Plugin Manual
Causality Inference Using Structural Equation Model

Mahnaz S. Qafari

Rheinisch-Westfälische Technische Hochschule Aachen(RWTH), Aachen, Germany
m.s.qafari@pads.rwth-aachen.de

1 Introduction

One of the main purposes of using process mining is to enhance processes to achieve higher performance. Improving process performance indicators can result in a great amount of money-saving and higher service quality. The first step of enhancing a process is identifying friction points. The second step is finding the root causes of each friction point and estimating the possible effect of changing each factor on the performance of the process. The final goal is planning process enhancement actions and then re-engineering the process. But doing process enhancement using the results of currently available tools for root cause analysis is prone to making even more problems. This occurs as the current techniques are based on machine learning techniques such as clustering or rule mining, which are designed to find correlations, not causation. Here the vital task is distinguishing if the correlation between features of a process signifies a causal relationship between them or just a mere correlation (i.e., there are some common causes for both of them).

The “causality inference using structural equation model” plugin is an interactive plugin implemented in ProM. Given a target and a set of descriptive features, the goal of this plugin is to find the (possible) statistically supported causal relationships among the features in the event data (gathered from an event log) and then estimating the strength of the causal effect of each one of features on the target feature. The target feature is mainly related to an observed problem in the process. Moreover, the outcome of this plugin can be used to predict the effect of an intervention on the target feature. Please note that, we may use the terms “feature” and “variable” and also the terms “case”, “trace”, and “process instance” interchangeably.

If you are interested in answering a question in one of the following forms, then this plugin can help.

– What is the causal structure of the features of my process?
– Does a specific feature have any influence on the target feature, and if so, what is the magnitude of the effect?
– Which features have the causal effect on the target feature?
– What is the effect of setting one of the descriptive features (variables) to a specific value (intervention on some variable) on the target feature?

To answer these sorts of questions about a process, we need to go through the following steps:

1. specifying the target feature and the descriptive features and then extracting the data from the event log on the process,
2. discovering the causal structure of the features,
3. estimating the causal effect of each feature on the target feature.

The result of these three steps is a *structural equation model* that can be further used to anticipate the effect of intervention on different features. A structural equation model is a set of equations that encode the observational and international distributions [1]. For more information on causal inference, please refer to [1,2].

Considering the complexity of processes, it is easy to imagine cases where the process expert does not have concrete enough information on which descriptive feature to use for causal inference. To help users in such complex cases, this plugin also include several feature recommendation methods that not just recommend features with possible highest causal effect on target feature (and consequently intervention on them have the most potential effect on improving the problem), but also those values of them that possibly contribute the most to the problem.

In the sequel, we describe how to do the setting and perform each of these three steps using the implemented plugin on an event log.

Running example (What is the problem?). Download the *receipt*[a] event log here. This is a real event log gathered from different municipalities in Netherlands. Suppose that some of the cases (process instances) have been delayed in this process. Moreover, the acceptable duration of a case is 3 percent of the maximum duration of all the recorded cases (Utterly, some of the cases are too delayed ☹.). Also, the process manager has told us that the only possible causes of these delays are the employee working on one of the three main activities in each trace.

[a] This data originates from the CoSeLoG project executed under NWO project number 638.001.211.

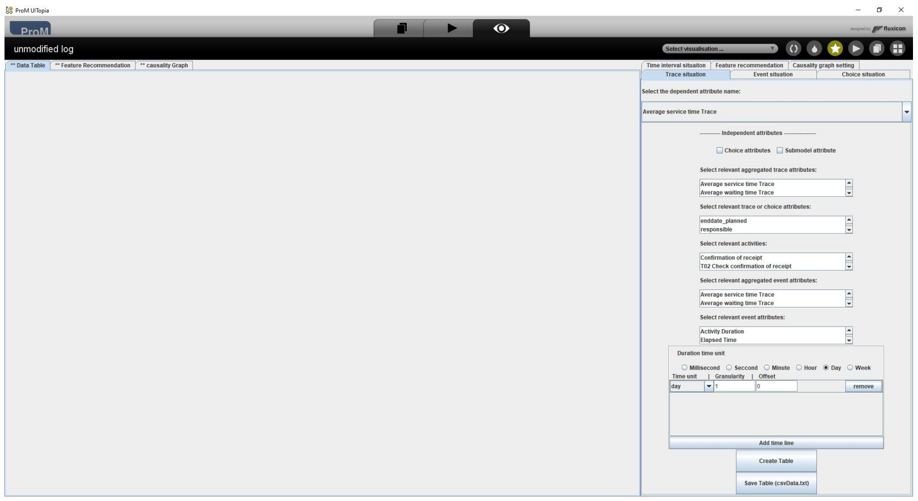
2 How to use the plugin

The inputs of this plugin are the event log, the Petri-net model of the process, conformance checking results from replaying the traces on the given Petri-net model. These inputs are used to enrich the event log and extract tabular data. Then, the extracted data table is used for causal discovery purposes. Besides the features that already exists in the event log, some of the driven features that can be used while extracting the data are as follows:

