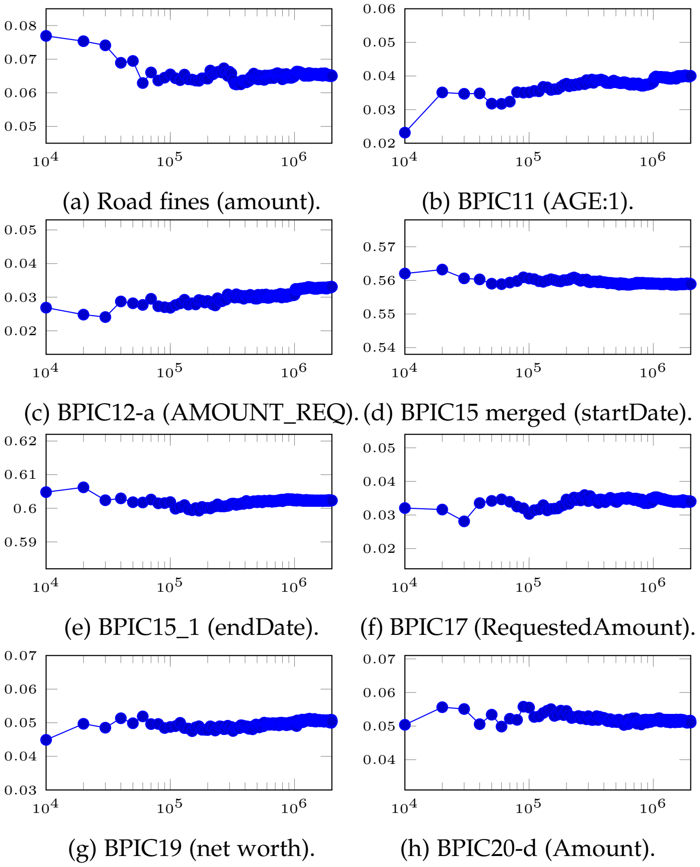
In order to evaluate the sensitivity of the Process vs. Numerical Attribute association measure (P-NA association), we performed an experiment on our real-life logs and their numerical attributes. We varied the number of repetitions $n$ from 10 000 to 1 000 000.

Figure 6 shows the results. It is clear that the number of samples does not influence the result much: the Road fines log, having 150 370 traces, stabilises last, but stabilises on two decimals at just 40 000 samples. Nevertheless, sampling is very fast and, after log loading, 1 000 000 samples were obtained in at most 25 seconds (BPIC11), but typically in less than a second (BPIC12, Road fines) on a standard laptop. Anecdotally: filtering the traces not having the numerical attribute was the most time-consuming step. Thus, we argue that an $n$ of 1 000 000 suffices for the real-life logs considered here, thus we recommend this $n$ for standard-sized event logs. If more certainty is necessary, the experiment could be repeated a few times.

### 6.2.5 P-CA association

The P-CA association measure has one parameter: the number of repetitions $n$; we test the sensitivity by varying $n$ from 100 to 1 500 on our real-life logs. Furthermore, we similarly evaluate the sensitivity to the log size by adjusting the sampling size ($|L|$ in Algorithm 6).

Figure 7 shows the results. We observe that for all logs, it is necessary to take a sufficient number of samples $n$, and that the log needs to be large enough. However, the logs can reliably be ranked even with a low $n$. We conclude that sampling sizes equal to the number of traces in the log, as suggested by the bootstrap method, seem not to be necessary for logs of these sizes and complexities.



(a) Road fines (amount).

(b) BPIC11 (AGE:1).

(c) BPIC12-a (AMOUNT_REQ).

(d) BPIC15 merged (startDate).

(e) BPIC15_1 (endDate).

(f) BPIC17 (RequestedAmount).

(g) BPIC19 (net worth).

(h) BPIC20-d (Amount).

Fig. 6: Sensitivity of P-NA association (y-axes) to number of samples (x-axes). The y-axes have been translated.

