Real-Time Business Process Recommendations

João Rodrigues
Instituto Superior Técnico, joaovalentimrodrigues95@gmail.com

Pedro Sousa
Instituto Superior Técnico, pedro.manuel.sousa@tecnico.ulisboa.pt

José Rodrigues
Link Consulting Sa., jose.rodrigues@linkconsulting.com

Recomendações em Processos de Negócio em Tempo-Real

Real-Time Business Process Recommendations

João Rodrigues, Instituto Superior Técnico, Portugal, joaovalentimrodrigues95@gmail.com

Pedro Sousa, Instituto Superior Técnico, Portugal, pedro.manuel.sousa@tecnico.ulisboa.pt

José Rodrigues, Link Consulting Sa., Portugal, jose.rodrigues@linkconsulting.com

Resumo

Process Mining é a disciplina focada na análise dos dados registados durante a execução de processos de negócio. A execução destes processos de negócio não é linear e pode conter pontos de decisão que afetam o decorrer do processo. *Decision Mining* é uma subdisciplina de *Process Mining* focada em descobrir e fornecer suporte nos pontos de decisão. A lógica de decisão nestes pontos pode não estar explicitamente definida ou otimizada. Grande parte dos trabalhos e soluções propostas nesta área estão focados em fornecer apoio à gestão *off-line,* com o objetivo de tornar explícita a lógica de decisão implícita. O projeto proposto neste documento apresenta uma solução que fornece recomendações, "*Next Best Action*", nos pontos de decisão, durante a execução do processo. Para isso, identifica automaticamente os pontos de decisão e os seus dados, aplica um algoritmo probabilístico de aprendizagem supervisionada e prevê a melhor ação.

Palavras-chave: Mineração de Processos; Mineração de Decisões; Aprendizagem Supervisionada; Suporte Operacional de Processos; Logs de Eventos.

Abstract

Process Mining is a discipline focused on the analysis of the data logged by the execution of deployed business processes. Business process' execution is not linear and might entail many decisions that affect the process execution. Decision Mining is a sub-field of Process mining, focused on finding and supporting these decision points. The decision criteria used in these decision points is often not explicit or optimized. Most research, techniques and algorithms in this area have been focused on providing off-line management support as means of explicitly representing implicit decisions. The solution proposed in this document presents a system that will provide the business actors a "Best Next Action" recommendation during the execution of business processes. To do so, it will be automatically identifying possible decision points, mine its data objects, apply probabilistic supervised learning algorithms and predict the best actions.

Keywords: *Process Mining; Decision Mining; Supervised Learning; Process Operational Support; Event Logs.*

1. INTRODUÇÃO

Business Processes are at the core of any organization, and organizations often deploy some sort of Business Process Management engine (BPME) to model and manage its processes. The execution of deployed processes leaves a trail of execution, created and saved by said BPMEs. These trails are commonly called **event logs** and enabled a whole new area of research called **Process Mining.**

Process Mining has proven to be beneficial to organizations, since it can discover processes, check its conformance to hand-modeled processes, enhance its specification and provide operational support. Business processes are not linear and often business actors will have to face decisions where many options are available. The decisions might be influenced by a variety of factors, some are dependent on the process instance data. However, it can also rely on the business actor's experience and knowledge about the process at hands. In a management point of view, it would be preferable that the decisions taken during the process would only rely on the process instance and not on the actor, since two different actors might decide differently under the same conditions and inevitably lead to different process outcomes and performance measures.

Decision Mining is a sub-field of **Process Mining**, which from the event logs aims at deriving decision points and decision logic explaining under which circumstances one course of action is preferable to another. To find the decision rules, **Decision Mining** techniques must identify decision points in the process, find their features (expressed by the process data) and apply some machine learning algorithm. Although there is already a wide range of research conducted in the field of Decision Mining applied to business process, this was mainly focused on discovering and representing decision logic, to provide process insight and to annotate the decision point with the tacit decision logic. However, these concepts can also be applied in an online recommendation setting, aimed at providing a ”Next Best Action” prediction to a business actor who is unsure as to which action to perform. This can help organizations turn their processes into more agile procedures, that learn the most fitting course of action from historical executions.

Process Mining covers a wide range of analysis tools and methods that can provide valuable insight about a deployed process. Decision Mining applied to business processes, has been mainly focused on discovering and representing decision logic. However, as pointed out in (Van Der Aalst, 2011) ”process mining should not be restricted to off-line analysis and can also be used for on-line operational support. Three operational support activities can be identified: detect, predict, and recommend”. In this document we propose a solution that lies on this field of operational support and will **predict** and **recommend** actions.

