The Jensen-Shannon Distance for Stochastic Conformance Checking

Tian Li[1,2], Sander J.J. Leemans[1,3], and Artem Polyvyanyy[2]

[1] RWTH Aachen University, Germany
t.li,s.leemans@bpm.rwth-aachen.de
[2] The University of Melbourne, Australia
artem.polyvyanyy@unimelb.edu.au
[3] Fraunhofer FIT, Germany

Abstract. A sub-field of process mining, conformance checking, quantifies how well the process behavior of a model represents the observed behavior recorded in a log. A stochastic-aware perspective that accounts for the probability of behavior in both model and log is necessary to support conformance checking. However, existing stochastic conformance checking measures are not comparable for a broad framework that includes log-to-log (L2L), log-to-model (L2M), and model-to-model (M2M) comparison settings. Therefore, we propose a stochastic conformance checking measure based on the Jensen-Shannon Distance (JSD), which interprets models and logs as probability distributions over traces. It can be applied to perform L2L, L2M, and M2M conformance, while the latter requires approximation. Notably, it is the only known stochastic conformance measure that is a metric. JSD has been implemented and is publicly available. Our quantitative evaluations show the feasibility of computing JSD over real-life event logs, and that it provides diagnostic results different from those of existing measures. Moreover, experiments in the M2M setting confirm that our measure can be approximated using unbiased sampling.

Keywords: Stochastic process mining, Stochastic conformance checking

1 Introduction

Information systems in modern organizations record process executions performed by employees, managers, and customers as event data. Such data can be extracted as an event log, which is a collection of recorded traces, where each trace is a sequence of activities observed in a process execution. By leveraging the historical event data in event logs, process mining studies ways to optimize real-world processes [1].

In process mining, process models are used to better understand the process and identify issues. *Conformance checking* relates events in the event log to activities in the process model to identify commonalities and differences between them, i.e., log-to-model comparison (L2M). For example, the results of L2M

Postprint, November 2024

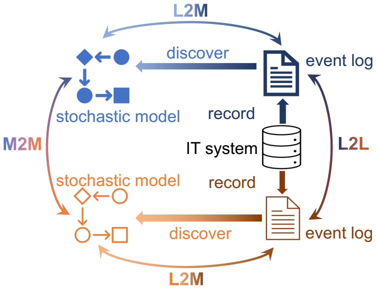


Fig. 1: Three settings of stochastic conformance checking: L2L, L2M, and M2M.

conformance checking can be used to inform auditing efforts. Additionally, conformance checking may involve model-to-model comparisons (M2M) [1, p265], for instance, to compare models discovered from event logs recorded in geographically different regions. Furthermore, log-to-log conformance checking (L2L) compares two logs directly with one another. L2L conformance checking can be used for detecting process drift [28], which refers to the problem of finding changes in a process over time.

In real-life processes, certain behavior can occur more frequently than the other. Consider two event logs $[\langle a,b\rangle^{50}, \langle b,a\rangle^{50}]$ and $[\langle a,b\rangle^{80}, \langle b,a\rangle^{20}]$, which have the same trace variants but differ in the frequency of observations of these variants. As these logs are different, this should be reflected in conformance checking measurements. As for M2M conformance checking, one can detect and quantify changes in stochastic behavior by comparing the latest discovered model with a previous version of the model. Similarly, event logs that cover long periods or merge data from multiple organizations may contain different versions of process behavior. By applying stochastic conformance for L2L settings, one can avoid misleading conclusions when addressing process drift. We illustrate these settings of stochastic conformance checking in Fig. 1.

None of the existing stochastic conformance checking techniques support all of L2L, L2M and M2M, or the measure results in values that are incomparable across the three settings [20,22,25]. In this paper, we propose a stochastic conformance checking metric based on the Jensen-Shannon Distance (JSD) [13]. This metric interprets process behavior in an event log or a stochastic process model as a probability distribution over traces. JSD can be applied for stochastic conformance checking across three settings: i) L2L setting, ii) L2M setting where the stochastic model has a finite state space, and iii) M2M setting that relies on unbiased sampling.

The JSD metric has been implemented and is publicly available [4]. We compared it quantitatively with existing stochastic conformance techniques on several real-life event logs and stochastic process models, which demonstrated that JSD measurements can lead to different conclusions. Moreover, for the M2M

[4] https://bpm.rwth-aachen.de/ebi

setting, we evaluated the influence of the sample size and confirmed that the approximation converges with the growth of simulated event logs.

The remainder of the paper proceeds as follows. We first discuss related work in Section 2 and introduce preliminaries in Section 3. In Section 4, we discuss how JSD can be applied in L2L, L2M, and M2M settings, after which we evaluate it in Section 5. Finally, Section 6 concludes the paper.

2 Related Work

Recently, several techniques for stochastic process discovery have been proposed, including the weight estimation techniques that discover a stochastic labeled Petri net (SLPN) from the input event log and control flow model [7,18,6], and techniques that directly construct a stochastic model from an input event log [26,3].

