Quantifying Process Model Conformance Through Minimal-Cost Approximations

Bruce Hamilton and Liqiang Geng

National Research Council of Canada

Institute for Information Technology

Introduction

Process Mining **refers to the automated discovery of process models from event logs. It can be applied to auditing software processes or discovering workflow tendencies. Our method uses a new approximation algorithm for testing the conformance between process models and their corresponding event logs.**

Our Approach

Given a process model and an instantiation of the process, our algorithm will approximate the most likely path in the model. Our method extends on the heuristic search used in [1] by incorporating concurrency. Incorporating concurrency requires maintaining a list of current execution points, or places. With this new addition, process instances may be approximated for all process models (i.e. Petri nets, finite state machines, heuristic nets).

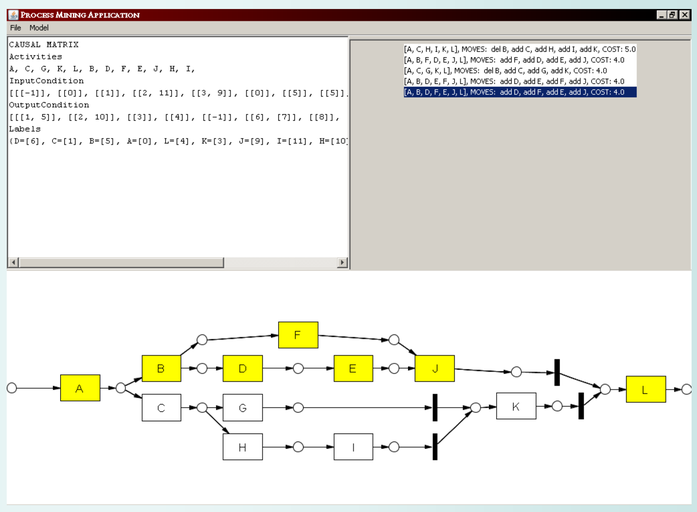
The minimal-cost approximation algorithm can be used to give detailed reports to developers or administrators about the discrepancies in process execution. Our motivation is to use these reports to give administrators insight into employee workflow so that security threats may be reported.

Reports can be in the form of visual representations (see visual implementation), or conformance metrics (see results).

Visual Implementation

Using the approximation algorithm, we can give visual reports to administrators on the most likely paths for a given process instance. In the figure below, we show an execution of the event sequence, "A, B, L". The set of solutions below k cost are listed, then the user may select a path to highlight in the graph visualization in the bottom frame of the GUI.

This implementation was programmed in Java, along with an extension program for laying out graphs developed at AT&T



Results

Using a set of noisy logs from [2], we show how the algorithm can display statistics on the conformance between a process model and its corresponding event logs.

Measure	Formula
Given a max cost k, and an event log P, consisting of process instances p_i and events e, where cost(p_i) returns the cost of the lowest approximation of p_i.	
Mean	$\frac{\sum_{i=1}^{\lvert P\rvert} cost(p_i)}{\lvert P\rvert}$
Std Dev	$\sqrt{\frac{\sum_{i=1}^{\lvert P\rvert} (cost(p_i) - mean)^2}{\lvert P\rvert - 1}}$
Max	$max\{cost(p_i) \lvert p_i \in P\}$
Min	$min\{cost(p_i) \lvert p_i \in P\}$
Fitness	$\frac{\lvert\{e \in P\}\rvert - \left[\sum_{i=1}^{\lvert P\rvert} cost(p_i) + k \cdot \lvert\{p_i \in P \mid cost(p_i) > k\}\rvert\right]}{\lvert\{e \in P\}\rvert}$

The table to the left shows the measures used, and the table below shows the results using the process models shown (as Petri nets).