Given this context, we will now propose an approach that aims at **providing real-time recommendations**, **ranking the possible actions** in terms of their probabilities and that learns continuously from new observations. To do so, we will use novel decision mining techniques and a probabilistic classifier (Naive Bayes). These capabilities will be enabled by the analysis of historical and real-time event logs, bridging the gap between off-line and on-line analysis and support.

In Section 2, we will present the case study that will be worked on. Followed by related solutions in Section 3. In Section 4, our solution is presented with its results in Section 5. Finally, in Section 6, a conclusion is presented.

2. CASE STUDY

The solution will follow a methodology that can be applied in multiple environments. However, in the scope of this project, we are focusing on a specific case. We will work with the event logs of one of the major retail companies in Portugal, and these event logs will be relative to its device repair process. These event logs have two perspectives. The meta-data perspective (with the information about activity names and execution times) and the data perspective (which contains the data-objects of the process, also known as payload).

However, the solution can be configured in other environments, if the data has the canonical form defined in Table 1 and Table 2.

CaseID	Activity ID	Task Number	Outcome
2132013	Analyze Budget	213	APPROVE
2132013	Deliver Device	214	DELIVERED
1234100	Analyze Budget	401	REJECT

Table 1 – meta-data perspective

Task Number	Data Object ID	Data Object Value
213	Amount	120.00€
213	Own Brand	Yes
401	Amount	80€
401	Own Brand	No

Table 2 – data perspective

The meta-data perspective table is the one needed to mine the process' control-flow perspective and to discover the decision points. The data perspective is used to discover the process' data-flow and the activity's features. One important feature of these event logs is the **Outcome** column in the meta-data perspective, since this gives us the information about the decision of the process actor at the time he executed that activity.

3. RELATED WORK

Over the last years, the topics of Process Mining and Decision Mining have been gaining relevance and valuable research has been conducted in the area. This was mainly fueled by the growing popularity of Enterprise Resource Planning systems (ERP-systems) and other Process Aware Information Systems (PAIS). In this section, a literature review on the subject is introduced, mainly focusing on **Decision Mining,** in the scope of process mining, and **Recommendation and Prediction Systems**

3.1. Decision Mining

In (Rozinat, 2010) the author defines the concept of Decision Mining in the context of Process Mining as the application of data mining techniques for the detection of frequent patterns in business processes, providing valuable insight into the process and making tacit knowledge explicit. In this book, Anne Rozinat identifies the two major steps for deriving the decision logic of a process from its event logs.

First, the decision point must be identified. The author identifies a decision point as a place with multiple outgoing arcs. That is, an activity that, in different traces (process instances), has two or more distinct successors.

Second, the decision point needs to be turned into a classification task. In this classification task, the classes are the different decisions that can be made, and the attributes used for classification are the data values.

The classification algorithm used is the **Decision Trees Classification**, where for each decision point there is an associated decision tree. This classification method is also the one chosen in most of the implementations on this topic (Rozinat, 2010) (Ghattas, Soffer, & Peleg, 2013) (Aalst, 2015). In this book, a plug-in for the tool ProM is presented, the Decision Miner[1] plug-in.

The ProM[2] tool is an Open Source framework for process mining algorithms and is one of the most popular process mining solutions currently available. It is a central application that can be extended with plug-ins and benefits from a large supporting community. The many plug-ins that are currently available have made it a very complete tool, specifically in the fields of Process Discovery, Conformance and Enhancement.

Given the event logs in a canonical form, the tool discovers the underlying process model and the decision trees for each of the decision points identified. The author also identifies the main challenges inherent to the decision mining process. First, the usual challenges related to supervised learning, such as noise in the data, incomplete training sets and over-fitting. Second, the challenges related to Process Mining, such as invisible tasks, duplicate tasks and loops.

---

[1] http://www.processmining.org/online/decisionmining
[2] http://www.processmining.org/prom/start

In (de Leoni & Van der Aalst, 2013) the authors propose a novel approach for decision mining based on alignments. The authors present a way of aligning an event log with data and a process model with decision points. These alignments can be used to generate a well-defined classification problem per decision point. The authors use Petri Nets discovered with the control-flow information and enrich it with the data-flow information, creating a Petri Net with Data.

3.2. Recommendation and Prediction Systems

The idea of a Recommendation and Prediction System to support and improve business processes is not new. This area has a lot of potential, but it depends heavily on the process at hands and can achieve better or worse results based on the quality of the event logs.