Conformance checking for non-stochastic models has been extensively discussed [8]. Van der Aalst [2] emphasized the importance of considering probabilities in conformance checking. Entropic Relevance (ER) [25] computes the average number of bits to compress each log trace by leveraging the trace likelihood information in a stochastic model. Entropy Recall (E-Recall) and Entropy Precision (E-Precision) [20] quantifies frequent and rare deviations between an event log and a stochastic model by treating both log and model as stochastic automata, and comparing the entropy of these automata with the entropy of a third automaton that represents the joint behavior. Probabilistic Alignments [4] consider the frequencies of traces in logs and calculate the likelihood of a move being synchronous or not in the stochastic process model. Bogdanov et al. [5] proposed an alignment-based algorithm that computes the conformance cost between a model and a stochastically known log [14]. Alpha Precision [10] uses the stochastic language of the model and the event log, and inferences about the underlying system that generated the log. Another recent work proposed unit Earth Movers' Stochastic Conformance (uEMSC) and Earth Movers' Stochastic Conformance (EMSC) [16] that measure the effort of transforming the distribution of traces in the log to that described in the stochastic model. Although EMSC can be applied to compute L2M and M2M conformance, it relies on a biased truncation to sample traces from models.

These stochastic conformance checking techniques either only support L2M settings, or the measures provide values that are incomparable across the L2L, L2M, and M2M settings.

3 Preliminaries

Given a set of elements S, a multiset $X : S \to \mathbb{N}$ maps the elements of S to the natural numbers, such that X allows for multiple instances for each of its elements. For example, $X = [a, b^4, c^5]$ is a multiset with ten elements: one a, four b's, and five c's. The union of two multisets X_1 and X_2 is denoted as $X_1 \uplus X_2$. Multiset subset $X_1 \overset{+}{\subseteq} X_2$ denotes $\forall_{s \in S} X_2(s) \geq X_1(s)$. If $X_1 \overset{+}{\subseteq} X_2$, then $X_3 = X_2 \setminus X_1$ is the multiset difference, such that $\forall_{s \in S} X_3(s) = X_2(s) - X_1(s)$.

An *event log* is a collection of *traces*, which are sequences of activities. We can transform an event log into a stochastic language by dividing the frequency of each trace by the total number of traces.

Definition 1 (Stochastic Languages). *Let Σ be a finite set of* activities *and let Σ^* be the set of all finite sequences of activities (*traces*) over Σ. Then, a* stochastic language *l is a function that maps each trace in Σ^* to a probability, that is, $l : \Sigma^* \to [0, 1]$ such that $\sum_{\sigma \in \Sigma^*} l(\sigma) = 1$.*

A stochastic language assigns probabilities to traces so that the assigned probabilities sum up to one. Inherently, an event log denotes a finite stochastic language. For instance, given two event logs $L_1 = [\langle a, b\rangle^3, \langle b, a\rangle^2]$ and $L_2 = [\langle a, b\rangle^{80}, \langle a, b, b\rangle^{20}]$, their finite stochastic languages are $l_1 = [\langle a, b\rangle^{0.6}, \langle b, a\rangle^{0.4}]$ and $l_2 = [\langle a, b\rangle^{0.8}, \langle a, b, b\rangle^{0.2}]$, respectively.

A stochastic process model is a model that describes a stochastic language. We introduce two types of stochastic process models: stochastic labeled Petri nets and stochastic deterministic finite automata.

Definition 2 (Stochastic Labeled Petri Nets). *Let Σ be a finite set of activities, a* stochastic labeled Petri net *(SLPN) is a tuple (P, T, F, w, ρ, m_0) where P is a finite set of places, T is a finite set of transitions such that $P \cap T = \emptyset$, $F \subseteq (P \times T) \cup (T \times P)$ is a flow relation, $w \colon T \to \mathbb{R}^0$ is a weight function, $\rho \colon T \to \Sigma \cup \{\tau\}$ is a labeling function, and m_0 is the initial marking.*

A marking in an SLPN is a multiset of places. An SLPN starts its execution from its initial marking. Let ${}^\bullet t = [p \mid (p, t) \in F]$ be the set of places directly before transition t, $t^\bullet = [p \mid (t, p) \in F]$ be the set of places directly after t, and $T_m = \{t \mid {}^\bullet t \overset{+}{\subseteq} m\}$ denote all *enabled* transitions in marking m. An enabled transition $t \in T_m$ can *fire* with probability $\mathrm{p}(t \mid m) = \frac{w(t)}{\Sigma_{t' \in T_m} w(t')}$, which results in a new marking $m' = m \uplus t^\bullet \setminus {}^\bullet t$.

A *path* is a sequence of transitions $\langle t_1, \ldots, t_n\rangle$ that are fired along with a sequence of markings $\langle m_0, \ldots, m_n\rangle$, such that $\forall_{1 \leq i \leq n} {}^\bullet t_i \overset{+}{\subseteq} m_{i-1} \land m_i = m_{i-1} \uplus t_i^\bullet \setminus {}^\bullet t_i$ and $T_{m_n} = \emptyset$ for m_n. That is, a path brings the model from its initial marking m_0 to a deadlock marking, in which no marking is enabled. The probability of the path $\langle t_0, \ldots, t_n\rangle$ is $\prod_{0 \leq i \leq n} \mathrm{p}(t_i \mid m_{i-1})$. A transition t with $\rho(t) = \tau$ is unobservable, or silent. The projection of a path by labeling function ρ on the non-τ transitions is a trace, and there may be several (even countably-infinite many [19]) paths that project to the same trace.