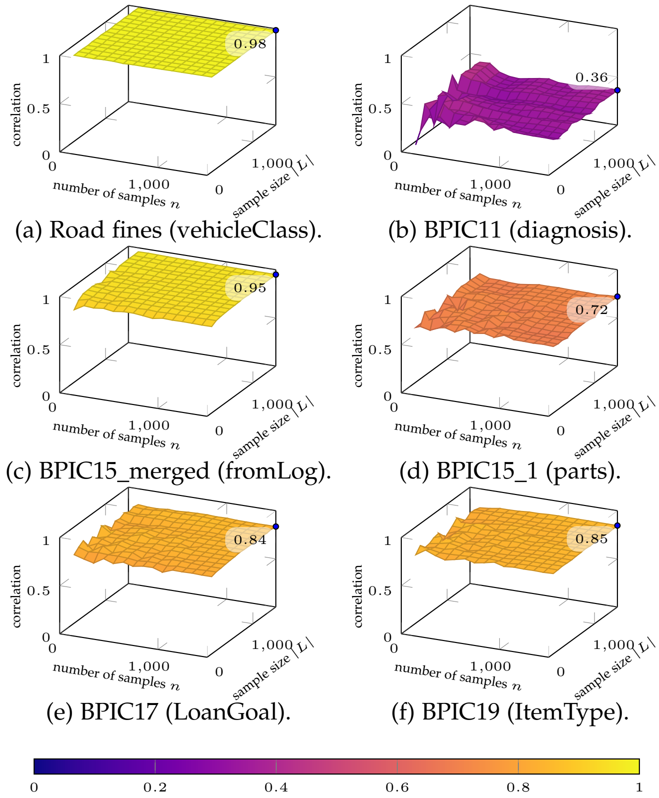
Maximum run time for a single correlation computation ranged from less than 8 minutes (Road fines) to almost 3 hours (BPIC11).

### 6.2.6 C-NA association

The C-NA association measure has one parameter: the number of samples $n$. To test the sensitivity of this measure to its parameter, we instantiated the measure with the alignments conformance measure [17]. We applied the instantiated C-NA measure to our 8 logs with a numeric or time attribute, while varying the number of samples from 1 000 to 10 000. The conformance of these logs was taken with respect to a model discovered by the Directly Follows Model Miner [21] for each log, with a noise threshold of 0.5 chosen to introduce some non-conformance. The results, summarised by Pearson correlation, are shown in Figure 8.

Notice that the C-NA association measure is expensive, as for each sample an optimal alignment [17] needs to be computed, which may be infeasible for models with hundreds of activities. Results for BPIC15_1 and BPIC15_merged could not be obtained due to this; maximum run time ranged from 4 minutes (BPIC20) to 9 hours (BPIC11), The C-NA association measure allows the leverage of other conformance measures, which may alleviate this problem.

We conclude that the sensitivity on logs of complexity and size comparable to used here is sufficient to rank the logs reliably (variation is in the order of 0.02), but larger sample sizes may be necessary for more precision.



(a) Road fines (vehicleClass).

(b) BPIC11 (diagnosis).

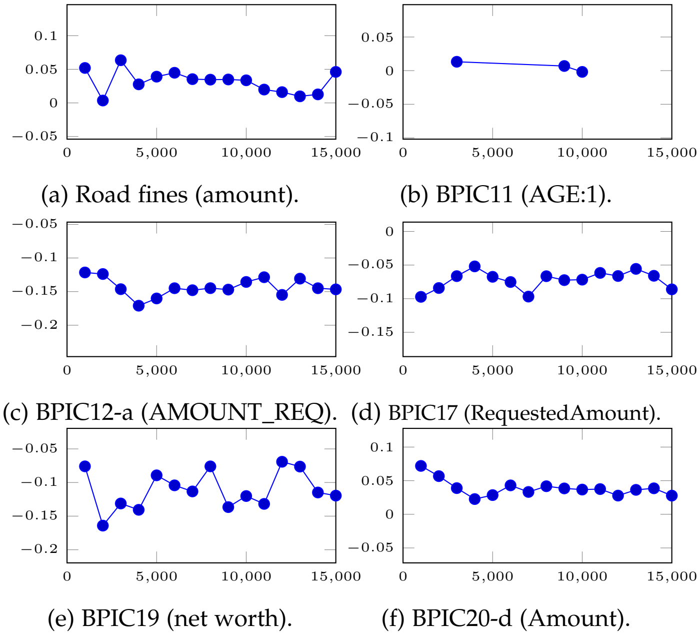
(c) BPIC15_merged (fromLog).

(d) BPIC15_1 (parts).

(e) BPIC17 (LoanGoal).

(f) BPIC19 (ItemType).

Fig. 7: Sensitivity of P-CA association to its parameters.



(a) Road fines (amount).