The solution proposed in (Schonenberg, Weber et. al, 2008) provides the users a recommendation service based on the control-flow perspective. It uses the historical traces provided in the event-logs to predict what the next action should be. To do this, the system matches the user's partial trace, i.e. that belongs to the running process, and matches it with the historical traces. Then, each trace "votes" on what the next action should be. The more similar the trace is to the partial trace the more its "vote" counts. So, this is a recommendation system based solely on the sequence of actions. The solution proposed on this document has a similar approach in terms of the final objective, which is providing real-time recommendations. However, our solution takes into account the process payload. Therefore, the recommendations will be better suited for each recommendation request, since this process will be based on the process data and not only on the sequence of actions.

In (Ghattas, Soffer, & Peleg, 2013) a semi-automatic approach is proposed. This solution improves the business performance of processes by learning and deriving decision criteria formulated as decision rules from the experience gained through past process executions. This recommendation system uses decision mining, decision tree learning and path finding on the decision trees to determine which paths lead to the best outcomes. To allow for this notion of outcome ranking, the system uses the notion of process hard goals and process soft goals, which are, respectively, the process termination states and the process performance indicators. To determine these measures, a significant amount of process specific knowledge is necessary. The learning procedure only considers the process instances that reach desirable end states, ignoring instances that, for example, reached exceptions, even if the deviant behavior has a significant amount of instances. This detail differs from our approach, since we take into account all the process cases that we have at our disposal, therefore the system learns from all kinds of process cases and not only those that ended in desireable states.

In (Hamid Reza Motahari-Nezhad & Claudio Bartolini, 2011), the authors address the problem of recommending activity steps in collaborative IT support Systems by automatically discovering and annotating process models and with the introduction of a recommender. To do so, the authors developed a solution that analyses past case executions, discovers the step flow model, annotates it with case metadata and uses the metadata in open cases to match it with the annotated model and recommend the best next actions. In this solution, the authors opted for a more model-centered solution, whereas in our solution we are only focusing on the activities that were identified as decision points and the process payload in those activities, applying machine learning capabilities in said decision points.

4. SOLUTION

In this section we will go through the solution formulated through the analysis of the problem and of other solutions in the area. In the project specification, it was defined that the recommendation shouldn't consist of one and only one action, we defined that the recommendation shouldn't exclude the human factor, but rather simply assist it. Since in some cases the best action might consist of 2 or more actions, which might be equally likely, and therefore, the system must be capable of identifying those cases and threat them accordingly. Therefore, a probabilistic approach was chosen, so that, in addition to providing the best action, it provides the probability associated with each recommendation. This added layer of complexity differentiates this project and its solution from others in this area since most solutions on decision mining are mainly focused on explicitly representing the decision logic and recommendation systems solutions are mainly focused on displaying one and only one recommendation to the user.

4.1. Decision Point Identification

As discussed in the previous section and as pointed out in (Rozinat, 2010), the first step in decision mining is to identify the decision points in the process. To do so, one must analyze the control and data-flow of the process and identify possible places where more than one decision is possible.

To discover the decision points, our solution looked at the outcomes of each activity, since in our historic logs, this information is what defines the decision taken by the actor who performed said activity. Therefore, if in two different observations, the same activity has 2 or more unique outcomes, it was flagged as a decision point. For example, in Table 3, we can see that

CaseID	Activity Name	Outcome
537132415	Analyse Budget	Budget Approved
56807594	Analyse Budget	Budget Denied

Table 3 – Event Log Excerpt with Decision Point

In the event logs, there are 81 different activities. From these 81, 32 were flagged as decision points. These 32 were carefully analyzed and filtered. Some were ignored because the decision was not in the hands of the process actor but in the hands of the client. Others were ignored because there weren't enough observations to work with. From the previous 32, 24 were filtered out, leaving 8 activities as decision points. These decision points are different from one another, a few examples are: "Debit Notification Validation", "Responsibility Validation" and "Situation Analysis".

4.2. Turning Decision Points into Classification Tasks

After identifying the activities that constitute decision points, they need to be turned into classification tasks. To do so, for each occurrence of a decision point, its footprint needs to be mined. The more occurrences, the more footprints we will have and therefore the better for the classification algorithm. The activity's footprint is composed by the values of the process data objects at the time the activity was executed. Each data object is a feature for the classifier. With this data, one can create a table to summarize it, analyze and apply the classification algorithms.

Considering Table 1 and Table 2, we can see that the activity "Analyze Budget" has two different outcomes, therefore, it will be flagged as a decision point. Next, we need to extract its footprint. To do so, we match the task numbers from Table 1 with the task number from Table 2 and create a footprint table like the one in Table 4.

