RS4PD: A Tool for Recommending Control-Flow Algorithms

Joel Ribeiro and Josep Carmona*

Universitat Politècnica de Catalunya, Barcelona, Spain.
{jribeiro,jcarmona}@cs.upc.edu

Abstract. The use of process discovery algorithms is in practice hindered by many factors, being the algorithm's representational bias, parameter configuration and algorithm's capabilities the most important ones. Nowadays, a user of these algorithms needs an expert knowledge in order to successfully apply them. In this demo, we present the RS4PD, a recommender system that uses portfolio-based algorithm selection strategies to face the following problems: to find the best discovery algorithm for the data at hand, and to allow bridging the gap between general users and process mining algorithms.

1 Introduction

Instead of learning the usage of each algorithm from a collection, one may simply rely on previous experiences on applying the algorithms in the collection in order to decide which is the best one for the data at hand. This is the idea underlying the system we present in this demo: RS4PD is a recommender system for process discovery that follows the same strategy as the portfolio-based algorithm selection [1]. Basically, this selection relies on a set (portfolio) of algorithms, which are executed over a repository of input objects (e.g., datasets or problems). Information about the executions (e.g., performance or results) is used to identify the best algorithms with regard to specific input objects. By characterizing these objects as sets of features, it is possible to build a prediction model that associates a ranking of algorithms with features. So, the prediction of the best-performing algorithms on a given input object can be achieved by first extracting the features of that object and then using the prediction model to compute the ranking of algorithms. This approach is used to build the RS4PD, with event logs as input objects and discovery techniques as algorithms. Further details can be found in [2].

The RS4PD is based on a server-client architecture (Figure 1). The main function of the server is to generate knowledge about the performance of techniques on different event logs. This knowledge consists of prediction models based on quantitative and qualitative information about the execution of discovery techniques as well as the discovered process models. The server includes both the evaluation framework and the repository, which support the training function of the

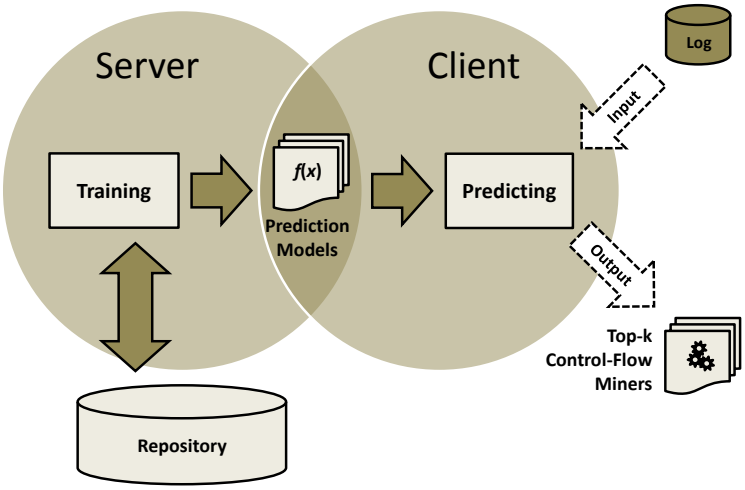

Fig. 1: Overview of the RS4PD.

recommender system. The training function as well as the evaluation framework are implemented as a package in the CoBeFra framework[1], while the repository is supported by a transactional database. The main function of the client is based on the knowledge generated in the server, and consists of predicting (recommending) the best-performing techniques for a given event log. This function is implemented as a ProM plugin.

A recommendation in the RS4PD is based on the prediction models produced by the training function. A prediction model can be defined as a function that maps a set of features of logs to a ranking of discovery techniques. In the RS4PD, different prediction models are built to predict the discovery techniques that are expected to perform better on logs characterized by specific features, according to different quantity and quality measurements (e.g., the runtime of the discovery technique or the precision of the resulting process model). Therefore, examples of recommendations from the RS4PD are, for a given event log, (i) the top-k fastest discovery techniques, (ii) the top-k most-precise discovery techniques (according to a precision conformance checking measure), or (iii) the top-k discovery techniques producing best-fitting models (according to a fitness conformance checking measure). It is important to mention that, unlike the previous examples that describe a single measurement, a recommendation can describe multiple measurements (e.g., the top-k discovery techniques combining i, ii, and iii).