For instance, M_1 in Fig. 2 is an SLPN with two silent transitions τ_1 and τ_2 and three transitions with labels a, b, and c. $\langle a, \tau_1, \tau_2, \tau_1, b\rangle$ and $\langle a, \tau_1, b\rangle$ are two paths that correspond to the trace $\langle a, b\rangle$. M_1 can generate infinitely many different traces, thus its stochastic language is infinite.

Definition 3 (Stochastic Deterministic Finite Automaton). *A stochastic deterministic finite automaton (SDFA) $A = (S, Q, \delta, p, s_0)$ consists of a finite set*

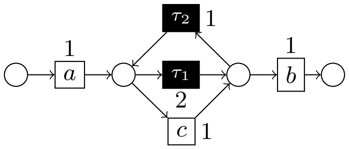


Fig. 2: SLPN M_1.

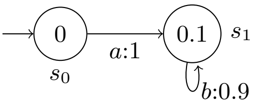


Fig. 3: SDFA M_2.

of states S, with $s_0 \in S$ the initial state, a set of actions $Q \subseteq V$, a transition function $\delta : S \times Q \to S$, and a transition probability function $p : S \times Q \to [0, 1]$ such that for each state $s \in S$, it holds that $\sum_{q \in Q} p(s, q) \leq 1$.

For example, Fig. 3 shows an SDFA using graphical notation. The states and transition function are visualized as circles and arcs, respectively. It has two states, s_0 and s_1. The initial state is s_0, and its transition function is defined by $\{(s_0, a, s_1), (s_1, b, s_1)\}$. Arc from s_0 to s_1 with label a:1 specifies that $(s_0, a, s_1) \in \delta$ and $(s_0, a, 1) \in p$. Likewise, arc from s_1 to s_1 with label b:0.9 specifies that $(s_1, b, s_1) \in \delta$ and $(s_1, b, 0.9) \in p$.

By converting event logs and stochastic models to stochastic languages, we reduce stochastic conformance to the problem of computing the similarity of two stochastic languages. Given two stochastic languages, the Kullback-Leibler Divergence quantifies the difference between the probability distributions over traces in these languages.

Definition 4 (Kullback-Leibler Divergence [9]). *Let Σ be a finite set of* activities*, Σ^* be the set of all finite sequences of activities (*traces*) over Σ, and l and l' be two stochastic languages over Σ. The Kullback-Leibler Divergence (KLD) of l with respect to l' is defined as:*

$$\mathrm{kld}(l, l') = \sum_{\sigma \in \Sigma^*} l(\sigma) \log_2 \frac{l(\sigma)}{l'(\sigma)}.$$

We accept that $0 \log{}_2 0 = 0$. KLD is not symmetric, as $\mathrm{kld}(l, l')$ may not equal $\mathrm{kld}(l', l)$. If one trace has a zero probability in l' and a non-zero probability in l, $\mathrm{kld}(l, l')$ is undefined. The Jensen-Shannon Distance (JSD) overcomes this limitation by comparing two stochastic languages based on their average stochastic language.

Definition 5 (Average Stochastic Languages). *Let Σ be a finite set of* activities*, Σ^* be the set of all finite sequences of activities (*traces*) over Σ, and l and l' be two stochastic languages. The stochastic languages l'' for which it holds that $\forall_{\sigma \in \Sigma^*} l''(\sigma) = 0.5(l(\sigma) + l'(\sigma))$, is the* average stochastic language *of l and l' denoted by $avg(l, l')$.*

For example, $avg(l_1, l_2) = [\langle a, b \rangle^{0.7}, \langle b, a \rangle^{0.2}, \langle a, b, b \rangle^{0.1}]$ is the average stochastic language of l_1 and l_2. Given two stochastic languages, their Jensen-Shannon Distance is defined as follows.

Definition 6 (Jensen-Shannon Distance [13]). *Let l and l' be two stochastic languages. The* Jensen-Shannon Distance *(JSD) between l and l' is defined as:*

$$\mathrm{jsd}(l, l') = \sqrt{\frac{\mathrm{kld}(l, avg(l, l')) + \mathrm{kld}(l', avg(l, l'))}{2}}$$

JSD is bound between 0 and 1. Moreover, JSD using a square root is a metric [13], thus for any stochastic languages l, l' and l'', we have: i) Reflexivity: $\mathrm{jsd}(l, l') = 0 \Leftrightarrow l = l'$, ii) Symmetricity: $\mathrm{jsd}(l, l') = \mathrm{jsd}(l', l)$, and iii) Triangle inequality: $\mathrm{jsd}(l, l') + \mathrm{jsd}(l', l'') \geq \mathrm{jsd}(l, l'')$.

4 Stochastic Conformance Checking with JSD

In this section, we discuss how to compute JSD in L2L, L2M, and M2M settings.