(b) BPIC11 (AGE:1).

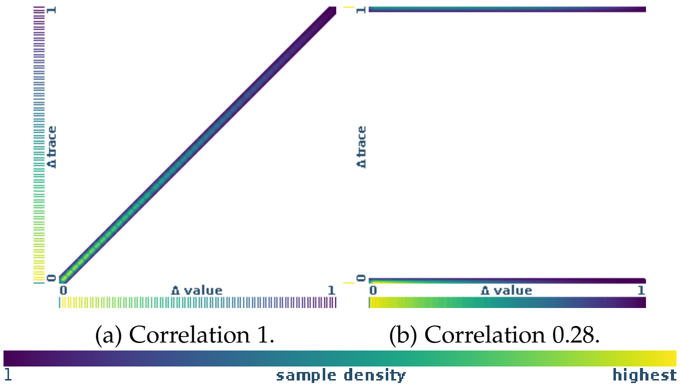
(c) BPIC12-a (AMOUNT_REQ).

(d) BPIC17 (RequestedAmount).

(e) BPIC19 (net worth).

(f) BPIC20-d (Amount).

Fig. 8: Sensitivity of C-NA association (y-axes) to number of samples (x-axes). Z-axes are translated.



(a) Correlation 1. (b) Correlation 0.28.

Fig. 9: Example of P-NA association on synthetic models with $n = 1\,000\,000$.

### 6.3 Example Applications

In this section, we show several applications of the techniques presented in this paper. We first highlight their properties in a controlled setting, after which we discuss an example application of tests, followed by an example application of association measures.

#### 6.3.1 Controlled Experiments

First, we illustrate the sensitivity of our association measure to two event logs: (1) an event log of 100 000 traces derived from a model $[\langle a_1 \dots a_x, b_{x+1}, \dots b_{100}\rangle^{\text{value}:x} \mid x \sim U(0, 100)]$; and (2) an event log of 100 000 traces derived from a model that expresses for a given annotated $x$, trace $\langle a \rangle$ has a probability of $x$ to be included, and trace $\langle b \rangle$ has a probability of $1 - x$ to be included: $[\begin{cases} \langle a \rangle^{\text{value}:x} & \text{if } p \leq x \\ \langle b \rangle^{\text{value}:x} & \text{otherwise} \end{cases} \mid x \sim U(0,1) \wedge p \sim U(0,1)]$. To these logs, our P-NA association measure was applied.

The results are shown in Figure 9, by means of association density plots: the horizontal axis is the difference in value ($x$), and the vertical axis is the difference in Levenshtein distance. Each sample is a dot, and where more dots overlap, the colour of the plot ranges from purple to yellow, as indicated in the sample density legend.

For the first event log, Figure 9a shows a straight line, indicating a correlation of 1, which was to be expected from the model: for a higher difference in $x$, there is a corresponding higher difference in Levenshtein, as the higher the $x$, the more $b$s get replaced by $a$s. The colouring of the association density plot shows that the sampling favoured closer traces.

For the second event log, Figure 9b shows two horizontal lines, one at 0 and one at 1, yielding a low correlation. This shows a limitation of the technique: the test uses trace distance (i.e. Levenshtein) and in this toy example, the difference between two traces is either 0 or 1. A measure using a process distance (e.g. EMSC) rather than a trace distance (e.g. Levenshtein) might give a smoother and more useful visualisation, at the cost of more computations.

#### 6.3.2 Comparing Executions of the Same Process

In [5], a case study was described in which participants from 5 municipalities discussed several differences in process, with the aim of finding commonalities for process standardisation. We replicate this setting using a log derived from an Italian road traffic fine management process. Similar to [5], one could imagine that a workshop is organised for process participants to attempt to standardise processes. There are 26 dismissal codes in this log, and we are interested if there is any difference in the process followed for each dismissal code. In this section, we perform an analysis that can be performed before stakeholders get involved, by establishing which dismissal reasons coincide with the most similar processes, which can be used to focus workshop efforts.

**Hypothesis**: *All dismissals follow the same underlying (unknown) process.*

To test this hypothesis, we apply the L-CA test (Section 4.3) with $n = 10\,000$ repetitions. The test rejects the hypothesis ($p = 0.0$, $\alpha = 0.05$), thus we conclude that not all sub-logs were derived from the same process. Next, we perform a follow-up test to reveal which processes are different from one another.

