





Entropia: A Family of Entropy-Based Conformance Checking Measures for Process Mining


Artem Polyvyanyy 
The University of Melbourne
artem.polyvyanyy@unimelb.edu.au


Hanan Alkhamash 
The University of Melbourne
halkhamash@student.unimelb.edu.au


Claudio Di Ciccio 
Sapienza Università di Roma
claudio.diciccio@uniroma1.it


Luciano García-Bañuelos 
Tecnológico de Monterrey
luciano.garcia@tec.mx

Anna Kalenkova 
The University of Melbourne
anna.kalenkova@unimelb.edu.au

Sander J. J. Leemans 
Queensland University of Technology
s.leemans@qut.edu.au

Jan Mendling 
Wirtschaftsuniversität Wien
jan.mendling@wu.ac.at

Alistair Moffat 
The University of Melbourne
ammoffat@unimelb.edu.au

Matthias Weidlich 
Humboldt-Universität zu Berlin
matthias.weidlich@hu-berlin.de

Abstract— This paper presents a command-line tool, called *Entropia*, that implements a family of conformance checking measures for process mining founded on the notion of entropy from information theory. The measures allow quantifying classical non-deterministic and stochastic *precision* and *recall* quality criteria for process models automatically discovered from traces executed by IT-systems and recorded in their event logs. A process model has “good” precision with respect to the log it was discovered from if it does not encode many traces that are not part of the log, and has “good” recall if it encodes most of the traces from the log. By definition, the measures possess useful properties and can often be computed quickly.

I. INTRODUCTION

Process mining is a research field concerned with extracting knowledge from event sequence data that is stored in event logs. Conceptually, process mining techniques assume that events have at least three attributes: a timestamp, a case identifier and an activity type [1]. Process mining techniques support various process analysis tasks including automatic process discovery, conformance checking, and variant analysis [2].

Conformance checking refers to those process mining techniques that compare the behavior captured in an event log with a normative process model [3]. A key challenge for research on conformance checking is the definition of appropriate measures that quantify the extent of correspondence between the log and the model. A rich spectrum of measures have been proposed, albeit many of them in an ad hoc manner [4]. The recent stream of work on entropy-based techniques provides a solid theoretical foundation for conformance checking measures with sound properties [5], [6], [7], [8], [9], but in the past tool support has been somewhat limited.

In the paper at hand, we address this gap. Specifically, we present a command-line tool, called *Entropia*, that implements entropy-based conformance checking techniques. The tool is

publicly available¹ and supports process analysts in several scenarios in which commonalities and discrepancies between process models and event logs are measured. Finally, the reader can take a look at a screencast² that demonstrates the tool and check the user guide³ that contains a comprehensive collection of examples and tutorials on using *Entropia*.

The paper proceeds as follows. Section II gives an overview of the theoretical foundations of conformance checking. Section III introduces the *Entropia* tool using a practical use case highlighting its analysis features. Section IV discusses the maturity of the work. Section V provides illustrative examples. Section VI discusses computational performance and current limitations, before Section VII concludes.

II. CONFORMANCE CHECKING

The assessment of the model quality with respect to an event log is paramount for process mining [1]. Buijs et al. [10] introduce four main quality dimensions, namely *fitness*, *precision*, *generalization*, and *simplicity*, which are currently considered the de facto standard. Fitness captures the degree to which the traces recorded in the event log can be replayed on the process model. Precision penalizes the extra behavior introduced by the model that is not recorded in the event log. Conversely, generalization indicates how well the model can support unforeseen traces. Finally, simplicity denotes the capability of the model to express the behavior of the event log while keeping the model easy to understand.

Conformance checking techniques provide a number of approaches for assessing the four quality dimensions. One can broadly classify them into two categories: *descriptive* and *quantitative*. Descriptive techniques construct comprehensive artifacts that aim to explain various aspects of the studied

¹<https://github.com/jbpt/codebase/tree/master/jbpt-pm>

²<https://youtu.be/RZVEFMuH684>

³<https://github.com/jbpt/codebase/tree/master/jbpt-pm/entropia/guide.pdf>

criterion, e.g., a description of all the commonalities and discrepancies between a trace and a process model. Quantitative techniques measure the quantity of the studied phenomenon, e.g., as a number between zero and one. Orthogonal to this classification is the partitioning of conformance checking techniques into non-stochastic and stochastic ones. Stochastic conformance checking techniques study *relations* between some stochastic aspects of the compared model and log, e.g., distributions of traces recorded in the log and described in the model. In contrast, non-stochastic techniques, even though they may rely on the probabilistic aspects of the individual compared artifacts, do not analyze the relations between them.