Fig. 1: A12 Model

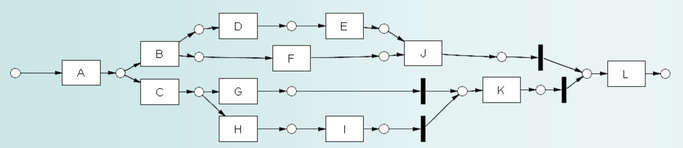


Fig. 2: BN1 Model

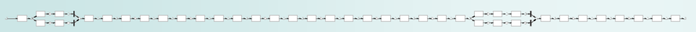


Fig. 3: Herbst Fig. 3.4

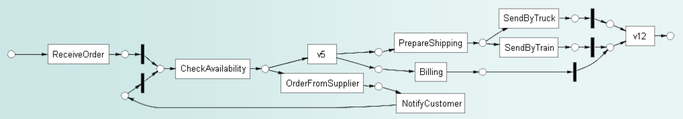
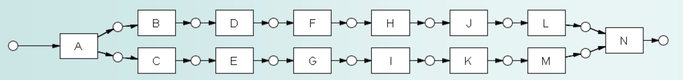


Fig. 3: Herbst Fig. 6.37



Model	Noise	k	Mean	Std dev	Max	Min	(>k)	Fitness	Time (ms)
A12	5.00%	5	1.53	1.42	4	0	0	0.735	40.60
		4	1.53	1.42	4	0	0	0.735	11.99
		3	0.82	1.60	2	0	3	0.765	9.02
		2	0.82	1.24	2	0	3	0.796	5.61
		1	0.24	1.31	1	0	8	0.878	4.01
	10.00%	5	1.71	1.30	4	0	0	0.698	17.79
		4	1.71	1.30	4	0	0	0.698	15.51
		3	1.00	1.49	2	0	5	0.730	13.35
		2	1.00	1.14	2	0	5	0.761	8.59
		1	0.29	1.34	1	0	15	0.855	5.79
BN1	5.00%	5	2.00	1.91	5	0	0	0.942	1410.71
		4	1.62	1.96	4	0	1	0.944	865.94
		3	0.38	2.41	2	0	5	0.955	219.62
		2	0.38	1.79	2	0	5	0.966	78.11
		1	0.23	1.32	1	0	6	0.980	16.62
	10.00%	5	1.90	2.22	5	0	4	0.921	4974.37
		4	1.55	2.12	4	0	6	0.928	1272.91
		3	0.72	2.32	3	0	12	0.940	275.18
		2	0.41	1.97	2	0	15	0.956	100.56
		1	0.28	1.40	1	0	17	0.974	20.32
Herbst 3.4	5.00%	5	0.35	0.83	4	0	0	0.975	11.87
		4	0.35	0.83	4	0	0	0.975	11.30
		3	0.25	0.84	2	0	1	0.976	12.11
		2	0.25	0.73	2	0	1	0.978	10.62
		1	0.10	0.68	1	0	4	0.985	9.62
	10.00%	5	0.58	0.93	4	0	0	0.955	14.66
		4	0.58	0.93	4	0	0	0.955	14.44
		3	0.50	0.93	3	0	1	0.956	15.60
		2	0.44	0.87	2	0	2	0.959	14.75
		1	0.20	0.83	1	0	8	0.972	11.96
Herbst 6.37	5.00%	5	0.10	0.54	4	0	0	0.993	61.91
		4	0.10	0.54	4	0	0	0.993	78.69
		3	0.04	0.55	2	0	2	0.994	69.57
		2	0.04	0.44	2	0	2	0.995	34.95
		1	0.01	0.36	1	0	4	0.997	33.70
	10.00%	5	0.24	0.76	4	0	0	0.983	86.24
		4	0.24	0.76	4	0	0	0.983	91.25
		3	0.16	0.76	3	0	3	0.984	79.97
		2	0.14	0.67	2	0	4	0.986	40.18
		1	0.04	0.58	1	0	11	0.992	35.95

Conclusion

We have developed a working algorithm for discovering the most probable sequence of transitions to be taken in a process model, given a process instance. The algorithm has been shown to be applicable as a basic metric for determining the conformance between a process model and an event log, and has been proposed as a highly descriptive way for determining common faults in a log or model.

References

[1] Jonathan E. Cook and Alexander L. Wolf. Software process validation: quantitatively measuring the correspondence of a process to a model. *ACM Transactions on Soft ware Engineering and Methodology*, 8:147–176, 1999.

[2] A.K. Alves de Medeiros. *Genetic Process Mining*. PhD thesis, Eindhoven University of Technology, 2006.

NRC · CNRC