Real-Time Process Mining?

Phil Weber, Behzad Bordbar and Peter Tiňo
School of Computer Science, University of Birmingham

EPSRC
Engineering and Physical Sciences Research Council

Introduction — Process Mining

Process mining [1]: the extraction of (business) process models from information systems' log files,

- what activities? in what order?
- which people or resources interact?
- are audit requirements satisfied?
- where are the bottlenecks? what happens if ... ?

drawing from *machine learning* and *data mining*; informing *business analysis*, *BPM*, *service-oriented architectures*, etc.

Which Mining Algorithm is 'Best'?

Many mining algorithms and modelling languages, but

- Which is 'best' in a particular situation?
- On what does this depend?
- How correct is the mined model?
- How much data is needed?

Objective choice of algorithm + *just enough* data
⟶ process mining in near real time
⟹ improved process monitoring & response to change.

A Probabilistic Approach

Abstract from the representation: treat business processes as probability distributions over strings:

1. Activity = symbol; process trace = string.
2. Underlying process model M generates traces according to (unknown) distribution P_M *Fig. 1 (1) and (6)*.
3. Mined model $M1$ is a (different?) distribution Q_{M1} *(2), (6)*.
4. Compare Q_{M1} and P_M to assess convergence *(7), (8)*.
5. Use stochastic automata as a minimal representation of both process models and distributions *(5)*.

References

[1] W. M. P. van der Aalst and A. J. M. M. Weijters, *"Process mining: a research agenda"*, *Computers and Industry*, vol. 53, no. 3, pp. 231–244, 2004.

[2] W. van der Aalst, T. Weijters, and L. Maruster, *"Workflow mining: discovering process models from event logs"*, *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 9, pp. 1128–42, 2004.

Framework: Theoretical Analysis

Assess algorithms' ability to learn these distributions:

1. Develop probability formulae for discovery of process structures.
2. Extend to discovery of transition probabilities, where used.
3. Aggregate to overall discovery probability of arbitrary process models.
4. Bound to the required level of accuracy and confidence.

⟹ investigate algorithms' behaviour (rate of convergence, issues causing lack of convergence, relation to other algorithms).

