Process Discovery Contest @ BPM 2016

Background

Process Mining is a relatively young research discipline that sits between computational intelligence and data mining on the one hand, and process modeling and analysis on the other hand. The idea of process mining is to discover, monitor and improve real processes (i.e., not assumed processes) by extracting knowledge from event logs readily available in today's (information) systems.

These techniques provide new means to discover, monitor, and improve processes in a variety of application domains. There are two main drivers for the growing interest in process mining. On the one hand, more and more events are being recorded, thus, providing detailed information about the history of processes. On the other hand, there is a need to improve and support business processes in competitive and rapidly changing environments.

The lion's share of attention of Process Mining has been devoted to **Process Discovery**, namely extracting process models – mainly business process models – from an event log.

The IEEE CIS Task Force on Process Mining (http://www.win.tue.nl/ieeetfpm/doku.php) aims to promote the research in the field of process mining and its application in real settings. In collaboration with it, to foster the research in the area of process discovery, we are proud to introduce the 1st Process-Discovery contest, which will be collocated with the BPM Conference in Rio de Janeiro in September 2016.

Objectives and Context

The Process Discovery Contest is dedicated to the assessment of tools and techniques that discover business process models from event logs. The objective is to compare the efficiency of techniques to discover process models that provide a proper balance between "overfitting" and "underfitting". A process model is overfitting (the event log) if it is too restrictive, disallowing behavior which is part of the underlying process. This typically occurs when the model only allows for the behavior recorded in the event log. Conversely, it is underfitting (the reality) if it is not restrictive enough, allowing behavior which is not part of the underlying process. This typically occurs if it overgeneralizes the example behavior in the event log.

A number of event logs will be provided. These event logs are generated from business process models that show different behavioral characteristics. The process models will be kept secret: only "training" event logs showing a portion of the possible behavior will be disclosed. The winner is the contestant that provides the technique that can discover process models that are the closest to the original process models, in term of balancing between "overfitting" and "underfitting". To assess this balance we take a classification perspective where a "test" event log will be used. The test event log contains traces representing real process behavior and traces representing behavior not related to the process. Each trace

of the training and test logs will record complete executions of instances of the business processes. In other words, each trace records all events of one process instance from the starting state till the end state.

A model is as good in balancing “overfitting” and “underfitting” as it is able to correctly classify the traces in the “test” event log:

- Given a trace representing real process behavior, the model should classify it as allowed.
- Given a trace representing a behavior not related to the process, the model should classify it as disallowed.

With a classification view, *the winner is/are the contestant(s) who can classify correct the largest number of traces in all the test event logs. All event logs will have the same weight*.

Additionally, CPU time may be used to untie tools that perform identical in the aforementioned criteria.

There is also a limit to 4 GB for what concerns the amount of RAM memory that can be employed.

The contest is not restricted to any modelling notation and no preference is made. Any procedural (e.g., Petri Net or BPMN) or declarative (e.g., Declare) notation is equally welcome. The context is not restricted to open-source tools. Proprietary tools can also participate.