- **Aggregated features**, process workload, average service time (of trace or event), number of active event, resource workload, and average resource service time (of a specific resource). Each aggregated feature has three parameters including time unit (week, day, hour, minute, second, millisecond), granularity, and offset, where granularity and offset is defined as a factor of its time unit.

– **Other features**, deviation, number of log move, number of model move, trace duration, trace delay (in trace level), event duration, elapsed time, and remaining time.

Let's start. Download the latest version of the nightly built of ProM from here. Run "PackageManager.bat" and add "causal inference using structural equation model" to ProM. From now on, we call this plugin "causal inference" plugin. Run "ProM.bat" and import the receipt event log.
Use *inductive miner* with the default setting to discover the Petri-net. Now, use "replay a log on Petri-net for conformance analysis" for the replay results. Please use the "event name" as the classifier. Now use these three as the input of "causal inference" plugin. The UI of the plugin (current version) would be as shown in Figure 2



Select the dependent attribute name: Average service time Trace

----------- Independent attributes -----------------

☐ Choice attributes ☐ Submodel attribute

Select relevant aggregated trace attributes:
Average service time Trace
Average waiting time Trace

Select relevant trace or choice attributes:
enddate_planned
responsible

Select relevant activities:
Confirmation of receipt
T02 Check confirmation of receipt

Select relevant aggregated event attributes:
Average service time Trace
Average waiting time Trace

Select relevant event attributes:
Activity Duration
Elapsed Time

Duration time unit
Millisecond Seccond Minute Hour [✓ Day] Week

Time unit | Granularity | Offset
day | 1 | 0

Fig. 1. The first UI of the causal inference plugin.

As it is shown in Figure 2, the UI of this plugin is divided in two main part: the *right panel* which is for the setting and the *left panel* which is for displaying the outputs.

2.1 Creating a table

In this plugin, we focus on problems related to performance, compliance, decision making, or one of the aggregated characteristics of the process (where the target feature is an aggregated feature defined with a specific time interval and time span, we call this type of target features *time interval situations*). Consequently, we classify the possible problems (and therefore target features) in four types, which we call the *situations*. A target feature can indicate a trace, event, choice, or time interval situation. In the following we provide some examples of different situations.

- Trace situation. We may want to know why deviations happen in some cases.
- Event situation. We might be interested in discovering what causes the bottleneck in an event.
- Choice situation We are interested to find out which factors influence the decision made in a choice place.
- Time interval situation. We are concerned about the causes of the high workload of the process where the workload is measured in a daily basis.

The situation type of the target feature highly influence the data extraction method (i.e., the target feature concerns an event situation). For example, if the target feature captures the occurrence of a problem relate to a specific event (or similarly the decision made in a choice place), then the problem may happens several time in a given trace. As a result, this given trace is mapped to several row in the table (one row for each occurrence of the problem in that trace). Moreover, the data in each row is extracted from that prefix of the trace that happened before the occurrence of its corresponding problem. Please note that this method of data extraction is important as we concern the causes of the target feature and in causal inference theory, the time precedence of cause and effect (indicating that cause must happen before the effect) is one of the most important properties of a causal relationship.

2.2 Selecting target feature

Related to these four types of situations, there exists four tabs in the right panel; *trace situation*, *event situation*, *choice situation*, and *time interval situation*. The main difference of these four tabs is the target feature selection procedure.

Trace situation tab. In this tab, you can select one of the trace features as the target feature.

Event situation tab. Here, you can select a feature of a group of the events in the event log as the target feature. By a group of events, we mean a set of events that share the same value(s) for a specific feature. The target feature can be selected using the first drop-down list and the group of events can be selected using the second drop-down list.