III. ENTROPIA

This section presents *Entropy* by specifying the use cases it supports (Section III-A), the core principle behind the entropy-based measuring of precision and recall (Section III-B), and the command-line interface (CLI) of the tool (Section III-C).

A. Use Cases

Entropy implements the techniques for quantifying the *precision* and *recall* quality criteria in conformance checking presented in [5], [6], [7], [8], and [9]. Two techniques [8], [9] can be used to measure aspects that relate to stochastic precision and recall quality criteria.

B. Entropy-Based Conformance Checking

The key idea for quantifying precision and recall between a model that describes “relevant” behavior and a model that captures “retrieved” behavior is to measure the magnitude of the behavior the two models share in relation to the magnitude of the behavior of one of the models.

Specific to the process mining context, one can think of an event log as a model that specifies the relevant behavior, i.e., the behavior that provides information about the true behavior it was sampled from. On the other hand, a process model discovered from an event log specifies the “retrieved” behavior, i.e., the behavior the applied discovery algorithm constructed from the input event log.

Then, by following the principle for defining the corresponding quality criteria in information retrieval [5], precision can be defined as the ratio of the magnitude of the shared behavior specified by the models of relevant and retrieved behaviors to the magnitude of the retrieved behavior. Similarly, recall is the ratio of the magnitude of the shared behavior to the magnitude of the relevant behavior.

Figure 1 visualizes these ideas. Note that $rel \cap ret$ refers to the behavior shared by the relevant and retrieved behaviors of the compared models.

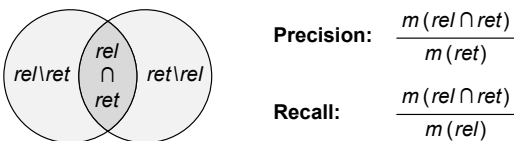


Fig. 1. Precision and recall quotients [5].

In the figure, *rel* and *ret* represent the relevant behavior and retrieved behavior, respectively. Function *m* is used to measure the magnitude of the corresponding (part) of the behavior. The conformance checking approaches implemented in *Entropy* interpret the behaviors of the compared models as collections of the traces that these models describe, where a trace is a sequence of process actions. Function *m* is implemented as the measure of the entropy of a collection of traces; note that the implemented conformance checking techniques use different notions of entropy, and in different ways, refer to Section IV.

The benefit from using the entropy to measure the magnitudes of collections of traces when calculating precision and recall is twofold. First, one can measure entropy of an arbitrary (potentially infinite) collection of traces. Second, the entropy-based precision and recall measures can achieve a range of desired properties [4], [11], [5], [8].

C. Interface

As of August 2020, the *Entropy* tool is in version 1.5. It is invoked by executing the command:

```
java -jar jbpt-pm-entropy-1.5.jar <options>
```

The core CLI options of *Entropy* are listed in Table I.

TABLE I
CORE CLI OPTIONS OF THE *Entropy* TOOL.

| Option (full) | Option | Parameter | Description |
|---------------|--------|-----------|---------------------------------------|
| --help | -h | | print help message |
| --relevant | -rel | <path> | model that describes relevant traces |
| --retrieved | -ret | <path> | model that describes retrieved traces |
| --silent | -s | | run tool in the silent mode |
| --version | -v | | get version of this tool |

The `-h` and `-v` options print the help message and tool version, respectively, while options `-rel` and `-ret` are used to specify the models of relevant and retrieved traces, respectively. To refer to a model, the user specifies its file path. Option `-s` runs the tool in silent mode, in which the result of the invocation is printed, without any debug information or execution data. The tool accepts input models specified in one of the following formats: eXtensible Event Stream (XES) [12], Petri Net Markup Language (PNML) [13], Stochastic Petri Net Markup Language (sPNML), Directly-Follows Graph (DFG), Stochastic Deterministic Finite Automaton (S DFA). The latter three formats are specific to our tool.