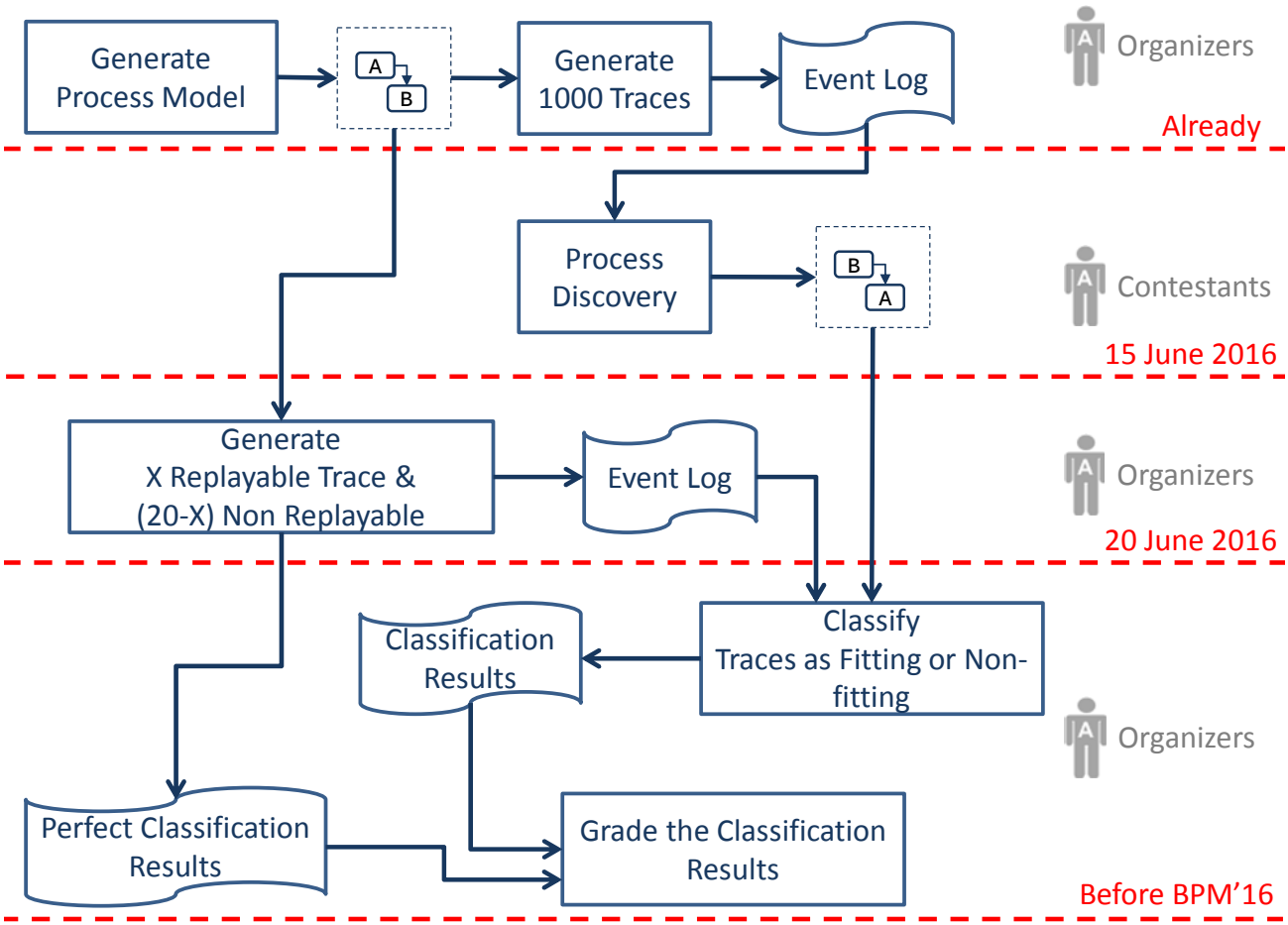
The winner will be announced on September, 18th or 19th, 2016 during the BPM 2016 conference, Rio de Janeiro, Brazil. The actual details of how the announcement is made will follow.

A contestant can be a single individual or a group that belongs to any institution, public or private. A prize will be awarded to the winning contestant. Details will follow.

Positioning of the Process Discovery Contest

The only other contest related to process mining is the annual Business Processing Intelligence Challenge (BPIC). The BPIC uses real-life data without objective evaluation criteria: It is about the perceived value of the analysis and is not limited to the discovery task (also conformance checking, performance analysis, etc.). The report is evaluated by a jury. The Process Discovery Contest is different. The focus is on process discovery. Synthetic data are used to have an objectified "proper" answer. Process discovery is turned into a classification task with a training set and a test set. A process model needs to decide whether traces are fitting or not.

Organization and Important dates



The figure above summarizes the organization of the contest and the important dates. The figure refers to the procedure for each of the 10 process models.

Now: the organizers have produced 10 process models with different behavioral characteristics (see also the appendix). For each model, an event log consisting of 1000 traces will be generated and made available to the contestants on http://www.win.tue.nl/ieeetfpm/doku.php?id=shared:process_discovery_contest. The process models will not be disclosed to the contestants. These logs will be noise-free, i.e. they do not contain behavior which is not part of the underlying process. The event logs are provided in both XES and CSV formats. For more information about the XES format, please refer to http://www.processmining.org/openxes/start.

15 June 2016: A process model needs to be submitted by the contestants for each of the 10 event logs.

20 June 2016: For each model, a test log containing 20 traces will be published on http://www.win.tue.nl/ieeetfpm/doku.php?id=shared:process_discovery_contest. A certain number of these traces can be replayed on the model; the remaining traces cannot. The test logs are the same for all

contestants. The BPMN representation of the 10 original processes, which produced the valid traces in the event log, will also be disclosed.

Before BPM 2016: For each of the 10 processes, the organizers will use the models submitted by the contestants to classify each trace of the respective event log. The winner will be the contestant that has provided models that can correctly classify the largest number of process traces with respect to all 10 processes. Correctly classifying traces as valid or invalid will receive equal weights. The same weight will also be given to all 10 processes. If two or more contestants can correctly classifying the same number of traces, their standing will be sorted by increasing CPU time (i.e. the contestant with lower CPU execution time will score first). There is also a limit to 4 GB for what concerns the amount of RAM memory that can be employed.

18-19 September 2016: The winner will be announced. The precise date will follow.

To provide support, **in any moment**, contestants can contact the organizers expressing the intention of submitting. To all contestants who expressed their intention, the organizers will send two test event logs for each of the 10 process models on **15 April 2016** and **15 May 2016**, respectively. Each of these event logs will be characterized by having 10 traces that can be replayed and 10 traces that cannot on the respective event log. However, no information will be given about which of the traces can or cannot be replayed. The contestants can submit their classification attempt to the organizers, which reply by stating how many traces have been correctly classified. The two feedback loops can be used as a mean to assess the effectiveness of the discovery algorithms.

