ProLift: Automated Discovery of Causal Treatment Rules From Event Logs (Extended Abstract)

Zahra Dasht Bozorgi[1], Aleksei Kopõlov[2], Marlon Dumas[2], Marcello La Rosa[1] and Artem Polyvyanyy[1]

[1]*University of Melbourne, 700 Swanston St, Carlton, VIC 3053, Australia*
[2]*University of Tartu, Narva mnt 18, 51009 Tartu, Estonia*

Abstract
ProLift is a Web-based tool that uses causal machine learning, specifically uplift trees, to discover rules for optimizing business processes based on execution data (event logs). ProLift allows users to upload an event log, to specify case treatments and case outcomes, and to visualize treatment rules that increase the probability of positive case outcomes. The target audience of ProLift includes researchers and practitioners interested in leveraging causal machine learning for process improvement.

Keywords
process mining, causal machine learning, rule discovery

1. Introduction

Causal Process Mining is an emerging sub-field at the intersection of process mining and machine learning that seeks to develop methods to discover and quantify causal-effect relations by analyzing event logs. For decision making, recommendations based on causal relationships generalize better than purely predictive models because predictions can be made from spurious correlations that might not hold in future traces [1]. Therefore, we discover causal rules from event logs that serve as recommendations to end users for optimizing the process outcome. For example, in the context of a loan application process, loan managers can use these recommendations to implement on-the-fly interventions aimed at maximizing the likelihood of a loan applicant accepting an offer for a loan.

In this paper, we present ProLift, an open-source Web-based causal rule discovery tool that helps decision makers to identify sub-groups of process cases that may benefit from a given intervention (a.k.a. treatment). We describe the architecture of the tool, provide an example scenario, and discuss the maturity and limitations of the tool.
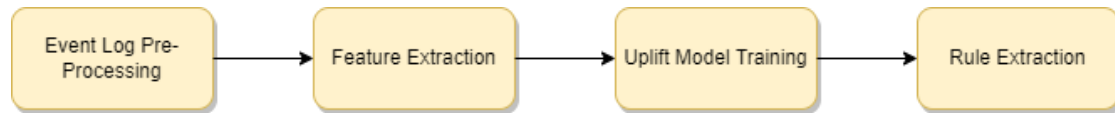
Figure 1: ProLift's functional components

2. Architecture

ProLift is a Web application that implements the Causal Rule Discovery method proposed in [1]. It takes an event log as input, preprocesses the log, extracts relevant features, trains a causal model, and finally extracts rules. ProLift has four functional components: the Event Log Pre-processing Component, the Feature Extraction Component, the Uplift Model Training Component, and the Rule Extraction Component (fig. 1). We describe each component below.

2.1. Event Log Pre-Processing

The input to ProLift is an event log in which events have at least these three features: unique case identifier, activity name, and activity end timestamp. ProLift parses the log, ensures the requirements are fulfilled, and performs data cleaning steps, such as the removal of duplicate events.

2.2. Feature Extraction

This component takes the pre-processed log and extracts feature vectors. The feature selection takes into account the user's preferences (i.e., which attributes to ignore in model training). For event-level features, the tool uses the encoding methods commonly used in process mining (e.g., aggregation or last-state encoding) [2], and one-hot encoding for categorical features.

2.3. Uplift Model Training

This component uses feature vectors to build a causal model. Specifically, we use uplift trees as our modeling algorithm because the conversion of trees to rules is straightforward, and also, uplift trees were designed to deal with binary treatments and outcome settings, which is an appropriate setting for optimizing process outcomes via a single treatment. In addition, uplift trees can deal with biases present in observational data. We use the CausalML library [3] to build uplift trees.

2.4. Rule Extraction

In this component, we extract rules from the tree by tracing each path in the uplift tree from root to leaf. Each split in the tree will narrow down the sup-group of cases. Each leaf of the tree provides an uplift estimate, which we interpret as the probability of the treatment changing the outcome.

3. Example

As an example process, we consider a loan application process. A loan application process is successful if the customer is happy with the loan offer made to them and accepts it. There are many treatments that one can apply to increase the rate of success in this process. For example, one treatment is to increase the number of loan offers if the customer is not happy with the first one. In this example, the number of offers is a controllable attribute. Another possible treatment is to decrease the monthly cost of the loan to make it more attractive to the customer (e.g., by offering a lower interest rate). In this case, the monthly cost is a controllable attribute.

Project Configuration Page: Not all attributes in the event log are controllable by the organization that runs the process. For instance, the customer determines the loan goal (e.g., home loan) and as such, this is an uncontrollable attribute for the organization. In ProLift, after the user uploads an event log, they are taken to the project configuration page, where they can specify the types of attributes. The following attribute types are supported: uncontrollable, controllable, outcome, group identifier, ignore, and order by (e.g., event number).

The user can also specify the data type for each attribute. The next step for the user is to specify the outcome and treatment rules. The outcome rule is defined in the outcome configuration section. For example, in our loan application example, we define the outcome as positive if the column called *Selected* is equal to True. Similarly, in the treatment configuration section, we define the treatment rule. For example, for the treatment related to sending multiple offers, the tool user can specify that the treatment is the occurrence of activity *Send Offer* more than once within a case.