Amount	**Own Brand**	**Outcome**
120.00€	Yes	APPROVE
80.00€	No	REJECT

Table 4 – Footprint Table for activity Analyze Budget

With this, we are almost ready to apply our classifier, since we have the features and the classes. However, a great deal of feature selection had to be conducted. Each decision point had at least 100 features, which had to be analyzed one by one to reduce it and select only a few good features. Various types of features can be identified:

- Continuous numerical features, i.e. budgets;
- Discrete numerical features, i.e. numerical ID's, which are not good features for a classifier;
- Discrete textual features, i.e. a set of possible textual values; and
- Open text features, i.e. textual descriptions entered by the process actors. These features can be valuable if interesting information can be extracted. For example, if the feature is open but common traits can be observed.

The selection followed some criteria but was most conducted with intuition and applying some of the techniques described in (Guyon & Elisseeff, 2003). After selection, each decision point had between 2 and 7 features.

4.3. Classifier Algorithm

To choose the algorithm we had to consider the requirements. The classifier must be able to provide probabilistic insight about the possible actions, ranking them. Therefore, decision trees will not be the core algorithm used, unlike most solutions in this area, as reviewed in the previous section. The classifier must be fast when predicting the class, since in a real-time process the system has to output the decision in useful time window. And it must learn continuously from new observations. These requirements differentiate our solution from others in the area.

Considering these requirements, we developed a Hybrid Naïve Bayes classifier, capable of learning in bulk and with new singular instances. Since it is a probabilistic classifier it can rank the possible actions. It is also very fast when predicting the decisions (under 1 second). The classifier is hybrid because it can deal with both numeric and categorical features. One added benefit to our implementation is that it can deal with data that hasn't been seen in the training dataset, which is common since the features in a process like the one we're dealing with, have a wide range of possible values and we want the recommendation system to be able to deal with unseen data. We solved this by applying probability smoothing (Laplace or Add one smoothing), where unseen feature values have its counter set to 1, instead of 0, which means the probability for these unseen features values won't be 0.

4.4. Example

In this subsection we present an example of the behavior of the recommendation service. In this example we will focus on a specific decision point, called "Debit Notification Validation", which has two possible decisions, "Reject Debit" and "Accept Debit". To train the classifier we mined its footprint, in Table 5 we can see an excerpt of this footprint. As we can see, there are different types of features that are used. Some are self-explanatory, while others, like "roleENT" are business specific. However, one does not need to understand what it represents, as long as its values are useful for the classifier to learn from.

Contestation Reason	roleENT	Amount	Observations	Motif	Outcome
No schedule Violation	SCR	41.56	Debit authorized by Insurer	Exchange Authorized	Reject Debit
NA	FORN_6008	23.73	Client in Store	Exchange Authorized	Accept Debit

Table 5 – Footprint Table for activity Debit Notification Validation

Now consider a process actor that asks for a recommendation, with the process payload in Table 6. As we can see, the recommendation request has the same features, but the outcome, which is what the process actor wishes to know and act upon is missing. This process payload, belonging to an open process case, is sent to the classifier, which will decide upon.

Contestation Reason	roleENT	Amount	Observations	Motif	Outcome
No schedule Violation	SCR	61.21	Debit authorized by Insurer	Exchange Authorized	**???**

Table 6 – Process payload in activity Debit Notification Validation

If we test this with our trained classifier for this decision point the output will consist of the two possible outcomes and the respective probability, like in Table 7. This is the output presented to the process actor that requested the recommendation.

Outcome	Probability
Accept Debit	**23%**
Reject Debit	**77%**

Table 7 – Process payload in activity Debit Notification Validation

5. Evaluation and Results

To evaluate our solution, an automatic classifier evaluation was conducted. The dataset used was the one presented in the case study. Each decision point is a classifier, therefore, all 8 decision points were trained and evaluated with its respective dataset. The dataset was split into train and test sets. Three different classifiers were evaluated.

- A baseline classifier that for each recommendation request, simply outputs que most common decision observed in the dataset. This classifier works well if there is a significant class imbalance in the dataset;
- A decision tree classifier, which is the one chosen in most solutions in this area, as explained in Section 3. This classifier was also evaluated in order to compare the results; and
- The hybrid Naïve Bayes classifier presented in Section 4.