Where, when and how to submit

Not later than 15 June 2016, each contestant needs to submit a document that at least contains the following sections:

- One section that discusses the replaying semantics of the process modelling notation that has been employed. In other words, the section needs to discuss how, given any process trace **t** and any process model **m** in that notation, it can be unambiguously determined whether or not trace **t** can be replayed on model **m**. *As an alternative to this section, the contestant can provide a link to a paper or any other document where the replaying semantics is described.*
- One section that contains the pictures of the 10 process models that have been discovered from the 10 event logs.
- One section that provides a link where one can download the tool(s) used to discover the process models as well as the step-by-step guide to generate one of the process models. In case the tool is not open-source, a license needs to be provided, which needs be valid at least until 30 September 2016. The license will only be used by the organizers to evaluate the submission.

No specific format is requested for this document.

Contestants submit the document by sending an email with subject “Process Discovery Contest - Submission” to discoverycontest@tue.nl. The same email should also be used for those who want to express their intention to submit.

Organizers

- Josep Carmona, Universitat Politècnica de Catalunya (UPC), Spain
- Massimiliano de Leoni, Eindhoven University of Technology, The Netherlands
- Benoît Depaire, Hasselt University, Belgium.
- Toon Jouck, Hasselt University, Belgium.

APPENDIX. Behavioral characteristics of Process Models from which the provided event logs were generated.

This appendix contains information that can be used by contestants in order to tune their discovery algorithms.

The 10 events logs are generated from 10 different process models. The event logs only record the information about the order with which activities are completed. Therefore, there is no life-cycle transition information and, also, no timestamp information. However, these information types are not relevant for the contest in question.

Each of these models is characterized by the following aspects:

- **Sequences.** Certain activities need to be sequentially executed. For example, when a given activity A occurs, it is eventually followed by a certain activity B in all runs of the process.
- **Exclusive Choices.** Certain process model branches at given decision points are mutually exclusive. For example, a decision point exists between activity A and B. In any run of the process, if activity A is executed, then activity B cannot, or vice versa.
- **Parallel Executions.** Certain branches are “parallel”, meaning that they can be completed in any order. For example, if a branch “A followed by B” is parallel to a branch “C followed by D”, activities A, B, C and D can be executed in any order with the only constraint that B cannot finish before A and D cannot finish C. For instance, the execution runs <…, A, C, B, D> or <A, C, D, B> are valid whereas <A, D, C, B> is not, with the latter being because D cannot conclude before C concludes.

Each of this model can optionally contain the following characteristics:

- **Loops.** Certain parts of the model can be repeated an arbitrary number of times.
- **Optional Activities.** Certain activities are optional and can be skipped in certain runs of the process.

- **Inclusive Choices.** Within the process, multiple sets of activities are optional, i.e. at least one set should be executed, but multiple sets of activities are also allowed. The difference with an exclusive choice resides on the fact that, in an exclusive choice, exactly one branch is activated; conversely, in an inclusive choice, more than one branch can be activate.
- **Recurrent activities:** Activities can be executed in multiple non-subsequent points during runs of the process.
- **Long-term dependencies**: A decision made at one point in the process can restrict the possibilities at subsequent decision points. For example, at the beginning of a process, a choice is made between an activity A and an activity B. When activity A is chosen, later during any run, an activity C cannot be executed; if activity B is chosen, activity C can still be executed. In the Petri-net terminology, this corresponds to Petri nets with non-free-choice constructs.

Exclusive choices can be characterized by **balanced** or **unbalanced paths**. If an exclusive-choice is characterized by being balanced, in any run of the process, each mutually exclusive set of activities has equal probability of being chosen. If conversely it is unbalanced, one set has a 90% probability of being chosen and the other sets, together, have 10%, with each of them having the same probability. In the remainder, we generated event logs such that either all decision points are balanced or they are all unbalanced.

With reference to the characteristics above, the processes to which the generated the event logs refer to are as follows:

Optional Characteristics (Always 2 for each process model)		**Exclusive-choice decision points**	**Process model / Event Log**
Optional activities	Loops	Unbalanced Paths	1
Optional activities	Inclusive Choices	Balanced Paths	2
Optional activities	Reoccurring activities	Unbalanced Paths	3
Optional activities	Long-term dependencies	Unbalanced Paths	4
Loops	Inclusive Choices	Balanced Paths	5
Loops	Reoccurring activities	Balanced Paths	6
Loops	Long-term dependencies	Balanced Paths	7
Inclusive Choices	Reoccurring activities	Balanced Paths	8
Inclusive Choices	Long-term dependencies	Unbalanced Paths	9
Reoccurring activities	Long-term dependencies	Unbalanced Paths	10