Model Configuration Page: Next, the user is presented with five sections on the model configuration page (Fig. 3). In the Data Analytics section, two stacked bar charts are shown (Fig. 4). These charts give an overview of the data with respect to the treatment and outcome rules that have been specified in the project configuration page. The first graph illustrates the stacked bar of the confusion matrix with respect to case length. In other words, the x-axis indicates the size of an individual case in a given log while the y-axis is a confusion matrix that depicts how many cases of the same length had a positive or negative outcome, how many had a treatment present, and how many did not. The second graph illustrates a relation between positive/negative outcomes and the number of interactions before the application of a treatment. In other words, the x-axis indicates the number of actions/steps that have been taken before all the treatment rules have been applied, and the y-axis indicates the outcome of a case. The Data Balancing section allows end users to balance their data before model training. For instance, they can specify what percentile of the data should be used for model training or what is the maximum percentage difference in the confusion matrix values. The next tab is the Confusion Matrix which displays the confusion matrix of the data used in model training according to the specified balancing settings (Fig. 5). In the Model Settings section, the user can specify the configuration of the causal model. After the model is built and the rules are extracted, the rules will be displayed to the user in the Results section. The rules indicate sub-groups of cases that will benefit from the specified treatment and sub-groups for which the treatment should be avoided. Fig. 6 shows an example of the generated rules for sending multiple offers as a treatment.

Figure 2: Project configuration page example

Figure 3: Model configuration page example

4. Maturity

We validated ProLift on a log of a loan origination process (BPI Challenge 2017). We compared the discovered rules w.r.t. the findings of submissions to this challenge [1]. We selected this dataset because it is relatively large, there is a clear positive outcome (loan offer accepted), and several observed treatments. We also conducted tests on similar datasets, including a purchase-to-pay log (BPI Challenge 2019). ProLift takes less than two minutes to handle these logs. To further validate the tool, we plan to conduct case studies with domain experts.

The source code of ProLift, together with installation instructions via a Docker container, are available at: https://gitlab.com/aleksei.kopolov/action-recommendation. An online demonstration version of ProLift is available at: https://prolift.cloud.ut.ee. A video demo of ProLift can be found at https://youtu.be/rAfbx-nsR9I.
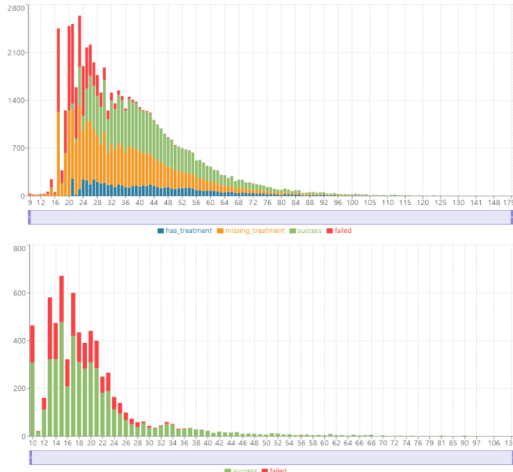
Figure 4: Data analytics bar charts
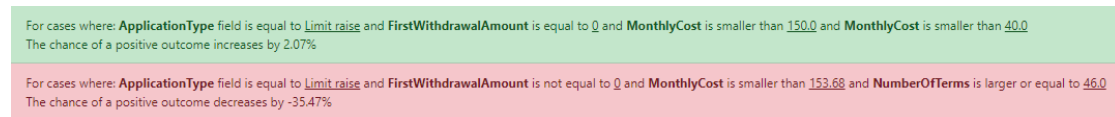

Figure 5: Confusion matrix

Figure 6: Example rules

5. Conclusion and Future Work

ProLift discovers rules for increasing the success rate of a process, based on causal relationships between treatments and case outcomes. The current version of ProLift supports only one causal modeling technique, namely uplift trees [4]. In the future, we will add support for other causal models, including meta-learners. Another limitation is that ProLift currently only supports recommendations for optimizing a binary process outcome. Another direction for future work is to add support for numerical outcomes, such as case duration.

References

[1] Z. D. Bozorgi, I. Teinemaa, M. Dumas, M. L. Rosa, A. Polyvyanyy, Process mining meets causal machine learning: Discovering causal rules from event logs, in: 2nd ICPM, 2020.

[2] Z. D. Bozorgi, I. Teinemaa, M. Dumas, M. L. Rosa, A. Polyvyanyy, Prescriptive process monitoring for cost-aware cycle time reduction, in: 3rd ICPM, 2021.

[3] H. Chen, T. Harinen, J.-Y. Lee, M. Yung, Z. Zhao, Causalml: Python package for causal machine learning, 2020. `arXiv:2002.11631`.

[4] P. Rzepakowski, S. Jaroszewicz, Decision trees for uplift modeling with single and multiple treatments, Knowl. Inf. Syst. (2012).