On Timing Process Interventions for Prescriptive Process Monitoring

Hans Weytjens, Wouter Verbeke, Jochen De Weerdt

Research Centre for Information Systems Engineering (LIRIS), KU Leuven, Belgium
hans.weytjens@kuleuven.be, wouter.verbeke@kuleuven.be, jochen.deweerdt@kuleuven.be

Introduction

The shift from the understanding and prediction of processes enabled by modern digital systems to their optimization, which is the subject of this work, delivers tangible benefits. In its most basic form, the optimization of a process assumes a correctly timed intervention (or non-intervention) in it. Examples range from the escalation of a customer complaint process to higher management echelons, a customer call to speed up an administrative process or to maximize turnover, an additional test to conduct to reduce patients' length of stay at hospitals, etc. *Prescriptive process monitoring* (PresPM) is a young subfield of process mining studying process optimization methods. Two of those methods, *causal inference* (CI) and *reinforcement learning* (RL) emerge as pathways; however, a quantitative comparison is missing in the existing literature. This is largely due to the prevailing use of offline historical data that imposes two limitations: online RL is not possible and experimental results are hard to quantify accurately for lack of *counterfactuals* (outcomes for the alternative intervention) in the datasets. Moreover, no online RL research considers business processes, while all CI research focuses exclusively on them.

We contribute to the PresPM research by introducing online RL to business processes and benchmarking it against CI in experiments using synthetic data.

Background

Causal Inference

CI (see example in Figure 1a) involves estimating the effect of treatments. We concentrate on the *individual treatment effect* (ITE), which is the difference between predicted outcomes of (possible) treatment(s) and non-treatment for a given sample. When calling (treatment) customer x is predicted to increase revenue by 200€, whilst x is expected to reduce sales by 100€ if not called (non-treatment), then the ITE$_x$ is 300€. Usually, a threshold (e.g., 50€ in our example) is determined to arrive at a policy for selecting (non-) treatments. The main challenges in CI are the absence of *counterfactuals* in the dataset and the bias introduced by the the data-gathering policy. Most CI methods were developed for cross-sectional problems and fail to find optimal policies for longitudinal problems such as process optimization. Nevertheless, some authors report positive results.

Reinforcement Learning

In contrast, online RL (see example in Figure 1b) can be shown to converge to optimal policies. RL algorithms learn policies that guide an agent's behavior or sequence of *actions* in an environment in order to maximize an expected cumulative *reward*. At its core, RL assumes an *online* environment that the agent can interact with. RL does not need a process model, instead real (or simulated) processes are executed and their rewards (outcomes) observed. In deep Q-learning, an important RL variant, a neural network (NN) learns a *state-action value* (Q) for all possible actions for every encountered *state* (corresponding to a process prefix). At every state (prefix), the policy will be to choose the action with the highest relative state-action value. In order to explore all areas of the state space and to prevent prematurely settling into a sub-optimal policy, a certain degree of exploration is introduced: the agent will sometimes overrule the policy and choose another action, especially at the beginning of the learning process.

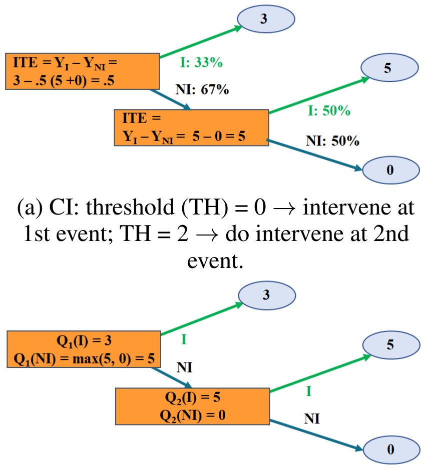
Problem Complexity

From the analysis of the related work, we identify two main drivers for problem complexity: *action width* and *action depth*. Action width relates to the number of different actions available to an agent: actions can be binary (we call those *interventions*, e.g., "apply" or "don't apply") or multi-class (e.g., several options such as "visit", "call" or "email" customer or "do nothing"). In the CI literature, actions are called "'treatments", a term we henceforth substitute by "actions". In the limit, multi-class actions become all activities, even including attributes, as in next best activity prediction. The action depth is a measure for the longitudinal dimension: an action's timing can be predetermined (fixed or irrelevant) or an action can happen once at any time during the process' lifetime (one-off, as in timed process interventions). Sequences of (repeated) actions are another, more complicated, setting.

Experiments

We compare online RL to CI using synthetic datasets and rely on the same LSTM neural network (NN) architecture (except for the last layer) for both methods.