Overview

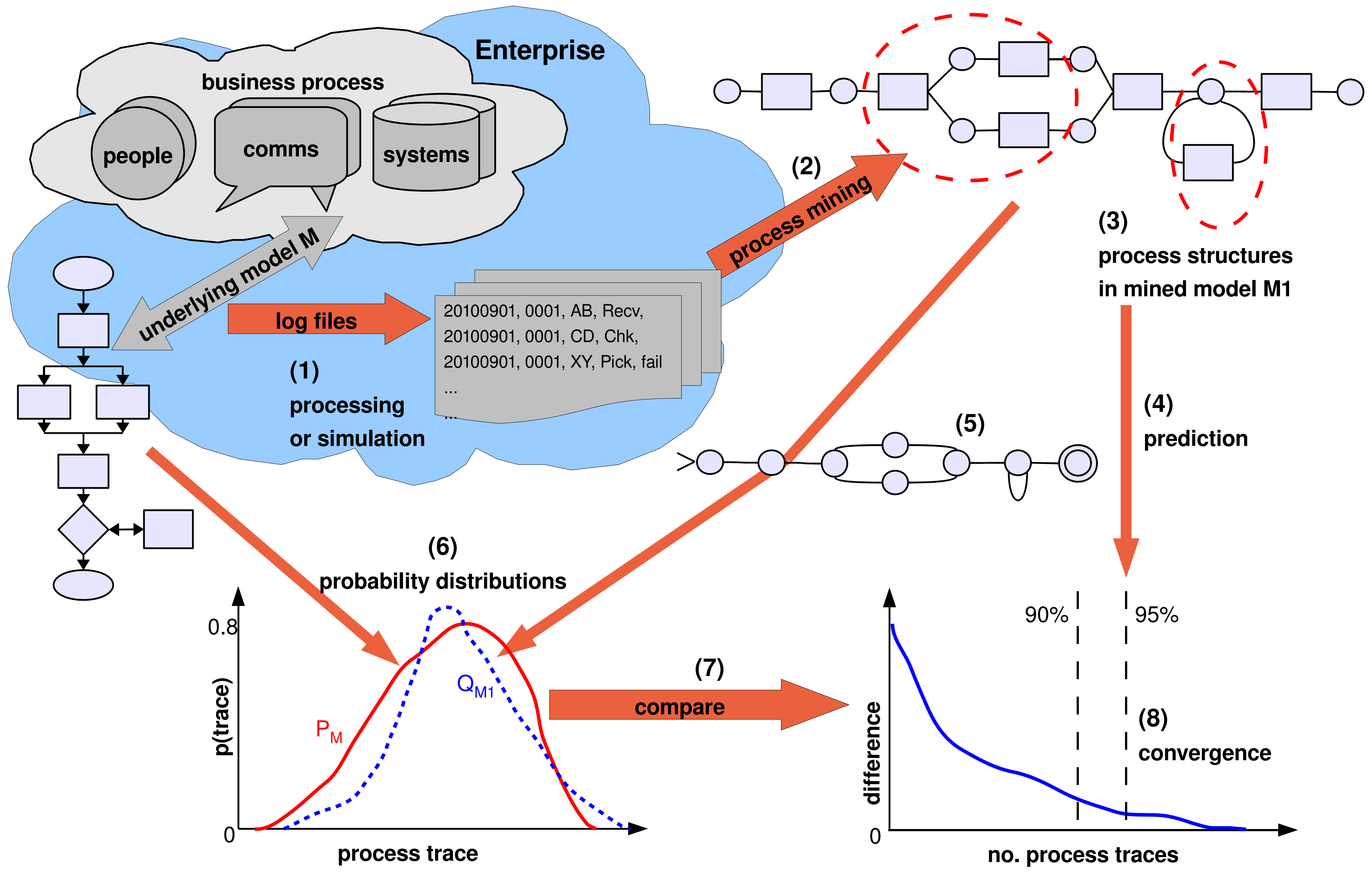

Figure 1: Approach to mining convergence

Framework: Experimental Evaluation

Test with process structures and full models: to validate the theory and gather data to inform choice of algorithm, amount of data, and confidence in results:

1. Design test models (varying structures and probabilities).
2. Analyse structures and predict number of traces for confidence levels *Fig. 1 (3) and (4)*.
3. Simulate to produce sample sets of logs of various sizes *cf (1)*.
4. Run mining algorithms *(2)*, converting results to stochastic automata *(5)*, and compare distributions represented by the mining results and by the designed model *(6), (7), (8)*.
5. Average results over multiple random simulated logs for statistical validity, and assess results against theoretical predictions.

Analysis of the Alpha Algorithm

We analysed the Alpha algorithm [2]. Figs. 2 and 3 show the behaviour when mining an XOR split with, and without noise.

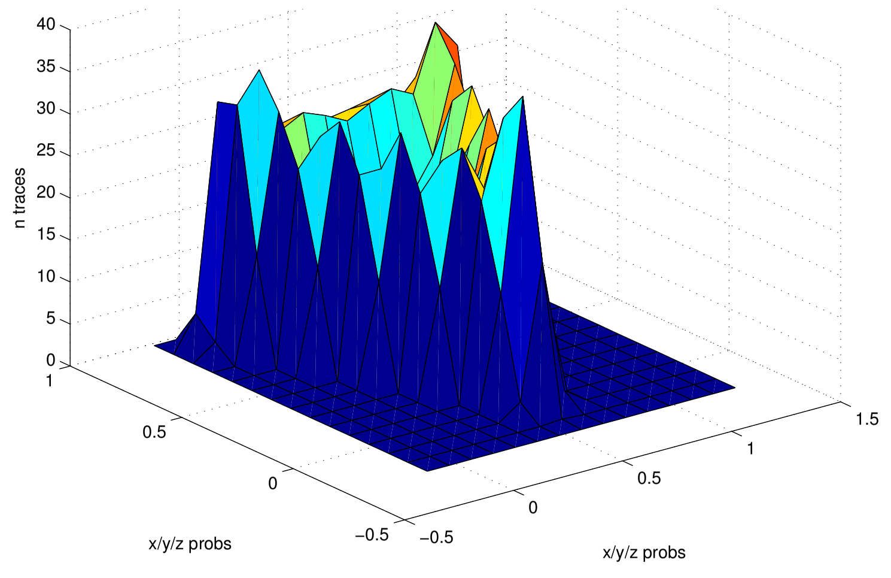

Figure 2: No. Traces for 95% Probability of mining XOR Split.