**Hypothesis**: *Each pair of dismissals was derived from the same underlying (unknown) process.*

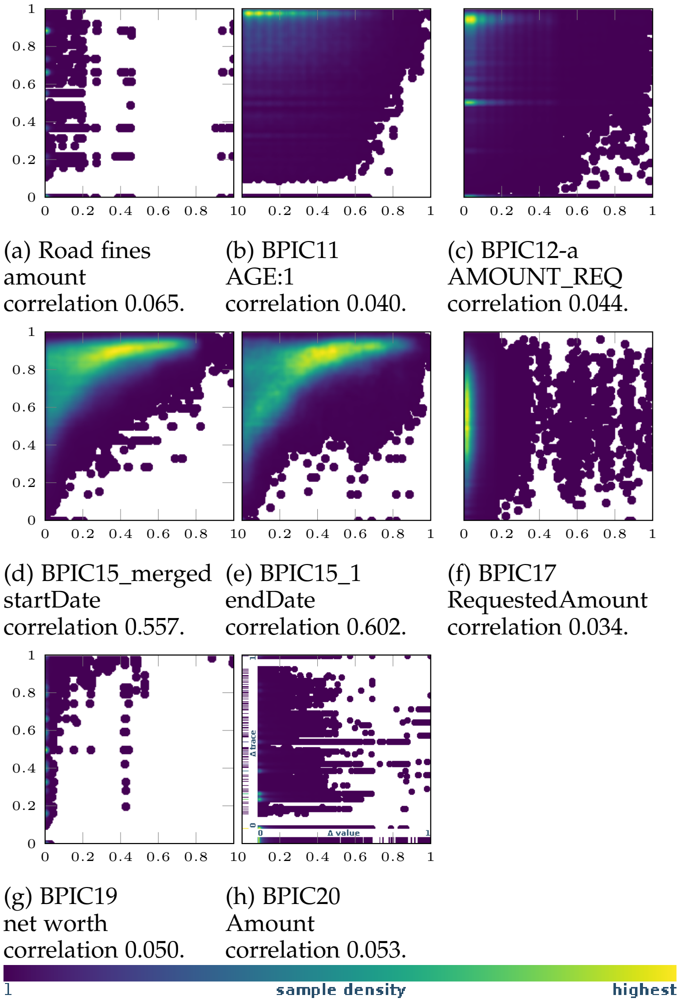
To test this hypothesis, we perform a series of P-P-UP tests (Section 4.1), each comparing a pair of dismissals' logs. As we are performing 325 tests, there is an increased risk of false positives [23]. To correct for this, we apply the Benjamini-Hochberg method [18], which sorts the $p$-values in reverse order, and rejects the hypotheses up to the last $p$-value for which it holds that $1 - p < \alpha r / x$, where $r$ is the rank of the $p$-value and $x$ is the number of tests (here: 325). We reject the hypothesis for all pairs of municipalities, except for 14 pairs, ranging from $p = 0.9495$ to $p = 0.7863$. The non-rejected pairs, for which there is not enough evidence to conclude that their processes are different, form interesting patterns. For instance, the pair of dismissal of category "D" and category "C" is not rejected, "C" and "Z" is not rejected, but "D" and "Z" is rejected. Thus, the approximate equality of processes is, sensibly, not transitive.

In a workshop setting, it would be infeasible to consider all 325 pairs of 26 processes to find pairs that could, for instance, share a single information system and thus save on development effort. Where existing process-based techniques (e.g. EMSC [20]) could be used to compare the stochastic behaviour of the pairs and pairs could be ranked accordingly, there would be no guidance on which and how many pairs to consider: an arbitrary cut-off threshold would need to be chosen. Furthermore, using the tests introduced in this paper, as shown, existing statistical methods can be leveraged to avoid standard statistical pitfalls.

#### 6.3.3 Applying Association Measures

To illustrate the association measures, in this section we apply them to several publicly available real-life event logs.

For instance, Figure 10a shows the recorded tuples for a log of a road fine collection process. The horizontal axis shows the difference in the amount of the fine, while the vertical axis shows the difference in process ($\delta$). From the correlation measure and Figure 10a, it is clear that the road fines collection process is not associated with the fine amount: the correlation is very low and for similar fine amounts large differences in process exist. The plot has several horizontal lines, which indicates that the number of trace variants is low (231). Similarly, Figures 10c and 10b, and their measure of 0.04 shows a weak association.