Examples of event groups. For example, related to receipt process, a group on events can be a set of events that
- are done by “resource23” or “resource11”,
- with activity name “confirmation of receipt”,
- have been started on Mondays between 8:00 and 9:00,
- has duration more than one hour and less than two hours.

Choice situation tab. In this tab, you can select one of the choice places on the Petri-net model as the target feature.

Choice situation tab. In this tab, you can select an aggregated features of the process as the target feature. To select the target feature, this tab includes three drop-down lists (menu). Using the first drop down list, you can select the level of the target feature; including process level, trace level, event level, and resource level. By the second drop down list, you can select the target feature name; like workload, average waiting time. In case of selecting event or resource level in the first drop-down list, using the third drop-down list, you can choose the set of events (though specifying activity names) or resources that you are concern about.

2.3 Selecting descriptive features

The descriptive feature selection procedure is the same for trace, event, and choice situation tabs which includes two main parts, descriptive feature selection part and time setting part.

Descriptive feature selection part. This part includes two check boxes and five drop-down lists as follows:

- Choice feature checkbox. By checking this checkbox, the Petri net model of the process would be demonstrated on a window and you can select some of the choice places on it. The choice made in these choice places will be considered as descriptive features.
- Sub-model checkbox. By checking this checkbox, the Petri net model of the process would be demonstrated on a window as well. You can select a sub-model of the process by selecting some of its transitions. The duration of the selected sub-model will be considered as one of the descriptive features.
- Relevant trace aggregated features. Using this drop down list, you can select some of the trace level aggregated features.
- Relevant trace features. Using this drop down list, you can select some of the trace level attributes (such as trace throughput, department, age, customer type) as descriptive features.
- Relevant activities. This drop down list can be used to specify the set of activity names of those events that we are interested to their properties.
- Relevant event aggregated features. This drop down is dedicated to selecting some of the event level aggregated features.
- Relevant event features. This drop down list is dedicated to selecting some of the event level attributes (such as event duration, timestamp, resource) as descriptive features.

Time setting part. In this part, a time unit can be set for measuring non aggregated descriptive features such as duration of events or the throughput of traces. Moreover, for aggregated features, there are the possibility of setting several timelines. To define a timeline, you need to specify its time unit, granularity, and offset. Please note that if you define k timelines, then there would be k values for each selected descriptive aggregated feature in the extracted table. If the target feature is also an aggregated feature (in case of time interval situation), then the value measured using the first defined timeline would be considered as the value of the target feature.

Running example (Table setting).

1. Use the *trace situation* tab.
2. Select *trace delay* in the *select the dependent attribute name* drop down list.
3. Set the delay threshold to 0.03.
4. Select
   - Confirmation of receipt,
   - T02 Check confirmation of receipt,
   - T04 Determine confirmation of receipt,

   in the *select relevant activities* drop down list.
5. Select *resource* in the *select relevant event attributes* drop down list.
6. Create the table using the *Create Table* button.

2.4 Feature Recommendation Tab

This tab provide several feature value selection methods to help identifying the relevant descriptive features. This is relevant specially in case of dealing with complex processes. The feature selection methods including SFVPR method[1], random forest based method, information gain based method, and correlation based method. Please note that for some of these methods some extra setting might be needed. For example, in case of SFVPR and information gain based method a threshold and in case of random forest based methods the number of decision trees are needed to be set. The other settings that are needed for feature recommendation are as follows:

- Number of bins if the data includes numerical data.
- The set of undesirable target feature values if the target feature is nominal and a threshold if the target feature is numerical.

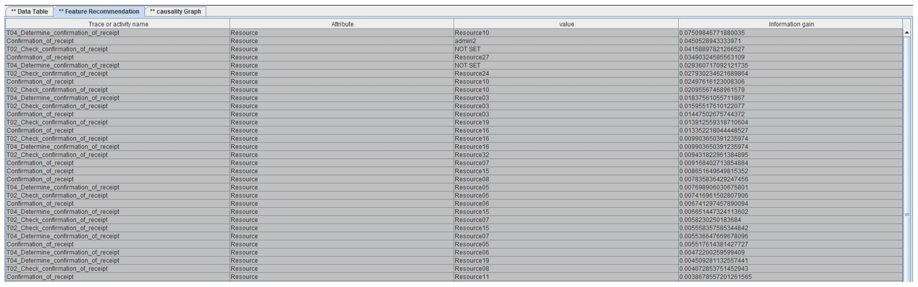
If you want to do causal inference just using the recommended features, you can use *Create feature with recommended features* button.