Figure 2 presents an overview of the RS4PD client. The RS4PD client uses the precomputed prediction models to obtain the top-k best-performing discovery techniques for a given event log, which can be achieved as follows. First, the features of the given event log are extracted. Then, for each prediction model, the ranking of techniques regarding a measurement is predicted using the extracted features. Next, all the predicted rankings are combined into a final (aggregated) ranking. Finally, the top-k techniques are retrieved from the final ranking.

[1] http://processmining.be/cobefra/

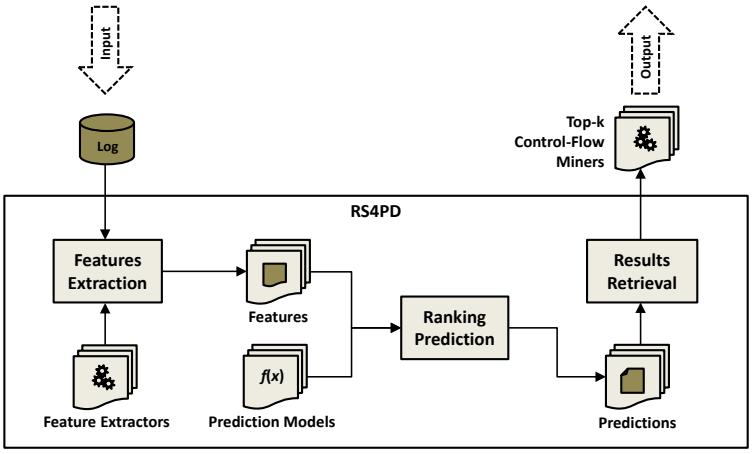

Fig. 2: Overview of the RS4PD client.

The current version of the RS4PD relies on a portfolio of 9 discovery techniques, which can be evaluated using 8 conformance checking algorithms. Table 1 presents the initial collection of techniques of the recommender system. The conformance checking algorithms are used to assess the quality of the results of the techniques. Table 2 presents the initial set of measures that can be assessed in the RS4PD.

Table 1: Portfolio of control-flow algorithms. These algorithms are available in the ProM 6 framework.

Technique	*Result*
Alpha Miner	Petri Net
Flexible Heuristics Miner	Causal Net
Flower Miner	Petri Net
Fuzzy Miner	Fuzzy Model
Heuristics Miner	Causal Net
Inductive Miner	Petri Net
ILP Miner	Petri Net
Passage Miner	Petri Net
TS Miner	Transition System

Table 2: Set of measures. The conformance checking algorithms that support these measures are available in CoBeFra.

Category	*Measure*
Performance	Runtime
	Used Memory
Simplicity	Elements in the Model
	Node Arc Degree
	Cut Vertices
Fitness	Token-Based Fitness
	Negative Event Recall
Precision	ETC Precision
	Negative Event Precision
Generalization	Neg. Event Generalization

An initial collection of 12 features are used in the RS4PD to characterize event logs: (1,2,3,4) the number of traces and events (total and distinct) in the log, (5) the average length of all traces in the log, (6) the average number of event repetitions intra trace, (7,8) the number of distinct start and end events in the log, (9,10) the amount of entropy and concurrency in the log, (11) the number of length-one loops in the log, and (12) the density of the log.

2 Maturity and Significance to the BPM field

Although we are in the process of incorporating new features (e.g., parameter optimization and exploration), the current distribution of the tool is stable. As the tool is developed under the ProM framework, it inherits the ProM interface and functionality.

Regarding the significance of `RS4PD` to the BPM field: to the best of our knowledge, the `RS4PD` is the first attempt to incorporate machine learning and information retrieval techniques for recommending process discovery algorithms. Also, the approach is very general and allows for the easy incorporation of new techniques, measurements and log features. Due to its continuous learning principle that makes the system to be decoupled in a server-client architecture, the initial promising results obtained in a set of experiments are expected to be even better when a larger training set will be available.

3 Download and Installation

The `RS4PD` (client component) is available as a ProM 6 plugin (Nightly Build version)[2] under the `Recommendation` package. This package has to be installed using the ProM's package manager, which not only installs (or updates) a `RS4PD` client but also downloads the most recent prediction models (cf. Figure 1).