(a) Road fines amount correlation 0.065.

(b) BPIC11 AGE:1 correlation 0.040.

(c) BPIC12-a AMOUNT_REQ correlation 0.044.

(d) BPIC15_merged startDate correlation 0.557.

(e) BPIC15_1 endDate correlation 0.602.

(f) BPIC17 RequestedAmount correlation 0.034.

(g) BPIC19 net worth correlation 0.050.

(h) BPIC20 Amount correlation 0.053.

Fig. 10: P-NA association plots of real-life logs ($n = 1\,000\,000$). x-axes: difference in trace; y-axes: difference in attribute.

A strong association can be seen in the BPIC15_1 log and the merged BPIC15 log: the measure of 0.557 indicates an association between the starting date of the process instances and the process, while the plots (figures 10d and 10e) show a similar clear pattern of a changing process: we have found that this log contains process drift. Note that these logs challenges current process mining techniques: (1) as the log contains 500 activities, process models are incomprehensible for analysts; and (2) due to the large number of unique patient-pathways [5 541 on a total of 5 649 traces] and the high number of events per trace [average 46, maximum 154], its state space is very large. Thus, the information provided by the correlation plot might provide a quick starting point to filter the event log to ease further analysis: the correlation measure takes less than 30 seconds for all samples combined, as the Levenshtein distance is quadratic in the length of the traces. Consequently, these filtered logs contain less process-based variability and thus might be easier to study using existing process mining methods.

For the log-attribute combinations with a low correlation, it can be concluded that the followed process does not associate with the used attribute. For instance, the amount requested in BPIC12-a and BPIC17 seems to not have any relation with the followed process. Such insights may steer ongoing analysis efforts: it makes little sense to split the log based on these variables. Furthermore, showing the absence of differences in process based on gender, postal code or nationality may also be of value for organisations. Existing process mining techniques (e.g. [6]) would be able to rank attributes based on process distance, but would not provide statistically grounded quantifications of these differences.

## 6.4 Discussion

With the introduction of statistically sound measures and tests, the sampling method becomes important. In process mining projects, typically all cases starting and ending in a particular time interval are extracted from an information system and used as the event log. In the best case, this can be considered a complete sample of the actual behaviour that happened. However, it is not a complete sample of all *potential* behaviour of the underlying process. Other factors that may influence the sample quality are data quality repairs and data cleaning efforts. We acknowledge existing preliminary work on sampling in the process mining field [24], [25], [26], but leave a detailed discussion of sampling quality for future work.

Next, we discuss the particulars of the tests and association measures introduced in this paper.

### *6.4.1 Tests*

In the previous section, we have shown that the association measures are sensitive to the resample size $s$. That is, $s$ must be chosen carefully: if chosen too low, the tested behaviour is not captured well, while if chosen too large, the answer tends to determinism. Thus, it might be possible to select a resample size $s$ that will lead to a desired answer to a statistical test. This should be avoided, and it is important to choose a reasonable resample size beforehand, just as with the statistical threshold $\alpha$.

A limitation of the statistical tests is that they are not sensitive to the number of traces in the log (the resample size). Intuitively, we can be more sure using a log of $10^6$ traces than on a log with $10^2$ traces, but this is not reflected in any of the tests introduced in this paper.

The run-time of most statistical tests is determined by the expensive process-distance computation ($\Delta$); only the L-CA test depends on the cheaper trace-trace comparison ($\delta$). For larger samples ($s$), this might become infeasible, though for the experiments reported in this paper, the test ran in 8 hours or less for $s = 1\,000\,000\,000$.

### *6.4.2 Association Measures*

In the previous section, we have shown that the association measures are not very sensitive to the number of samples $n$: getting enough samples ($n$) suffices to obtain a reliable answer, and more samples are better.

Association measures are not sensitive to the number of traces in the event log, which we argue is desirable.

The run time of the association measures is low: in our experiments, at most 3 seconds were spent on any measure, as all are based on the quick trace distance measure $\delta$.

For numerical attributes, we argue that while the measure of association is useful for comparison, the plots are much more informative, as they show a much more detailed view of the relationship between process behaviour and attribute value. Nevertheless, it still holds that association does not necessarily imply causation.