Please note that using SFVPR method, not just the subset of descriptive features with a possible causal effect on the target features are recommended, but also those values of them that possibly have the highest effect on the undesirable outcome are identified. Figure 2.4 shows an example of the output of this method. In this example, the target feature is "trace delay" and the undesirable value is "delayed". The pair of feature, feature values are sorted in the decreasing order of information gain which means that the earlier pairs is the list have (potentially) the highest effect on the undesirable outcome. We can interpret the first line of this table as follows: the information gain of trace delay is 0.075 where dividing by resource of "T04 Determine confirmation of receipt" is "Resource10" or not.

2.5 Discovering Causal Structure

Assumptions To use the implemented plugin for discovering the causal equation model of the data, the following assumptions in the data have to be met.

[1] Please refer to [3] for more information on this method.



Trace or activity name	Attribute	value	Information gain
T04_Determine_confirmation_of_receipt	Resource	Resource10	0.07509846771880035
Confirmation_of_receipt	Resource	admin2	0.0450528943333971
T02_Check_confirmation_of_receipt	Resource	NOT SET	0.04158897821266527
Confirmation_of_receipt	Resource	Resource27	0.03490324585563109
T04_Determine_confirmation_of_receipt	Resource	NOT SET	0.029360717092121735
T02_Check_confirmation_of_receipt	Resource	Resource24	0.027930234621689864
Confirmation_of_receipt	Resource	Resource10	0.02497616123008306
T02_Check_confirmation_of_receipt	Resource	Resource10	0.02095567468961579
T04_Determine_confirmation_of_receipt	Resource	Resource03	0.01837561055711867
T02_Check_confirmation_of_receipt	Resource	Resource03	0.01595517610122077
Confirmation_of_receipt	Resource	Resource03	0.01447502675744372
T02_Check_confirmation_of_receipt	Resource	Resource19	0.013912559318710604
Confirmation_of_receipt	Resource	Resource16	0.013352218044448527
T02_Check_confirmation_of_receipt	Resource	Resource16	0.009903650391235974
T04_Determine_confirmation_of_receipt	Resource	Resource16	0.009903650391235974
T02_Check_confirmation_of_receipt	Resource	Resource32	0.009431822961384895
Confirmation_of_receipt	Resource	Resource07	0.009168402713854884
Confirmation_of_receipt	Resource	Resource15	0.008651649649815352
Confirmation_of_receipt	Resource	Resource08	0.007835836429247456
T04_Determine_confirmation_of_receipt	Resource	Resource05	0.007698906030675801
T02_Check_confirmation_of_receipt	Resource	Resource05	0.007416961502807906
Confirmation_of_receipt	Resource	Resource06	0.006741297457890094
T04_Determine_confirmation_of_receipt	Resource	Resource15	0.006651447324113602
T02_Check_confirmation_of_receipt	Resource	Resource07	0.0058230250183684
T02_Check_confirmation_of_receipt	Resource	Resource15	0.005558357585344842
T04_Determine_confirmation_of_receipt	Resource	Resource07	0.005536647669678096
Confirmation_of_receipt	Resource	Resource05	0.005517614381427727
T04_Determine_confirmation_of_receipt	Resource	Resource06	0.00472200259599409
T04_Determine_confirmation_of_receipt	Resource	Resource19	0.004509281132557441
T02_Check_confirmation_of_receipt	Resource	Resource08	0.004072853751452943
Confirmation_of_receipt	Resource	Resource11	0.0038678557201261565

Fig. 2. The first few rows of the list of feature and feature value pairs recommended by SFVPR method for the running example.

1. Cases (situations) in the data are independent and identically distributed.
2. All the variables that has an influence on the target feature are measured. In other words, there is no hidden cause in the model.
3. The causal Markov condition holds among the variables. This condition states that a variable is independent of other variables except its descendants, given those variables that are its direct causes (parents). It expresses a form of local causality, which means other variables can not provide more information about a given variable given its direct causes and its descendants.
4. The causal faithfulness condition holds, which indicate that all the independence relationships among the measured variables are implied by the causal Markov condition.
5. There is no missing data. There are two main types of missing data in the context of the process mining, completely random missing values which are resulted by errors, and not random missing values that, regarding the logic of the process, should be missing (like when we are trying to see what is the resource of an activity in a trace where that activity did not and must not happen in that trace.). Here we assume that both types of missing values have been removed from the data in the pre-processing phase by for example removing the columns and rows that contain missing values (more than a threshold) and then imputing the remaining missing values.
6. There is no selection bias. This means that the chance a case (situation) had happened in the process did not depend on the values of any of the measured variables in the data.
7. There are no feedback cycles among the measured variables.