4.1 Log-to-Log Conformance

In the L2L setting, given that a log induces a finite stochastic language, and the average stochastic language of two logs is also finite, one can directly apply Definition 6. Let l and l' be the stochastic languages of two event logs, $L_{>0} = \{\sigma \mid l(\sigma) > 0\}$ and $L'_{>0} = \{\sigma \mid l'(\sigma) > 0\}$, it holds that:

$$\mathrm{jsd}(l, l') = \sqrt{\frac{\mathrm{kld}(l, avg(l, l')) + \mathrm{kld}(l', avg(l, l'))}{2}}, \text{ where:} \tag{1}$$

$$\mathrm{kld}(l, avg(l, l')) = \sum_{\sigma \in L_{>0}} l(\sigma) \log_2 \frac{l(\sigma)}{avg(l, l')(\sigma)} \text{ and,}$$

$$\mathrm{kld}(l', avg(l, l')) = \sum_{\sigma \in L'_{>0}} l'(\sigma) \log_2 \frac{l'(\sigma)}{avg(l, l')(\sigma)}$$

As l and l' for both event logs are finite, the terms in Eq. (1) are finite. We adopt $l(\sigma) \log_2 \frac{l(\sigma)}{avg(l,l')(\sigma)} = 0$ if $l(\sigma) = 0$, and $l'(\sigma) \log_2 \frac{l'(\sigma)}{avg(l,l')(\sigma)} = 0$ if $l'(\sigma) = 0$.

For instance, given the stochastic languages l_1, l_2, for logs L_1 and L_2, we have: $\mathrm{kld}(l_1, avg(l_1, l_2)) = 0.6 \log_2 \frac{0.6}{0.7} + 0.4 \log_2 \frac{0.4}{0.2} \approx 0.267$ and $\mathrm{kld}(l_2, avg(l_1, l_2)) = 0.8 \log_2 \frac{0.8}{0.7} + 0.2 \log_2 \frac{0.2}{0.1} \approx 0.354$. Hence, the JSD between L_1 and L_2 is $\mathrm{jsd}(l_1, l_2) = \sqrt{\frac{0.267+0.354}{2}} \approx 0.575$.

4.2 Log-to-Model Conformance

The definition of JSD relies on the average stochastic language of two input stochastic languages. Hence, the average stochastic language may be infinite. However, as an event log always corresponds to a finite stochastic language, we can avoid explicitly constructing the potentially infinite average stochastic language of an event log and a stochastic model with a finite state space by rewriting Definition 6.

Let l and m be the stochastic language of the input event log and the stochastic language of the stochastic model, respectively. Based on Definition 4 and Definition 5, we have:

$$\operatorname{jsd}(l,m) = \sqrt{\frac{\sum_{\sigma\in\Sigma^*} n(\sigma)}{2}}, \text{ where:} \tag{2}$$
$$n(\sigma) = l(\sigma)\log_2 \frac{2l(\sigma)}{l(\sigma)+m(\sigma)} + m(\sigma)\log_2 \frac{2m(\sigma)}{l(\sigma)+m(\sigma)}.$$

Let $\Sigma_0^* = \{\sigma \mid l(\sigma)=0 \wedge m(\sigma)=0\}$ denote the set of traces that are in neither the log nor the model. For all $\sigma \in \Sigma_0^*$, we have $n(\sigma)=0$. Hence, we only consider the traces in $\Sigma^*\backslash\Sigma_0^*$, i.e., traces that are observed in l or m.

It holds that $\Sigma^*\backslash\Sigma_0^* = \Sigma_1^* \cup \Sigma_2^* \cup \Sigma_3^*$ where $\Sigma_1^* = \{\sigma \mid l(\sigma)>0 \wedge m(\sigma)>0\}$, $\Sigma_2^* = \{\sigma \mid l(\sigma)>0 \wedge m(\sigma)=0\}$, and $\Sigma_3^* = \{\sigma \mid l(\sigma)=0 \wedge m(\sigma)>0\}$. By splitting set $\Sigma^*\backslash\Sigma_0^*$ into the union of three subsets, $n(\sigma)$ in Eq. (2) can be written as a piecewise function:

$$n(\sigma)=\begin{cases} l(\sigma)\log_2 \frac{2l(\sigma)}{l(\sigma)+m(\sigma)}+m(\sigma)\log_2 \frac{2m(\sigma)}{l(\sigma)+m(\sigma)}, & \text{if } \sigma\in\Sigma_1^* \\ l(\sigma), & \text{if } \sigma\in\Sigma_2^* \\ m(\sigma), & \text{if } \sigma\in\Sigma_3^* \end{cases}$$

Then, one can rewrite Eq. (2) as follows:

$$\operatorname{jsd}(l,m) = \sqrt{\frac{j_1(l,m)+j_2(l,m)+j_3(l,m)}{2}}, \text{ where:} \tag{3}$$
$$j_1(l,m) = \sum_{\sigma\in\Sigma_1^*} l(\sigma)\log_2 \frac{2l(\sigma)}{l(\sigma)+m(\sigma)}+m(\sigma)\log_2 \frac{2m(\sigma)}{l(\sigma)+m(\sigma)},$$
$$j_2(l,m) = \sum_{\sigma\in\Sigma_2^*} l(\sigma) = 1-\sum_{\sigma\in\Sigma_1^*} l(\sigma), \text{ and}$$
$$j_3(l,m) = \sum_{\sigma\in\Sigma_3^*} m(\sigma) = 1-\sum_{\sigma\in\Sigma_1^*} m(\sigma).$$

First, we query the model for the probability of each log's trace leveraging the technique discussed in [19] for $j_1(l,m)$. Note that this step is non-trivial, as there can be an infinite number of SLPN paths corresponding to one trace. Therefore, we avoid explicitly computing an infinite average stochastic language for l and m. For instance, for L_2 and M_1, we first calculate the probability of each trace from L_2 in model M_1, that is, $m_1(\langle a,b\rangle)=0.5$, $m_1(\langle b,a\rangle)=0$. Given that $l_2(\langle a,b\rangle)=0.8$, we have: $j_1(l_2,m_1)=0.8\log_2\frac{0.8}{0.65}+0.5\log_2\frac{0.5}{0.65}\approx 0.050$.

