Automating the Design of Process Mining Pipelines Through Meta-Learning (Extended Abstract)

Gabriel Marques Tavares
Università degli Studi di Milano, Italy
gabriel.tavares@unimi.it

With more than twenty years of history Process Mining (PM) techniques have now achieved the maturity level to cover the entire stack of the *data science pipeline*, from raw data to decisions [1]. To prepare process discovery or conformance checking, event logs can be extracted, lifted, cleaned, segmented, profiled, encoded. To support decisions PM models and metrics foster predictions and optimization procedures. Machine Learning (ML) algorithms are often integrated into PM pipelines to support, among others, noise or anomaly detection, clustering, feature selection, classification, and regression tasks. A consequence of this growth in the variety of tools available is that designing a PM pipeline is becoming complex. Identifying the best pipe of techniques to achieve the best results given a specific task and a specific event log is challenging even for experts. The spectrum of algorithms and concepts is larger and larger while the number of parameterizations among interacting algorithms is combinatorial. To deal with this trend, this thesis is aimed at studying PM pipelines to verify which steps can be automated.

1) Research Question: The general research question of this work is: *can the design of PM pipelines be automated?* Answering this question implies studying the relations between the algorithms composing a pipeline to verify if specific combinations are more effective than others. For example, event log characteristics may guide the choice of the appropriate discovery technique since high-level log descriptors can support the identification of the best discovery algorithm [2]. In particular, it will be of interest to verify which conditions, e.g. log complexity or noise level, and requirements, e.g. real-time response or quality measures, impact the optimal pipelining of PM techniques.

Clearly, our work will not be able to exhaustively address all the possible relations between all possible PM techniques or design requirements. However, we aim at introducing a novel methodology for studying these relationships, investigating some scenarios in a comprehensive way and using a reproducible method. A software framework for executing experimental analysis will also be delivered.

A practical side of our work is providing recommendations on the effective design of PM pipelines that can be exploited in training programs for specialists or in recommender systems.

2) Research Design: We investigate a solution based on Automated Machine Learning (AutoML), which allows non-experts to achieve satisfactory results and experts to optimize their tasks. For that, we propose a Meta-learning (MtL) strategy to approach the automation problem. MtL is the process of learning from the application of various learning algorithms on different data, thus, solving the algorithm selection problem by recommending the algorithm that produces the best performance for a particular data set. Given the results of multiple configurations observed during the training process, an MtL procedure recommends configurations for new tasks. We introduce a general framework where the observed configurations and tasks are abstract objects that can be instantiated according to the specific scenario one wants to study. Instantiating the framework means deciding descriptors, hyperparameter tuning, algorithms, and quality assessment metrics to be used in the MtL procedure. Figure 1 shows the abstract (non-instantiated) MtL-based framework applied to PM.

As observed in Figure 1, a set of event logs is required as input to the framework. The more heterogeneous the data set, the better because different process behaviors are explored. Failing in creating a representative group may negatively impact the framework's performance. The *Meta-Feature Extraction* step aims at obtaining event log features, known as meta-features according to MtL terminology. The challenge is to correctly capture log characteristics using a representative set of meta-features, capable of describing the process behavior from complementary perspectives. Furthermore, the feature extraction operation should have a low computational cost, otherwise, extracting meta-features would be more costly than testing all possible meta-targets. To this extent, based on information theory, statistical and PM feature extraction literature, we explore meta-features capturing activity, trace, and log descriptors. The meta-features cover several complementary perspectives, such as central tendency, statistical dispersion, probability distribution shape, log structuredness, and variability, among others. The *Meta-Target Definition* step is the most volatile in the abstract framework as its details depend on the task being studied. For instance, consider the problem of selecting a process discovery technique, a set of discovery techniques is applied to the event logs and, given a ranking function, the best discovery algorithm is selected. In this scenario, the ranking function could simply be a metric capturing the produced model quality. The ranking function can also be a result of aggregated metrics using average or even user-based weights. During the instantiation of the framework, one must consider the available techniques for a given problem and a respective ranking function able to rate the output of these techniques.

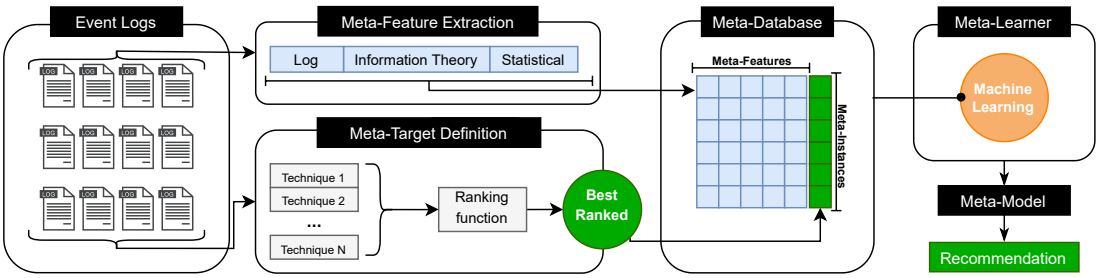

Fig. 1. Overview of the proposed framework.