## 7 Related Work

Recently, the process mining community intensified research of methods for identifying the significance, hence reproducibility, of inferred insights about the analysed processes. It was observed that existing process discovery techniques often do not guarantee that better quality input event logs result in better discovered models [24], [1]. Consequently, it was suggested that process discovery techniques should be accompanied by formal proofs or empirically established statistical results that justify the quality of produced models. Alkhammash et al. [27] used bootstrapping approach to quantify the risks of making wrong conclusions about process models automatically discovered from event logs. The statistical tests presented in this paper support the development of process mining techniques capable of reproducing their results.

Business process comparison and process variant analysis [28] study commonalities and discrepancies between processes and support the reuse and standardisation of processes. Such analysis is more reliable if grounded in statistically significant differences between the processes. The existence or absence of such differences can be justified using tests presented in this article.

It is possible to improve the efficiency of process discovery by constructing models from a subset of (a sample), rather than from the entire, event log. Several works confirm that different strategies for choosing the sample may speed up the construction and improve the quality of the discovered process models [25], [29]. A separate line of research studies ways to use statistics to estimate the completeness of event logs [30], [31]. If the log completeness is established, it can be related to the quality of the process mining results. Some of these methods for establishing the log completeness use sampling methods, which can be explored to instantiate our statistical tests.

Several works in process mining explore relationships between processes and their attributes or outcomes. For instance, [6] provides the trace attributes that characterise the highest distance in the processes; [32] can be used to discover collections of traces for which a controlled intervention has a high causal effect on the outcome of the process; in [33], statistical tests for identifying effects of different treatment sequences of patients were studied (the identification of different patient cohorts was supported by process mining techniques, while the difference in effects for the cohorts was identified using statistical tests over trace attributes, and not over the process behaviour); and [34] studies ways to identify statistically significant differences in the control-flow and activity duration of business process variants. The techniques presented in this paper could provide a statistical foundation to the aforementioned techniques, e.g. our association measures can be used to identify relationships between the ways patients were treated vs. the attributes of patients or treatment outcomes.

Existing works differ from the methods introduced in this paper as we allow quantifying uncertainty when analysing process data; providing a means to assess statistical significance and draw inference on processes.

## 8 Conclusion

In many fields of research, the use of statistical tests and association measures is omnipresent. However, in process mining, not a single method giving a statistical quantification of uncertainty has been proposed, that is, a method to establish statistical significance over process behaviour. In this paper, we formulated requirements for such methods, and introduced several statistical tests to compare (i) 2 processes or (ii) multiple processes, with either (a) an unknown process or (b) a known process. Furthermore, we introduced measures expressing the association between (i) a log or (ii) the conformance of a log to a model, with either (a) a categorical or (b) a numerical trace attribute. We have evaluated the sensitivity of the introduced methods to their parameters, and illustrated how they could be applied in practice.

An interesting area of future work is to establish the association of process and start time of traces as a means of concept drift detection. Cohort analysis studies the influence of the combination of trace variables on process behaviour [6]; it would be interesting to provide cohort analysis with a statistical foundation using the methods introduced in this paper. Finally, all techniques described in this paper do not consider concurrency, even though it could be argued that concurrency simplifies the stochastic perspective: if we know that $a$ and $b$ are concurrent in two traces $\langle a, b\rangle$ and $\langle b, a\rangle$, then these traces are equal and there is no need for a stochastic perspective to distinguish them. The study of the impact of concurrency on statistical methods is a subject of future work.

*Acknowledgement.* The research described in this paper was partially supported by the QUT Centre for Data Science.

## References

[1] W. M. P. van der Aalst, T. Weijters, and L. Maruster, “Workflow mining: Discovering process models from event logs,” *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 9, pp. 1128–1142, 2004.

[2] C. Fernández-Llatas *et al.*, “Analyzing medical emergency processes with process mining: The stroke case,” in *Business Process Management Workshops*, ser. LNBIP, vol. 342. Springer, 2018, pp. 214–225.

[3] A. Polyvyanyy, A. Moffat, and L. García-Bañuelos, “An entropic relevance measure for stochastic conformance checking in process mining,” in *International Conference on Process Mining*. IEEE, 2020, pp. 97–104.

[4] A. Augusto, R. Conforti, M. Dumas, M. L. 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.*, vol. 31, no. 4, pp. 686–705, 2019.

[5] J. C. A. M. Buijs and H. A. Reijers, “Comparing business process variants using models and event logs,” in *Enterprise, Business-Process and Information Systems Modeling Conference*, ser. LNBIP, vol. 175. Springer, 2014, pp. 154–168.