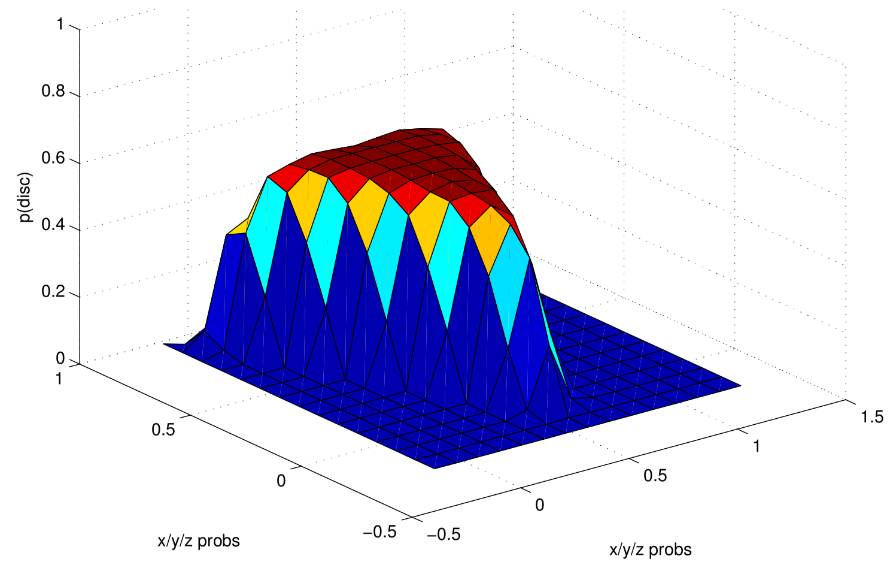

Figure 3: Probability of Mining XOR Split with Noise, in 20 traces.

ProM (`http://www.processmining.org`) was used to mine a simple test model from simulated logs.

Initial results (Fig. 4) show that the amount of data needed for mining can indeed be successfully predicted.

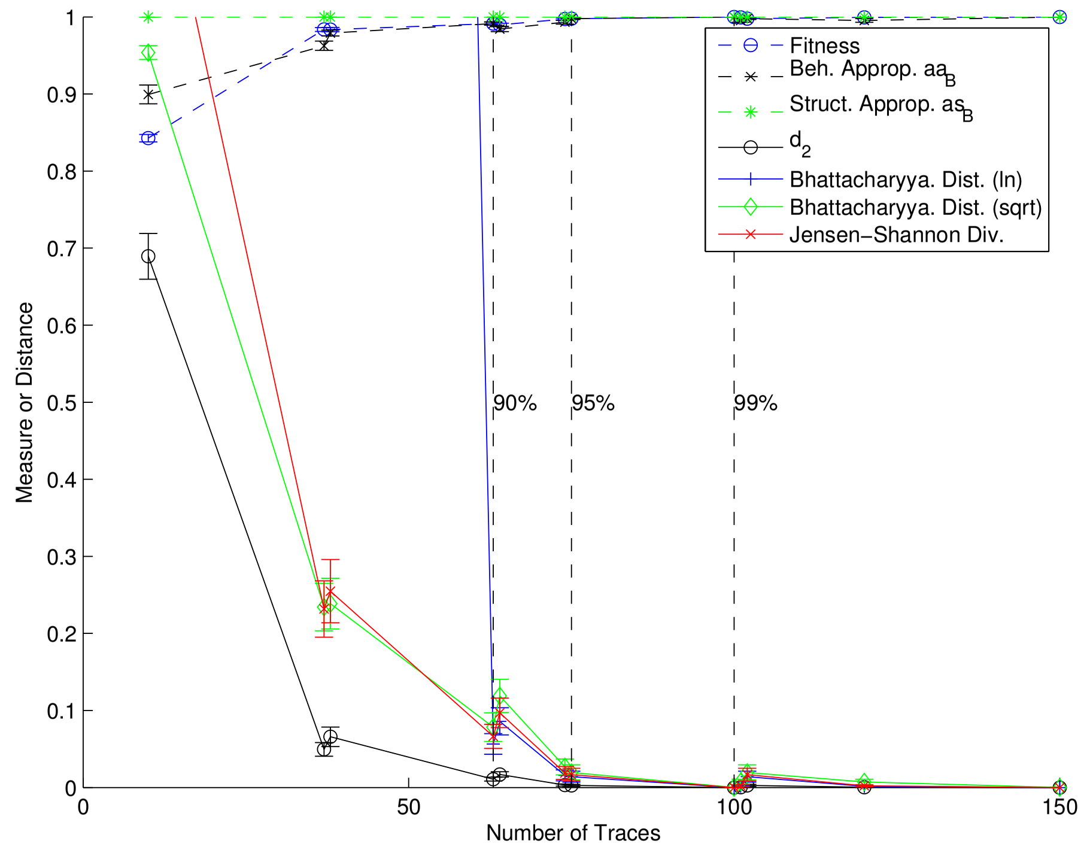

Figure 4: Convergence of Mined Model with Ground Truth

Future Work and Real-Time Process Mining

We aim to compare the behaviour of different types of algorithms and process structures, to develop general results for the capabilities of process mining algorithms.

- ⟹ methods to quantify the confidence that can be placed on the results of mining unknown process models, based on the amount of data used, the type of algorithm, and the structures discovered.
- ⟹ an understanding of how near to "real-time" can be achieved in a particular situation.
- ⟹ monitoring of running processes, and detection or prediction of fault situations.