Then, the second term $j_2(l_2,m_1)=1-0.8=0.2$, as trace $\langle b,a\rangle$ is only observed in log. The third term $j_3(l_2,m_1)=1-0.5=0.5$ is the probability sum of traces generated by M_1 while not observed in L_2. Finally, the JSD between L_2 and M_1 is $\operatorname{jsd}(l_2,m_1)=\sqrt{\frac{0.050+0.2+0.5}{2}}\approx 0.613$.

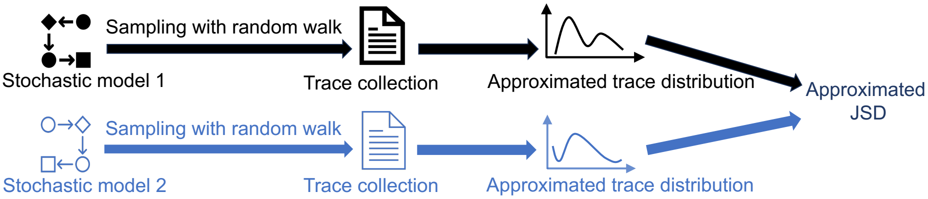

Sampling with random walk
Stochastic model 1
Trace collection
Approximated trace distribution
Approximated JSD
Sampling with random walk
Stochastic model 2
Trace collection
Approximated trace distribution

Fig. 4: JSD approximation for the M2M setting.

4.3 Model-to-Model Conformance

In the M2M setting, given two SDFAs, if their average stochastic language is an SDFA, then it is possible to transform the computation of JSD by leveraging Definition 5. Let m, m', m'' be three SDFAs, such that m'' is the SDFA that induces the average stochastic language of m and m'. The JSD of m and m' is:

$$\mathrm{jsd}(m, m') = \sqrt{\frac{\mathrm{kld}(m, m'') + \mathrm{kld}(m', m'')}{2}}. \tag{4}$$

Applying Eq. (4) relies on the average SDFA for two input SDFAs. However, such an SDFA does not always exist [27]. Hence, in this paper, we do not attempt to find a general strategy to construct an average SDFA for two input SDFAs. Instead, we approximate the true value of JSD by sampling, as illustrated in Fig. 4. For each model, we generate a collection of traces that represent the model's process behavior. In each sampling iteration, a random walk is performed to generate a trace from the model. During a random walk in an SLPN, the probability of firing an enabled transition depends only on the current marking. In an SDFA, the probability of taking the next action depends only on the current state. The walk continues until it reaches the final marking for SLPN or the final state for SDFA, and a trace is generated. Furthermore, each trace is generated independently. Subsequently, the collection of sampled traces is used to construct a finite stochastic language.

The difference between our approach and the truncation technique in [22] is that traces are generated by their probability rather than length, as the truncation approach favors shorter traces over lengthier ones. Thereby, an approximated JSD value can be computed following Eq. (1) using event logs sampled from the models.

5 Evaluation

JSD has been implemented and is publicly available [23]. First, we compare JSD with existing stochastic conformance checking measures. Then, we study the implication of sample size when approximating JSD in the M2M setting.

Table 1: Experiment results of different stochastic conformance values with row-wise ranking. The errors for E-Recall and E-Precision were due to an unknown exception.