structural causal discovery setting The *structural causal graph* is a directly acyclic graph(DAG) in which the nodes are the set of variables and there is an edge from nodes U to node V, if U is a direct cause of V.To see what is the causal structure of the data, you need to specify if your data is discrete or continuous. Now, you have two options, starting with an empty graph and adding edges(causal directions) manually; or, running a search algorithm and then modify the structural causal graph that has been discovered. Also you can influence the search result by adding your knowledge about your process in the form of *forbidden* and *required* directions. Note that if you add an edge as a forbidden direction, then you just restrict the causal structure from having that direction, but there might be other type of edges between those two attribute (Which do not contradict that declared forbidden direction). Also, If you add a required direction, still there is the possibility that it does not be included in the causal graph resulted by the search algorithm. The reason is that in this plugin, the GFCI algorithm has been used as the search algorithm and this algorithm has two phase, forward and backward. In the backward phase, some edges are removed if their removal improves some score. Note that, if you want to enforce the structure of the graph, the best option would be adding the desired causal directions as required edges and then using *generate the final graph* button to enforce the graph structure.

Also, discovering forbidden directions that are supported by the event log, is possible by using *Add forbidden directions supported by event log* button. This method adds the forbidden direction $a \rightarrow b$ to the knowledge if b is an attribute related to an event that never happened after the activity that attribute a belong to it in the event log.

- If you want to force a direction, add it as a knowledge graph and then use the *Generate final graph* button. This button merge the knowledge direction and the last graph resulted by a search (if exists).

- You can see the knowledge graph using ”view knowledge graph” button to see which edges has been added as forbidden or required directions. But, the knowledge graph is not interactive and modifiable. So, if you need to modify them you need to use the graph in ”causality graph” panel on the right side.

- If you add two contradicting edges as knowledge, like a forbidden and a required edge between two nodes in the same direction, just the last one is considered.

- We suppose that all the missing values has been removed from the data in the preprocessing phase. In the case of continuous values, we have implemented some simple missing value imputation methods that can be used in case of using data with missing values. In case of literal attributes, the null values are replaced with a new category, *NOT SET*, which is the best option for missing values that happen due to absence of an activity that it should not have been happened in the corresponding situation. Note that replacing missing value imputation may delude the results.

Running example (Causality graph setting).

1. Set *data type* to *discrete*.
2. Run the search algorithm by *search* button. The result is shown in Figure 2.5.a.
3. Use the *Add forbidden directions supported by event log* to add event log supported forbidden directions. The result is shown in Figure 2.5.b.
4. You can see a graph including the edges added as knowledge as shown in Figure 2.5.c using *View knowledge graph* button.
5. Select the *use knowledge* check box and run the search again. The result would be as shown in 2.5.d.
6. The graph is not not a DAG. Add two required edges to the graph. The edges that you need to add are in green in 2.5.e.
7. Enforce the structure of the graph by using *generate final graph* button. The result is shown in Figure 2.5..

In the sequel, we assume that the set of independent variables is $\boldsymbol{V} = \{V_1, \ldots, V_n\}$ and the class variable is C. Also, we denote the set of parents of variable $V \in \boldsymbol{V} \cup \{C\}$ by $\boldsymbol{PA}(V)$.

3 Estimation

The final step of discovering the causal model is estimating the strength of the causal effect of each descriptive feature, say $V \in \boldsymbol{V}$, on the target feature, C, using the observed data. To perform this step, The causal structure graph should be a directly acyclic graph. Also, we assume that all the missing values have been removed from the data in the pre-processing phase. Now, we just need to compute the effect of V on C condition on its parents, $\boldsymbol{PA}(V)$. We mention how the strength of the causal effect is estimated when the data is contentious or discrete.

Continuous Data In a given data table, we assume the linear dependency between the attributes. Also, we suppose that the attribute values are taken by a Gaussian distribution.

$$V_i = \sum_{V_j \in \boldsymbol{PA}(V)} \alpha_{ij} V_j + \mathcal{N}_{V_j}, \quad 1 \leq i \leq n,$$

$$C = \sum_{V_j \in \boldsymbol{PA}(V)} \alpha_j V_j + \mathcal{N}_{V_j}.$$

We can represent the CSM graphically by consider α_{ji} as the coefficient of the edge $V_j \rightarrow V_i$ in the corresponding DAG. Thus to estimate the magnitude of the effect of $V \in \boldsymbol{V}$ on the C, it is enough to sum the multiplication of the coefficient of the edges in each directed path from V to C.