Further CLI options allow selection of a conformance measure to be applied to the input data, and configuration of it, and are detailed in the next section.

IV. MATURITY

The work on the code base of the tool started in August 2017, together with the start of the work on the entropy-based approach for measuring precision and recall presented in [5]. The tool is integrated into the jBPT library [14], a compendium of open-source business process technologies, the work on which commenced in January 2009.

The approach presented in [5] suggests measuring precision and recall by interpreting the compared models, e.g., process model and event log, as collections of traces that they describe. The models are said to specify shared behavior if and

only if they describe identical traces. The magnitude of the behavior captured by each of the compared models, and of the behavior shared by the models, is determined as *topological entropy* [15] of the corresponding collection of traces.

In [6], the authors generalize the approach from [5] by replacing every collection of traces involved in the calculations of the precision and recall measures with the collection of all substraces of all the traces it contains. Consequently, the shared behavior of two collections of traces is identified as a collection of all sequences of actions that are substraces of some traces in both compared collections. That is, this approach considers all the shared subsequences of actions in the compared models of traces for the measurements.

The measures described in [5] and [6] can be seen as extremes of the spectrum, with either *no* or *all* substraces considered when determining the magnitudes of the collections of traces. The approach presented in [7] allows for a flexible analysis. In particular, based on knowledge of the compared models, the user can specify the maximal number of allowed skipped actions in a trace described by each of the models of traces for determining the shared substraces. This way, the user can tune the measures towards the desired sensitivity to the discrepancies in the compared behaviors.

In [8], entropy is used to extend conformance checking to stochastic process mining. An event log and a stochastic process model can be compared based on whether they exhibit the same control flow, but also based on whether the frequency of behavior in the event log matches the probabilities of behavior in the model. To this end, both log and model are translated into stochastic deterministic finite automata, the conjunction of these automata is constructed, and the entropy of these three automata yields two measures: stochastic recall and stochastic precision. In [8], an evaluation shows the practical applicability by searching for pairwise similar process models in a 4000-model repository.

Finally, the entropic relevance measure presented in [9] is a stochastic conformance measure computed as the average number of bits required to compress (i.e., to perform the entropy coding of) a trace from the log using the information on the relative likelihood of traces encoded in the model.

Table II lists the tool options to select and configure the supported conformance measures. Table III then summarizes the characteristics of the conformance checking approaches implemented in *Entropia* by specifying the input models of

TABLE II
SPECIFIC CLI OPTIONS OF THE *Entropia* TOOL.

| Option | Parameter | Description | Publ. |
|--------|-----------|---|-------|
| -emp | | exact matching precision | [5] |
| -emr | | exact matching recall | [5] |
| -pmp | | partial matching precision | [6] |
| -pmr | | partial matching recall | [6] |
| -cpmp | | controlled partial matching precision | [7] |
| -cpmr | | controlled partial matching recall | [7] |
| -srel | <num> | number of allowed skips in relevant traces | [7] |
| -sret | <num> | number of allowed skips in retrieved traces | [7] |
| -sp | | stochastic precision | [8] |
| -sr | | stochastic recall | [8] |
| -r | | entropic relevance | [9] |

TABLE III
CHARACTERISTICS OF CONFORMANCE CHECKING APPROACHES.

| Publ. | L-L | L-M | M-M | Stoch. | Log | Model |
|-------|-----|-----|-----|--------|-----|-----------|
| [5] | yes | yes | yes | no | XES | PNML |
| [6] | yes | yes | yes | no | XES | PNML |
| [7] | yes | yes | yes | no | XES | PNML |
| [8] | yes | yes | yes | yes | XES | sPNML |
| [9] | yes | yes | no | yes | XES | DFG, SDFa |

retrieved and relevant traces (L–event log, M–process model) supported by the approach presented in the corresponding publication (Publ.); the ability to address the stochastic aspect of the input models of traces (Stoch.); and the event log and process model formats supported (Log–event log, and Model–process model).

The approaches listed in Table III, except the technique presented in [9], can be used to quantify precision and recall conformance criteria between two (possibly infinite) collections of traces. The approach in [9] measures the *entropic relevance* of a stochastic process model to an event log. Relevance reflects a compromise between the precision and recall criteria and has meaningful units.