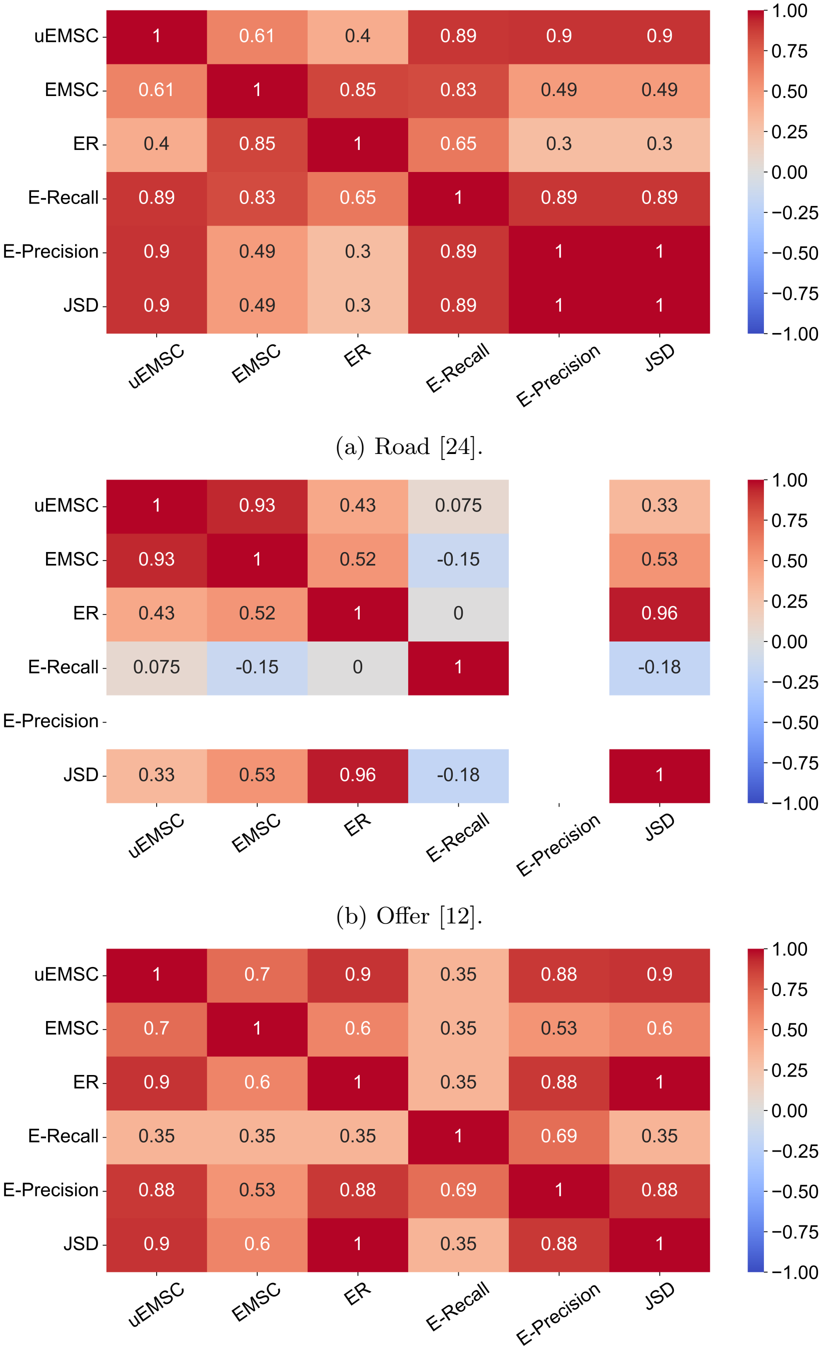
Event Log	Measure	d-uemsc	d-er	d-freq	d-align	d-scale
Road [24]	uEMSC	0.408 (1)	0.221 (2)	0.010 (5)	0.219 (3)	0.112 (4)
	EMSC	0.731 (3)	0.758 (1)	0.641 (5)	0.735 (2)	0.658 (4)
	ER	8.302 (4)	6.685 (1)	23.296 (5)	6.698 (2)	7.731 (3)
	E-Recall	0.909 (1)	0.909 (1)	Error	0.909 (1)	0.836 (4)
	E-Precision	0.783 (1)	0.692 (3)	Error	0.707 (2)	0.512 (4)
	JSD	0.338 (1)	0.389 (3)	0.818 (5)	0.387 (2)	0.514 (4)
Offer [12]	uEMSC	0.656 (1)	0.583 (2)	0.539 (4)	0.581 (3)	0.581 (3)
	EMSC	0.916 (1)	0.910 (2)	0.901 (5)	0.910 (2)	0.910 (2)
	ER	3.363 (4)	3.209 (1)	8.429 (5)	3.210 (2)	3.214 (3)
	E-Recall	0.996 (1)	0.996 (1)	0.996 (1)	0.996 (1)	0.923 (5)
	E-Precision	1.000 (1)	1.000 (1)	1.000 (1)	1.000 (1)	1.000 (1)
	JSD	0.114 (4)	0.108 (1)	0.208 (5)	0.108 (1)	0.109 (3)
Request [11]	uEMSC	0.778 (1)	0.766 (2)	0.005 (5)	0.712 (3)	0.418 (4)
	EMSC	0.886 (2)	0.885 (3)	0.457 (5)	0.897 (1)	0.644 (4)
	ER	8.465 (2)	8.368 (1)	28.471 (5)	8.604 (3)	11.389 (4)
	E-Recall	0.764 (1)	0.764 (1)	0.764 (1)	0.764 (1)	0.000 (5)
	E-Precision	1.000 (1)	1.000 (1)	0.003 (4)	0.814 (3)	0.000 (5)
	JSD	0.091 (2)	0.072 (1)	0.995 (5)	0.118 (3)	0.582 (4)

5.1 Quantitative Comparison

In this experiment, we compare the result of JSD with other stochastic conformance checking measures using three publicly available event logs [24,12,11]. First, given an event log, Inductive Miner [17] is used to construct a control-flow model. Next, we discover a stochastic model (SLPN) using stochastic discovery techniques, including d-uemsc, d-er, d-freq, d-align, and d-scale [7,18]. Finally, different measures have been applied to evaluate the stochastic conformance between each log and SLPNs, namely uEMSC, EMSC, ER, E-Recall, E-Precision, and JSD. The results are presented in Table 1. Note that for uEMSC, EMSC, E-Recall, and E-Precision, a higher value indicates better stochastic conformance. For distance measures ER and JSD, a lower value denotes a better conformance.