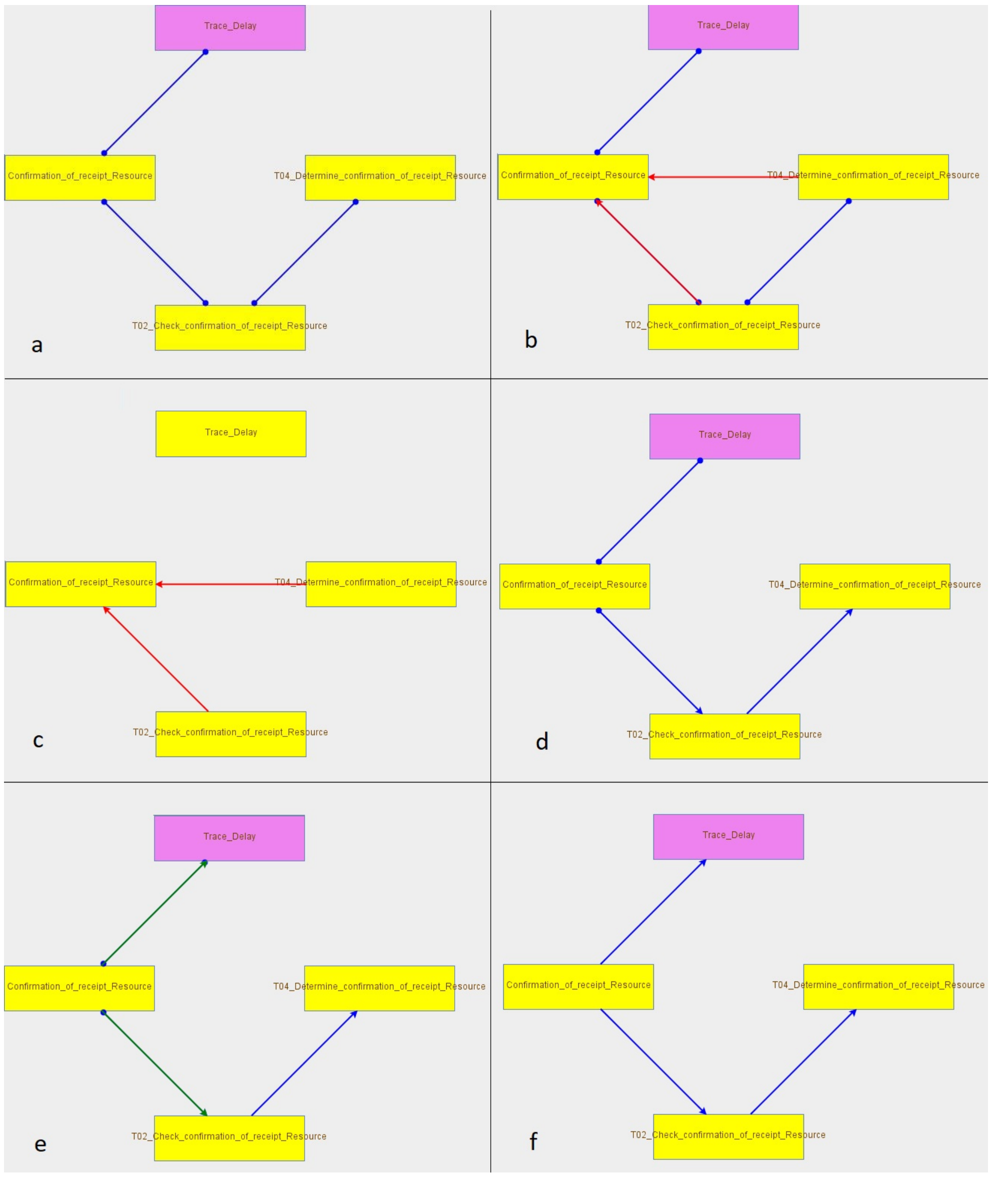

Trace_Delay
Confirmation_of_receipt_Resource
T04_Determine_confirmation_of_receipt_Resource
T02_Check_confirmation_of_receipt_Resource
a
b
c
d
e
f

Fig. 3. The causal structures resulting by doing the running example.