V. EXAMPLES

In this section we provide some examples of *Entropia*, using the Petri net N in Fig. 2, the SDFa A in Fig. 3, and an event log $E = [abce, ace, bce^2, abcdcbce, abdcbe, aaacbe]$. Note that E contains two instances of bce .

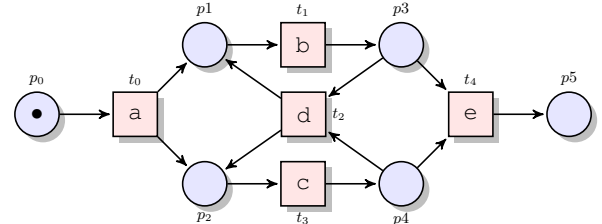


Fig. 2. A Petri net.

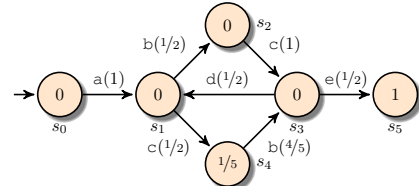


Fig. 3. An SDFa.

The entropy-based exact matching precision between N and E presented in [5] is computed using the CLI options:

```
-emp -rel=E.xes -ret=N.pnml
```

To allow up to two skips in traces of the Petri net and up to one skip in the traces of the log, as described in [7], when identifying similar traces in the computation of precision, these CLI options should be employed:

```
-cpmp -rel=E.xes -ret=N.pnml -srel=1 -sret=2
```

These options compute the entropic relevance of A to E :

```
-r -rel=E.xes -ret=A.sdfa
```

The computed values of exact matching precision, controlled partial matching precision, and entropic relevance using

the above CLI options for net N , SDFA A and log E are 0.776, 0.833, and 11.368 bits, respectively. Further examples of *Entropia* and the serialized models and log used in the examples discussed above appear in the user guide.³

VI. DISCUSSION

All the techniques implemented in *Entropia* ver. 1.5 support process models that describe arbitrary (potentially infinite) collections of traces and impose no limitations on input logs provided that they are explicitly recorded and, thus, are finite. However, process models must be bounded, i.e., they must induce finite reachability graphs. Various notions of semantic correctness of process models require process models to be bounded. Nevertheless, process models used in practice can be incorrect, thus potentially unbounded. Hence, each process model provided as input to *Entropia*, which is not guaranteed by definition to be bounded, is tested by default for boundedness using LoLA ver. 2.0 [16]. One can check if a process model is bounded using option `-b` of the tool. If the boundedness of process models is established, one can invoke *Entropia* with option `-t` to skip the model correctness tests.

Entropia is implemented in Java and integrates with the LoLA tool compiled for Windows. To use *Entropia* on another platform, one needs to recompile LoLA for that platform.

Different conformance techniques implemented in *Entropia* have different performance characteristics. The computation time of entropic relevance [9] is linear in the size of the event log (number of traces times average length of a trace). The computation of entropy-based precision and recall [5], [6], [7] is low polynomial in the size of the reachability graphs of the compared models of traces. However, in practice, a reachability graph can be large, and possibly exponential in the size of the original model due to state explosion. Empirical evidence suggests that the approach grounded in the exact matching of traces [5] runs in the order of seconds on real-world datasets, as the state explosion does not manifest often. Grounding in the partial matching of traces [6], on the other hand, induces large reachability graphs. Thus, it is recommended for small inputs, e.g., when calibrating a new automated process discovery technique. The controlled partial matching technique [7] can be configured by the user to the desired performance, balancing the number of allowed mismatches between similar traces and runtime, with fewer allowed mismatches allowing faster computation. The techniques reported in [5], [6] constitute the two extremes of the trade-off spectrum. Finally, the computation of the stochastic measures presented in [8] relies on an iterative procedure which converges deterministically to the correct values, and is often quick; but which can also lead to prolonged computation times on real-world datasets.

Future work on *Entropia* will both aim to improve the runtime performance of the current techniques and also implement new state-of-the-art information theoretic approaches to conformance checking, including those that assess quality criteria beyond precision and recall. For example, in [17] initial

ideas are provided on using entropy to measure the simplicity of a model automatically discovered from a log.