Overall, a model with a good uEMSC, EMSC, and ER also ranks higher for JSD. Although there is no unanimous agreement across JSD and other measures on the best stochastic models, there is partial agreement on the worst models. Stochastic models discovered using d-freq and d-scale have lower stochastic quality, as indicated by their ranks of JSD and other conformance measures.

When using the Spearman Correlation to examine the relationship between JSD and other conformance measures, JSD does not present a strong positive correlation with existing measures, as illustrated in Fig. 5. For instance, although JSD is strongly correlated to uEMSC for logs Road and Request, this pattern is not observed in log Offer. When comparing JSD and EMSC, a medium to

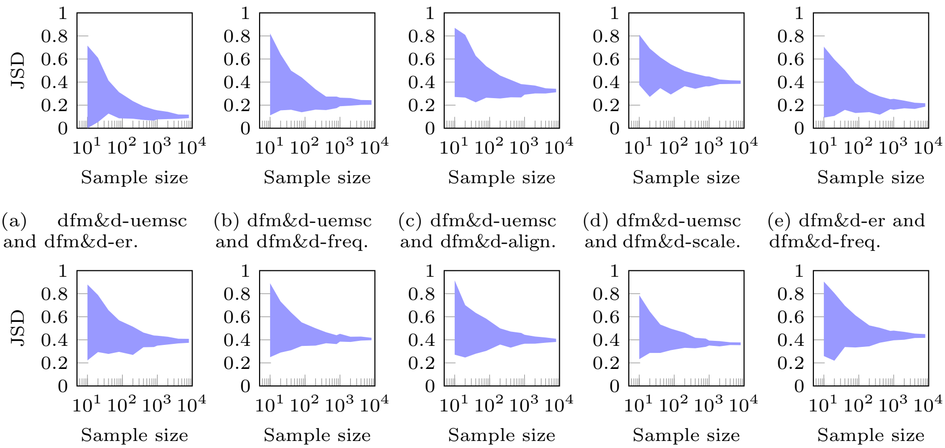


(a) Road [24].

(b) Offer [12].

(c) Request [11].

Fig. 5: Spearman correlation for stochastic conformance measures over different logs.



(a) dfm&d-uemsc and dfm&d-er.

(b) dfm&d-uemsc and dfm&d-freq.

(c) dfm&d-uemsc and dfm&d-align.

(d) dfm&d-uemsc and dfm&d-scale.

(e) dfm&d-er and dfm&d-freq.

(f) dfm&d-er and dfm&d-align.

(g) dfm&d-er and dfm&d-scale.

(h) dfm&d-freq and dfm&d-align.

(i) dfm&d-freq and dfm&d-scale.

(j) dfm&d-align and dfm&d-scale.

Fig. 6: Approximated JSD with sampling in the M2M settings.

high correlation is observed in all three logs. As E-Precision for log Offer is 1 for all models, it does not present any positive or negative correlation with JSD. Therefore, stochastic conformance checking with JSD leads to conclusions that are different from those of existing stochastic conformance measures.

5.2 Influence of Sample Size

In this evaluation, we study the influence of sample size on approximating JSD between two SLPNs. We first constructed control-flow models with loops for log Domestic [11] using Direct-Follow Miner (dfm) [21], and then discovered SLPNs with d-uemsc, d-er, d-freq, d-align, and d-scale. We increased the number of traces sampled from 10 to 8000 to study how the sample size influences returned values. To reduce the effect of randomness, we repeat the computation 500 times for each sample size and compute the average JSD across all the repetitions.

The results are shown in Fig. 6, in which the x-axis represents the sampled trace size and the y-axis is the JSD value. The blue region represents the range of JSD values obtained from repeated experiments. As the number of sampled traces increases, the blue region gradually converges. Specifically, if the sample size is small and insufficient traces are generated, the JSD values vary considerably.

JSD shows expected behavior with an increasing number of sampled traces. The sampling we use is unbiased, i.e., it does not favor shorter traces over longer ones like the truncation technique used in EMSC [22]. With a larger sample size, loops are unfolded in the model with more traces generated, and the stochastic language approaches the true trace distribution of the model.

6 Conclusion

This paper studies the applicability of Jensen Shannon Distance for stochastic conformance checking. JSD is a metric that compares the trace distributions of two stochastic languages with that of their average language. This distance measure can be applied for log-to-log, log-to-model, and model-to-model conformance checking, the latter setting in general requiring an approximation using, for instance, an unbiased sampling presented in this work.

We evaluated the feasibility of JSD for conformance checking using real-life event logs and stochastic process models discovered from these logs. The comparison with existing stochastic conformance measures demonstrated that JSD measurements may lead to different conclusions, an observation deserving of further exploration in future works. In addition, we confirmed empirically that the proposed approximation of model-to-model conformance converges with the growth of simulated event logs.

The metric property of JSD can be useful in applications like searching for similar models [15]. Another interesting direction for future work is to identify the explicit construction of an average stochastic language, i.e., extend the measure for accurate computation for subclasses of models (M2M). Finally, one can assess whether JSD satisfies desired properties for stochastic conformance measures, such as properties designed for stochastic recall and precision measures [20].