Discrete Data Suppose $values(C) = \{c_1, \ldots, c_m\}$ and $v_i \in values(V)$. We demonstrate the effect of intervention on v by setting it as $V = v_i$ on the probability distribution of the values of c as $P_{(do\ V=v_i)(c)}$[2] and estimate it as follows:

$$P_{(do\ V=v)}(C) = \sum_{\boldsymbol{v}} P(c|v, \boldsymbol{v})P(\boldsymbol{v})$$

where $\boldsymbol{v} = \{v_1, \ldots, v_n\}$ such that $v_i \in values(V_i)$, $V_i \in \boldsymbol{PA}_V$. Note that this probability distribution is different from $P(C|(V = v))$.

Running example (Estimation). Use the *estimate* button and then click on the nodes in the causal structure graph.

4 How to interpret the results

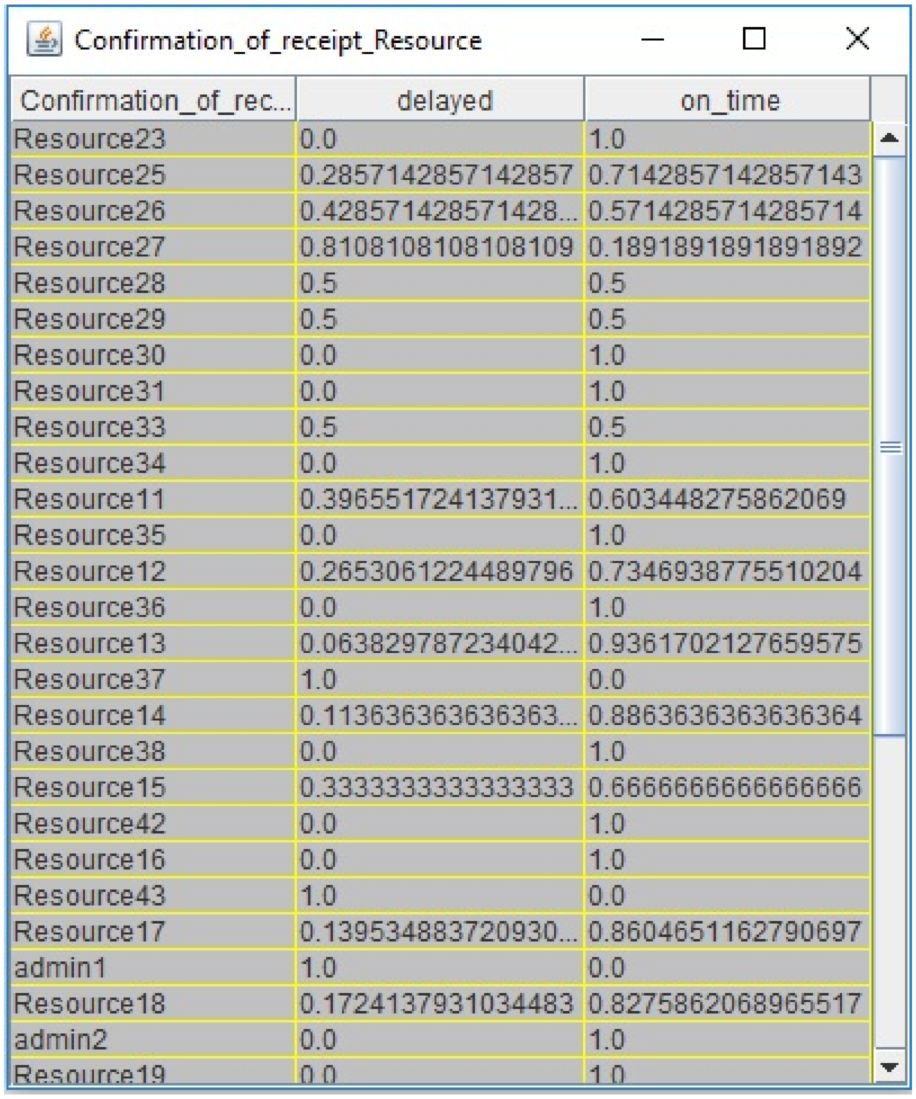
Imagine that the assumptions mentioned in Section 2.5 are satisfied and the discovered causal structure includes all the influential variables on the class variable. After doing the estimation, if you click on a node, say V, then you can see if that variable has any effect on the target feature, or not. If V and C are independent, in the discrete case, then it will be notified using a message and, in the continuous case, the coefficient of the equation that demonstrate the effect of V on C would be zero. Other wise, in the discrete case, a table denoting the effect of setting V to each possible value is Shown. For example, the table demonstrated in Figure 4 is the table resulted from doing estimation in the running example. We can see that if we set "$Confirmation\_of\_receipt\_Recourse$" $= Recourse15$, then we can expect that we observe delay in $1/3$ of traces.