4 Usage

The execution of a `RS4PD` client is performed on ProM 6, and it requires an event log as input to produce a recommendation of discovery techniques for that log. A recommendation consists of an aggregated ranking of discovery techniques that are expected to perform better on the given log, according to one or more measurements. The prediction of a ranking with regard to a measurement is based on some precomputed prediction model and the set of features of the given log. The combination of two or more rankings (into a final ranking) is achieved by a weighted aggregation of those rankings. The user can assign weights to measures in order to select the importance that measures should have in the final ranking.

Figure 3 depicts a screenshot of the user interface of a `RS4PD` client. The user interface is composed by three components: *features*, *measures*, and *results*.

Features (the top-left panel) summarizes the working event log in terms of features. This information is used on the prediction models for predicting the rankings of discovery techniques with regard to the different measures.

Measures (the bottom-left panel) allows the user to set a weight to each measure. Every time the user changes a weight value the button `Update` pops out. If this button is clicked, the final ranking of discovery techniques is computed on-the-fly taking into account the current set of weights; there must be at least one measure with a non-zero weight.

[2] `http://www.promtools.org/prom6/nightly/`

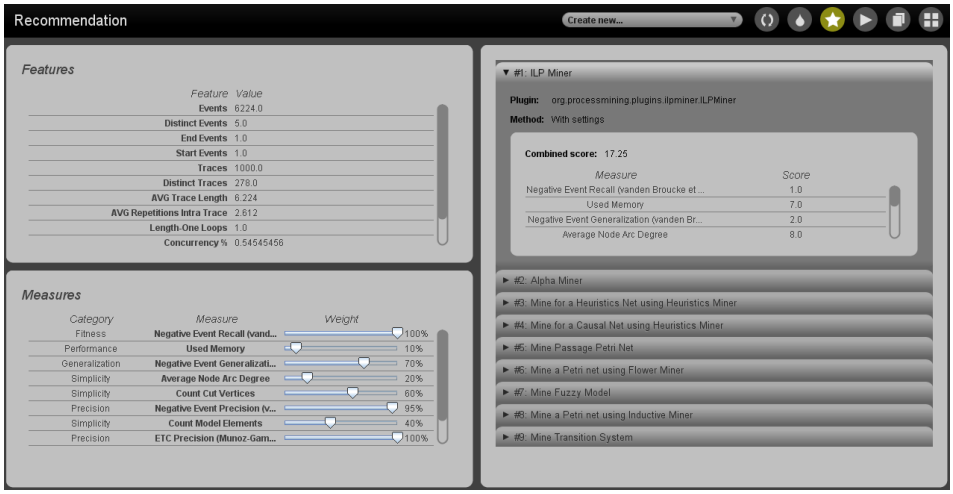

Fig. 3: A screenshot of the `RS4PD` client.

Results (the right panel) shows the ranking of discovery techniques that are expected to perform better on the given log. Details about the technique and its score values (the positions in the rankings for each measure as well as their weighted aggregation) are provided for each entry of the ranking.

Remark that, currently, the system simply considers 12 features, 10 measures, and 9 discovery techniques. The user cannot control any of these collections, but any suggestion for extension (or improvement) of these collections from the users will be taken into account for improving the system.[3] Therefore, the number of features, measures, and discovery techniques is expected to grow over time.

The screenshot of Figure 3 shows an example of a recommendation for some event log with 1000 process instances and 6224 events, in which different weights were assigned to different measures. The final ranking predicts the ILP Miner as the best discovery technique to be used on the given log (according to the selected measures). For this technique, 17.25 is the combined score, which consists of the sum of the weighted positions of the technique in the rankings regarding the measurements (i.e., $17.25 = 1.0 \times 100\% + 7.0 \times 10\% + 2.0 \times 70\% + 8.0 \times 20\% + ...$).

References

1. M. Mısır and M. Sebag. Algorithm Selection as a Collaborative Filtering Problem. Technical report, INRIA, 2013.
2. J. Ribeiro, J. Carmona, M. Misir, and M. Sebag. A Recommender System for Process Discovery. In *Proceedings of the 12th International Conference on Business Process Management*, BPM'14, Berlin, Heidelberg, 2014. Springer-Verlag.

[3] A functionality for submitting event logs (for improving the training of prediction models) as well as suggestions will be added to the tool in the near future.