References

1. van der Aalst, W.M.P.: Process Mining - Data Science in Action, Second Edition. Springer (2016)
2. van der Aalst, W.M.P.: Relating process models and event logs - 21 conformance propositions. In: ATAED@Petri Nets/ACSD. CEUR Workshop Proceedings, vol. 2115, pp. 56–74. CEUR-WS.org (2018)
3. Alkhammash, H., Polyvyanyy, A., Moffat, A.: Stochastic directly-follows process discovery using grammatical inference. In: CAiSE. LNCS, vol. 14663, pp. 87–103. Springer (2024)
4. Bergami, G., Maggi, F.M., Montali, M., Peñaloza, R.: Probabilistic trace alignment. In: ICPM. pp. 9–16. IEEE (2021)
5. Bogdanov, E., Cohen, I., Gal, A.: Conformance checking over stochastically known logs. In: BPM (Forum). LNBIP, vol. 458, pp. 105–119. Springer (2022)
6. Brockhoff, T., Uysal, M.S., van der Aalst, W.M.P.: Wasserstein weight estimation for stochastic petri nets. In: ICPM. pp. 81–88. IEEE (2024)
7. Burke, A., Leemans, S.J.J., Wynn, M.T.: Stochastic process discovery by weight estimation. In: ICPM Workshops. LNBIP, vol. 406, pp. 260–272. Springer (2020)
8. Carmona, J., van Dongen, B.F., Solti, A., Weidlich, M.: Conformance Checking - Relating Processes and Models. Springer (2018)
9. Carrasco, R.C.: Accurate computation of the relative entropy between stochastic regular grammars. RAIRO Theor. Informatics Appl. **31**(5), 437–444 (1997)
10. Depaire, B., Janssenswillen, G., Leemans, S.J.J.: Alpha precision: Estimating the significant system behavior in a model. In: BPM (Forum). Lecture Notes in Business Information Processing, vol. 458, pp. 120–136. Springer (2022)

11. van Dongen, B.: Bpi challenge 2020 (2020)
12. van Dongen, B.: Bpi challenge 2017 - offer log (2021)
13. Endres, D.M., Schindelin, J.E.: A new metric for probability distributions. IEEE Trans. Inf. Theory **49**(7), 1858–1860 (2003)
14. Gal, A.: Everything there is to know about stochastically known logs. In: ICPM. IEEE (2023)
15. Kunze, M., Weidlich, M., Weske, M.: Behavioral similarity - A proper metric. In: BPM. LNCS, vol. 6896, pp. 166–181. Springer (2011)
16. Leemans, S.J.J., van der Aalst, W.M.P., Brockhoff, T., Polyvyanyy, A.: Stochastic process mining: Earth movers' stochastic conformance. Inf. Syst. **102**, 101724 (2021)
17. Leemans, S.J.J., Fahland, D., van der Aalst, W.M.P.: Discovering block-structured process models from incomplete event logs. In: Petri Nets. LNCS, vol. 8489. Springer (2014)
18. Leemans, S.J.J., Li, T., Montali, M., Polyvyanyy, A.: Stochastic process discovery: Can it be done optimally? In: CAiSE. Lecture Notes in Computer Science, vol. 14663, pp. 36–52. Springer (2024)
19. Leemans, S.J.J., Maggi, F.M., Montali, M.: Enjoy the silence: Analysis of stochastic petri nets with silent transitions. Inf. Syst. **124**, 102383 (2024)
20. Leemans, S.J.J., Polyvyanyy, A.: Stochastic-aware precision and recall measures for conformance checking in process mining. Inf. Syst. **115**, 102197 (2023)
21. Leemans, S.J.J., Poppe, E., Wynn, M.T.: Directly follows-based process mining: Exploration & a case study. In: ICPM. IEEE (2019)
22. Leemans, S.J.J., Syring, A.F., van der Aalst, W.M.P.: Earth movers' stochastic conformance checking. In: BPM Forum. LNBIP, vol. 360, pp. 127–143. Springer (2019)
23. Leemans, S.J., Li, T., van Detten, J.N.: Ebi - a stochastic process mining framework. In: ICPM Doctoral Consortium and Demo Track. CEUR Workshop Proceedings, vol. to appear. CEUR-WS.org (2024)
24. de Leoni, M.M., Mannhardt, F.: Road traffic fine management process (2015)
25. Polyvyanyy, A., Moffat, A., García-Bañuelos, L.: An entropic relevance measure for stochastic conformance checking in process mining. In: ICPM. pp. 97–104. IEEE (2020)
26. Rogge-Solti, A., van der Aalst, W.M.P., Weske, M.: Discovering stochastic petri nets with arbitrary delay distributions from event logs. In: BPM Workshops. LNBIP, vol. 171, pp. 15–27. Springer (2013)
27. Vidal, E., Thollard, F., de la Higuera, C., Casacuberta, F., Carrasco, R.C.: Probabilistic finite-state machines-part I. IEEE Trans. Pattern Anal. Mach. Intell. **27**(7), 1013–1025 (2005)
28. Yeshchenko, A., Ciccio, C.D., Mendling, J., Polyvyanyy, A.: Comprehensive process drift detection with visual analytics. In: ER. Lecture Notes in Computer Science, vol. 11788, pp. 119–135. Springer (2019)