[6] S. J. J. Leemans, S. Shabaninejad, K. Goel, H. Khosravi, S. W. Sadiq, and M. T. Wynn, “Identifying cohorts: Recommending drill-downs based on differences in behaviour for process mining,” in *Conceptual Modeling Conference*, ser. LNCS, vol. 12400. Springer, 2020, pp. 92–102.

[7] R. P. J. C. Bose, W. M. P. van der Aalst, I. Zliobaite, and M. Pechenizkiy, “Dealing with concept drifts in process mining,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 25, no. 1, pp. 154–171, 2014.

[8] T. Brockhoff, M. S. Uysal, and W. M. P. van der Aalst, “Time-aware concept drift detection using the earth mover’s distance,” in *International Conference on Process Mining*. IEEE, 2020, pp. 33–40.

[9] B. F. van Dongen, A. K. A. de Medeiros, H. M. W. Verbeek, A. J. M. M. Weijters, and W. M. P. van der Aalst, “The prom framework: A new era in process mining tool support,” in *Applications and*

*Theory of Petri Nets Conference*, ser. LNCS, vol. 3536. Springer, 2005, pp. 444–454.

[10] V. Levenshtein, “Binary Codes Capable of Correcting Deletions, Insertions, and Reversals,” *Soviet Physics-Doklady*, vol. 10, no. 8, pp. 707–710, 1966.

[11] S. J. J. Leemans, A. F. Syring, and W. M. P. van der Aalst, “Earth movers’ stochastic conformance checking,” in *Business Process Management Forum*, ser. LNBIP, vol. 360. Springer, 2019, pp. 127–143.

[12] B. Efron, “Bootstrap Methods: Another Look at the Jackknife,” *The Annals of Statistics*, vol. 7, no. 1, pp. 1 – 26, 1979.

[13] K. Pearson, “X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling,” *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 50, no. 302, pp. 157–175, 1900.

[14] P. Marjoram, J. Molitor, V. Plagnol, and S. Tavaré, “Markov chain Monte Carlo without likelihoods,” *Proceedings of the National Academy of Sciences*, vol. 100, no. 26, pp. 15 324–15 328, 2003.

[15] A. Rogge-Solti, W. M. P. van der Aalst, and M. Weske, “Discovering stochastic petri nets with arbitrary delay distributions from event logs,” in *Business Process Management Workshops*, ser. LNBIP, vol. 171. Springer, 2013, pp. 15–27.

[16] S. J. J. Leemans and A. Polyvyanyy, “Stochastic-aware conformance checking: An entropy-based approach,” in *Advanced Information Systems Engineering Conference*, ser. LNCS, vol. 12127. Springer, 2020, pp. 217–233.

[17] W. M. P. van der Aalst, A. Adriansyah, and B. F. van Dongen, “Replaying history on process models for conformance checking and performance analysis,” *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 182–192, 2012.

[18] Y. Benjamini and Y. Hochberg, “Controlling the false discovery rate: A practical and powerful approach to multiple testing,” *Journal of the Royal Statistical Society. Series B, Methodological*, vol. 57, no. 1, pp. 289–300, 1995.

[19] A. J. Walker, “An efficient method for generating discrete random variables with general distributions,” *ACM Trans. Math. Softw.*, vol. 3, no. 3, pp. 253–256, 1977.

[20] S. J. J. Leemans, W. M. P. van der Aalst, T. Brockhoff, and A. Polyvyanyy, “Stochastic process mining: Earth movers’ stochastic conformance,” *Inf. Syst.*, vol. to appear, p. 101687, 2021.

[21] S. J. J. Leemans, E. Poppe, and M. T. Wynn, “Directly follows-based process mining: Exploration & a case study,” in *International Conference on Process Mining*. IEEE, 2019, pp. 25–32.

[22] A. Burke, S. J. J. Leemans, and M. T. Wynn, “Stochastic process discovery by weight estimation,” in *Process Mining workshops*, ser. LNBIP, S. J. J. Leemans and H. Leopold, Eds., vol. 406. Springer, 2020, pp. 260–272.

[23] M. Jafari and N. Ansari-Pour, “Why, when and how to adjust your p values?” *Cell Journal (Yakhteh)*, vol. 20, no. 4, p. 604, 2019.