We experiment with binary actions (interventions) that allow for clearer insights without loss of generalization. As


3
ITE = $Y_I - Y_{NI}$ = 3 – .5 (5 +0) = .5
I: 33%
NI: 67%
5
ITE = $Y_I - Y_{NI}$ = 5 – 0 = 5
I: 50%
NI: 50%
0
3
Q_1(I) = 3
Q_1(NI) = max(5, 0) = 5
I
NI
5
Q_2(I) = 5
Q_2(NI) = 0
I
NI
0

(a) CI: threshold (TH) = 0 → intervene at 1st event; TH = 2 → do intervene at 2nd event.

(b) RL: optimal policy follows max(Q): intervene not at 1st but at 2nd event.

Figure 1: Simple process to compare CI to RL. Both agree on the policy at the second event. At the first event, CI correctly estimates the outcome for intervention (Y_I), while the prediction model for the outcome for non-intervention (Y_{NI}) will observe two different outcomes (5 and 0) and **summarize** those to 2.5. This value depends on the loss function and the samples' distribution (percentages in the graph) in the training set which itself depends on the data-gathering policy. In contrast, RL selects the **maximum** of the two Q values in the second event.

CI struggles with action sequences, we further simplify by opting for one-off actions (timed interventions) to enable the CI-RL comparison; the RL method, however, can be extended to sequential actions without modification.

We work with two synthetic processes, as shown in Figure 2, to create both the offline dataset for CI and the online environment required for online RL as well as to compute counterfactuals unavailable in historical data.

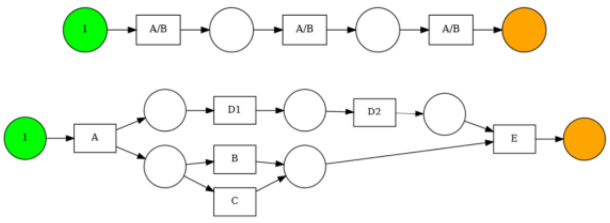


Figure 2: Synthetic generative Process_1 (top) and Process_2 (bottom)

.

Knowing the counterfactuals allows for correct evaluations of the experiments. Moreover, given a sufficiently small state space, a perfect policy can be derived and used to judge the absolute performance of methods. The values of the three activities and attributes in Process_1 are entirely random, whereas activities in Process_2 are governed by the given structure, with the attributes and case variables entirely stochastic as well. An intervention multiplies the value of the current (Process_1) or next (Process_2) attribute by a constant. The outcomes for both processes are the sums of all their respective attributes after flipping the sign of the changed attribute if no A occurred in Process_1 or Process_2 went through its C branch.

We summarized our experimental results in Table 1. RL clearly outperforms CI for both processes: not only are the mean scores significantly higher, but also the standard deviations of the RL scores are much lower, making RL by far the more robust method. Most or all of this outperformance can be attributed to RL's intrinsic ability to approach the optimal policy. The fact that online RL permits exploring all parts of the state space plays virtually no role here, as the CI training sets in our experiments were made large enough to contain the complete state space. Were this not the case, the observed CI-RL divergence would certainly widen.

Table 1: Experimental results comparing CI, RL, a perfect and random policy for two processes.

		Uplift		Computational effort		
		Mean	**StDev.**	**Unit**	**Mean**	**StDev.**
Process_1	CI	1,526	36.8	epochs	188	27.6
	RL	1,616	5.0	transitions	6,000	2,549
	perfect	*1,651*				
	RCT	*-515*				
Process_2	CI	1,682	203.1	epochs	212	51.7
	RL	1,806	34.7	transitions	26,800	10,628
	perfect	*1,845*				
	RCT	*-50,336*				

RL comes within 3% of the perfect policy results for both processes (1,616 against 1,651 and 1,806 against 1,845 respectively). When comparing to the RCT results, it also becomes apparent that CI does manage to improve considerably upon the policy that originally created the dataset (in this case entirely random interventions).

Having set both RL's memory and CI's batches to equal size, one NN optimization step involves the same number of samples per transition for RL as per epoch for CI. Table 1 shows that RL's computational requirements are two orders of magnitude higher that those for CI.

Conclusion and Future Work

Our experiments reveal that RL's policies outperform those from CI and are more robust at the same time. Indeed, the RL policies approach perfect policies. Unlike CI, the unaltered online RL approach can be applied to other, more generic PresPM problems such as next best activity recommendation. Nonetheless, CI has its merits in settings where online learning is not an option. Future work includes a similar comparison with incomplete datasets and offline RL for which techniques exist and indirect, two-state CI in which first a number of possible suffixes is predicted before an outcome is estimated for each of those to determine ITE(s).