VII. CONCLUSION

This paper presents *Entropia*, an open-source command-line tool for quantifying precision and recall conformance quality criteria in process mining. The current version of the tool implements several measures, all grounded in the notion of the entropy of a collection of traces described by a process model or event log. The supported measures can be used to assess both classical and stochastic precision and recall, and fulfill a wide range of desired properties suggested by the process mining community. The development of the tool's code base commenced in 2017 and is maintained by the authors of the implemented techniques.

Acknowledgment. Artem Polyvyanyy and Anna Kalenkova were in part supported by the Australian Research Council project DP180102839.

REFERENCES

- [1] W. M. P. van der Aalst, *Process Mining—Data Science in Action*, 2nd ed. Springer, 2016.
- [2] M. Dumas, M. L. Rosa, J. Mendling, and H. A. Reijers, *Fundamentals of Business Process Management*, 2nd ed. Springer, 2018.
- [3] J. Carmona, B. F. van Dongen, A. Solti, and M. Weidlich, *Conformance Checking—Relating Processes and Models*. Springer, 2018.
- [4] N. Tax, X. Lu, N. Sidorova, D. Fahland, and W. M. P. van der Aalst, "The imprecisions of precision measures in process mining," *Inf. Process. Lett.*, vol. 135, pp. 1–8, 2018.
- [5] A. Polyvyanyy, A. Solti, M. Weidlich, C. D. Ciccio, and J. Mendling, "Monotone precision and recall measures for comparing executions and specifications of dynamic systems," *ACM Trans. Softw. Eng. Methodol.*, vol. 29, no. 3, 2020.
- [6] A. Polyvyanyy and A. Kalenkova, "Monotone conformance checking for partially matching designed and observed processes," in *ICPM*. IEEE, 2019, pp. 81–88.
- [7] A. Kalenkova and A. Polyvyanyy, "A spectrum of entropy-based precision and recall measurements between partially matching designed and observed processes," in *ICSOC*, ser. LNCS. Springer, 2020.
- [8] S. J. J. Leemans and A. Polyvyanyy, "Stochastic-aware conformance checking: An entropy-based approach," in *CAiSE*, ser. LNCS, vol. 12127. Springer, 2020, pp. 217–233.
- [9] A. Polyvyanyy, A. Moffat, and L. García-Bañuelos, "An entropic relevance measure for stochastic conformance checking in process mining," in *ICPM*. IEEE, 2020, In Press.
- [10] J. C. A. M. Buijs, B. F. van Dongen, and W. M. P. van der Aalst, "Quality dimensions in process discovery: The importance of fitness, precision, generalization and simplicity," *Int. J. Cooperative Inf. Syst.*, vol. 23, no. 1, pp. 1–40, 2014.
- [11] A. F. Syring, N. Tax, and W. M. P. van der Aalst, "Evaluating conformance measures in process mining using conformance propositions," *Trans. Petri Nets Other Model. Concurr.*, pp. 192–221, 2019.
- [12] "IEEE standard for eXtensible Event Stream (XES) for achieving interoperability in event logs and event streams," *IEEE Std 1849-2016*, pp. 1–50, 2016.
- [13] J. Billington, S. Christensen, K. M. van Hee, E. Kindler, O. Kummer, L. Petrucci, R. Post, C. Stehno, and M. Weber, "The Petri Net Markup Language: Concepts, technology, and tools," in *ATPN*, ser. LNCS, vol. 2679. Springer, 2003, pp. 483–505.
- [14] A. Polyvyanyy and M. Weidlich, "Towards a compendium of process technologies—the jBPT library for process model analysis," in *CAiSE Forum*, ser. CEUR, vol. 998. CEUR-WS.org, 2013.
- [15] T. Ceccherini-Silberstein, A. Machí, and F. Scarabotti, "On the entropy of regular languages," *Theor. Comput. Sci.*, vol. 307, no. 1, 2003.
- [16] K. Schmidt, "LoLA: A low level analyser," in *ATPN*, 2000, pp. 465–474.
- [17] A. Kalenkova, A. Polyvyanyy, and M. La Rosa, "A framework for estimating simplicity of automatically discovered process models based on structural and behavioral characteristics," in *BPM*. Springer, 2020.