In the continuous case, after estimating the strength of the discovered causal relationships, a coefficient will appear on each edge of the DAG. The coefficient of edge $U \rightarrow V$ means how strong is the effect of variable U on V, condition on its parents. If you click on a node in the DAG a window will show up mentioning the effect of that variable on the class variable, and some statistical information about it (including its mean, median, variance, and standard deviation). An example of such a window is given in Figure 4. By this estimation, we expect that if the duration of activity Confirmation_of_receipt increases one unit and all the values of other variables remain the same (we intervene just on activity Confirmation_of_receipt), then we expect that the duration of activity T10_Determine_necessity_to_stop_indication will increase about 0.59 unit.

References

1. Pearl, J.: Causality. Cambridge university press (2009)
2. Pearl, J., Mackenzie, D.: The book of why: the new science of cause and effect. Basic books (2018)
3. Qafari, M.S., van der Aalst, W.: Feature recommendation for structural equation model discovery in process mining. arXiv preprint arXiv:2108.07795 (2021)

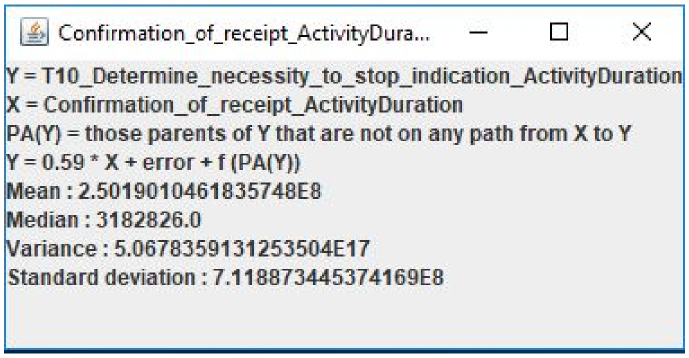
[2] We borrow this notation from [**?**].



Confirmation_of_receipt_Resource

Confirmation_of_rec...	delayed	on_time
Resource23	0.0	1.0
Resource25	0.2857142857142857	0.7142857142857143
Resource26	0.428571428571428...	0.5714285714285714
Resource27	0.8108108108108109	0.1891891891891892
Resource28	0.5	0.5
Resource29	0.5	0.5
Resource30	0.0	1.0
Resource31	0.0	1.0
Resource33	0.5	0.5
Resource34	0.0	1.0
Resource11	0.396551724137931...	0.603448275862069
Resource35	0.0	1.0
Resource12	0.2653061224489796	0.7346938775510204
Resource36	0.0	1.0
Resource13	0.063829787234042...	0.9361702127659575
Resource37	1.0	0.0
Resource14	0.113636363636363...	0.8863636363636364
Resource38	0.0	1.0
Resource15	0.3333333333333333	0.6666666666666666
Resource42	0.0	1.0
Resource16	0.0	1.0
Resource43	1.0	0.0
Resource17	0.139534883720930...	0.8604651162790697
admin1	1.0	0.0
Resource18	0.1724137931034483	0.8275862068965517
admin2	0.0	1.0
Resource19	0.0	1.0

Fig. 4. A sample table that shows the effect of setting independent variable ”Confirmation_of_receipt_Recourse” on ”Trace_Delay” (the class variable).



```
Y = T10_Determine_necessity_to_stop_indication_ActivityDuration
X = Confirmation_of_receipt_ActivityDuration
PA(Y) = those parents of Y that are not on any path from X to Y
Y = 0.59 * X + error + f (PA(Y))
Mean : 2.5019010461835748E8
Median : 3182826.0
Variance : 5.0678359131253504E17
Standard deviation : 7.118873445374169E8
```

Fig. 5. A sample equation estimating the effect of independent variable ”Confirmation_of_receipt_ActivityDuration” on ”T10_Determine_necessity_to_stop_indication_ActivityDuration” (the class variable).