This way, the meta-target definition step selects the appropriate technique for each event log. The results of the two preceding steps (meta-features and meta-targets) are joined in the *Meta-Database* phase, resulting in a data set similar to traditional ML applications, that is, a set of features describing instances and their associated labels. Hence, a *Meta-Learner* (e.g. a traditional supervised algorithm) can be fed using the meta-database. The meta-learner maps the relationships between meta-features and meta-targets and produces a *Meta-Model*. Given a new event log, the meta-feature extraction takes place to capture the process behavior and the meta-features are submitted to the meta-model, which can, in turn, recommend the most suitable technique for that event log.

3) Initial Results: The experiments to evaluate the framework are direct instantiations of the abstract model applied to different PM tasks. The first problem we studied is anomaly detection [3], which is traditionally performed by conformance checking techniques. In this scenario, we aim at enhancing anomaly detection performance by using encodings. Thus, the meta-targets are encoding techniques and the ranking function measuring detection accuracy is F-score. Encoding techniques are associated with meta-features extracted from the event logs, forming the meta-database. In this scenario, the framework averages 0.73 of F-score.

A second envisioned application is the task of process discovery [2]. Here the framework automates the selection of the optimal process discovery algorithm considering the meta-features extracted from the event logs. For that, a set of discovery techniques must be chosen, and the meta-target definition step ranks the techniques based on model quality metrics, such as fitness and precision. In this scenario, the framework averages 0.91 of F-score.

Recent experiments contemplated the clustering task, which can support variant analysis and serve as a preprocessing step for other PM tasks. The framework reached 0.69 F-score in recommending both encoding and clustering algorithms for a given event log [4]. Note that according to our research goal the accuracy of the framework is just an intermediate goal. The final aim is to verify which design options in PM can be automated because driven by regular and measurable factors, and which are intrinsically contextual or subjective.

4) Planned Activities and Challenges: If the ambition is obtaining solid conclusions about the relationship between the components of a PM pipeline, to offer reliable recommendations, important efforts must be spent to verify the solidity of the experimental design. Many aspects must be validated: (i) the representativeness of the event logs sample, (ii) the sensitivity of meta-features, (iii) the relevance of the algorithm selected (iv) the hyperparameter tuning. We plan to consolidate the experiments already executed by extending the set of data and algorithms used and providing a solid validation of the generalization power we can obtain in each scenario. The availability/unavailability of executable software implementing the last results of the literature will impose us some limitations without invalidating the generality of the approach.

An inherent challenge is that PM applications might be attached to certain purposes, ranging from abstract goals such as having an overview of the process to clearly defined questions such as identifying the delay root-cause for a sublog. The more specific questions may be assessed by quality metrics when well defined while more general goals are more subjective. We aim at instantiate the framework to a range of PM tasks in order to test its adaptability to narrowed scenarios.

5) Relation to the State of the Art: The flexibility of our framework allows the instantiation of a wide range of PM tasks. For each task we study, respective literature exists to compare with. Nevertheless, our framework can be compared (with restrictions) with recommender systems. For example, in [5], the authors use a portfolio-based algorithm selection strategy to recommend process discovery algorithms. However, the literature often lacks generalization as recommenders are designed for specific tasks, limiting their applicability to certain problems.

References

[1] W. M. P. van der Aalst, "Responsible data science: Using event data in a "people friendly" manner," in *Enterprise Information Systems*. Cham: Springer International Publishing, 2017, pp. 3–28.

[2] S. Barbon Jr., P. Ceravolo, E. Damiani, and G. M. Tavares, "Using meta-learning to recommend process discovery methods," 2021. [Online]. Available: https://arxiv.org/abs/2103.12874

[3] G. M. Tavares and S. Barbon Jr, "Process mining encoding via meta-learning for an enhanced anomaly detection," in *New Trends in Database and Information Systems*. Cham: Springer International Publishing, 2021, pp. 157–168.

[4] S. Barbon Jr., P. Ceravolo, E. Damiani, and G. M. Tavares, "Selecting optimal trace clustering pipelines with automl," 2021. [Online]. Available: https://arxiv.org/abs/2109.00635

[5] J. Ribeiro, J. Carmona, M. Mısır, and M. Sebag, "A recommender system for process discovery," in *Business Process Management*. Cham: Springer International Publishing, 2014, pp. 67–83.