[24] J. M. E. M. van der Werf, A. Polyvyanyy, B. R. van Wensveen, M. J. S. Brinkhuis, and H. A. Reijers, “All that glitters is not gold – towards process discovery techniques with guarantees,” in *CAiSE*, ser. LNCS, vol. 12751. Springer, 2021, pp. 141–157.

[25] M. F. Sani, S. J. van Zelst, and W. M. P. van der Aalst, “The impact of event log subset selection on the performance of process discovery algorithms,” in *New Trends in Databases and Information Systems*, ser. Communications in Computer and Information Science, vol. 1064. Springer, 2019, pp. 391–404.

[26] B. R. van Wensveen, “Estimation and analysis of the quality of event log samples for process discovery,” Master’s thesis, Utrecht University, 2020.

[27] H. Alkhammash, A. Polyvyanyy, A. Moffat, and L. García-Bañuelos, “Entropic relevance: A mechanism for measuring stochastic process models discovered from event data,” *Information Systems*, p. 101922, 2021.

[28] A. Syamsiyah, A. Bolt, L. Cheng, B. F. A. Hompes, R. P. J. C. Bose, B. F. van Dongen, and W. M. P. van der Aalst, “Business process comparison: A methodology and case study,” in *Business Information Systems Conference*, ser. LNBIP, vol. 288. Springer, 2017, pp. 253–267.

[29] C. Liu, Y. Pei, L. Cheng, Q. Zeng, and H. Duan, “Sampling business process event logs using graph-based ranking model,” *Concurr. Comput. Pract. Exp.*, vol. 33, no. 5, 2021.

[30] J. Pei, L. Wen, H. Yang, J. Wang, and X. Ye, “Estimating global completeness of event logs: A comparative study,” *IEEE Trans. Serv. Comput.*, vol. 14, no. 2, pp. 441–457, 2021.

[31] C. Li, J. Ge, L. Wen, L. Kong, V. Chang, L. Huang, and B. Luo, “A novel completeness definition of event logs and corresponding generation algorithm,” *Expert Syst. J. Knowl. Eng.*, vol. 37, no. 4, 2020.

[32] Z. D. Bozorgi, I. Teinemaa, M. Dumas, M. L. Rosa, and A. Polyvyanyy, “Process mining meets causal machine learning: Discovering causal rules from event logs,” in *International Conference on Process Mining*. IEEE, 2020, pp. 129–136.

[33] E. Tavazzi, C. L. Gerard, O. Michielin, A. Wicky, R. Gatta, and M. A. Cuendet, “A process mining approach to statistical analysis: Application to a real-world advanced melanoma dataset,” in *Process Mining Workshops*, ser. LNBIP, vol. 406. Springer, 2020, pp. 291–304.

[34] F. Taymouri, M. L. Rosa, and J. Carmona, “Business process variant analysis based on mutual fingerprints of event logs,” in *Advanced Information Systems Engineering Conference*, ser. LNCS, vol. 12127. Springer, 2020, pp. 299–318.

**Sander J.J. Leemans** is a professor at RWTH university, Aachen, Germany. His research interests include process mining, process discovery, conformance checking, stochastic process mining, and business process management. In particular, he specialises in making solid academic techniques available to end-users, analysts and industry partners.

**James McGree** is an applied and computational statistician in the School of Mathematical Sciences at the Queensland University of Technology (Australia). He received his PhD in statistics from the University of Queensland (Australia) in August 2008. The main focus of his research is the development of new methods in design of experiments, Bayesian computational algorithms and big data analytics, and has applied such methods across the medical, ecological and biological sciences.

**Artem Polyvyanyy** Dr. Artem Polyvyanyy is a senior lecturer at the School of Computing and Information Systems, Faculty of Engineering and Information Technology, at the University of Melbourne, Australia. His research and teaching interests include Computing Systems, Information Systems, Distributed Systems, Process Modeling and Analysis, Process Mining, Process Querying, and Algorithms. Artem is a member of the Steering Committee of the IEEE Task Force on Process Mining.

**Arthur H.M. ter Hofstede** received his PhD degree from the Katholieke Universiteit Nijmegen (since renamed to Radboud Universiteit), Nijmegen, The Netherlands, in 1993. His research interests lie in the area of business process management, in particular business process automation and process mining. He was involved in the well-known workflow patterns initiative (www.workflowpatterns.com) and at QUT he has managed the well-known YAWL initiative (www.yawlfoundation.org).