5.1. Automatic Classifier Evaluation

A classifier is an algorithm or mathematical function that maps input data to a category or class. In this solution, each decision point will be turned into a classification problem. There are several types of classifier algorithms (e.g. Decision Trees, Naive Bayes, Support Vector Machines, etc). The evaluated classifiers are Binary, when the decision point has only two possible decisions, and Multi Class when there are more than two possible decisions.

To evaluate the classifiers, the standard 4 metrics (Accuracy, Precision, Recall and Fscore) were calculated, as presented in (Solokova & Lapalme, 2009), with micro averaging for decision points that are Multi Class. In Figure 1 we can see these metrics for each classifier.

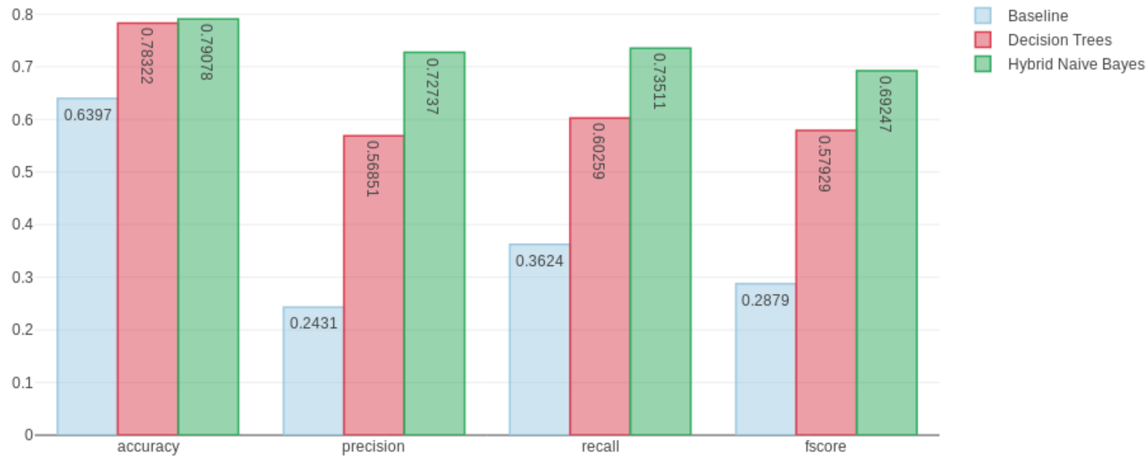

Figure 1 – Comparison between classifiers

The values in Figure 1 are the means of the metrics in the eight decision points that were chosen. As we can observe, for the same datasets, the probabilistic approach of the Hybrid Naïve Bayes classifier showed better results across the eight decision points.

These results are highly dependent on the dataset for each decision point. The common trait observed between the different decision points is that, the more features there are available, the better the results. Decision points with 5 to 7 features showed results around 0.98 for every metric. While others with 1 to 4 features showed significantly lower results, around 0.65 across all the metrics.

6. CONCLUSION

Process Mining and Decision Mining are valuable tools for business process managers. Tools that can help understand how deployed processes are really carried out. With our solution we bring the benefits of these tools to the frontline of the business processes, helping the process actors taking decisions where a decision might not be clear.

We presented the state of the art of these topics and our solution proposal, that will follow many of the concepts and methodologies proposed by other authors. Therefore, we hope to enrich this field of study with novel ideas.

Our solution uses an approach that has not been done yet. Most solutions in this area of Decision Mining and Recommendation Systems use Decision Trees as the main classifier algorithm or just provide recommendations that don't take full advantage of what is possible to learn from the event logs. Decision Tree classifying is a deterministic algorithm, and in our solution, we implemented a stochastic approach focused on the control-flow and data-flow perspectives of the process.

We think that this approach is of great value to the field, since it has never been done yet and is producing good results, tested on a real deployed process.

7. References

Aalst, W. V. (2015). Extracting event data from databases to unleash process. In J. v. Schmieadel, *BPM - Driving Innovation in a Digital World* (pp. 105-128). Springer International Publishing.

Bazhenova, E., & Weske, M. (2015). Deriving decision models from process models by enhanced decision mining. *International Conference on Business.* Springer International Publishing.

de Leoni, M., & Van der Aalst, W. (2013). Data-aware process mining: discovering decisions in processes using alignments. *Proceedings of the 28th Annual ACM Symposium on Applied Computing* , (pp. 1454-1461). Coimbra, Portugal.

Ghattas, J., Soffer, P., & Peleg, M. (2013). Improving business process decision making based on past experience. *Decision Support Systems*, 93-107.

Guyon, I., & Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, 1157-1182.

Rozinat, A. (2010). *Process Mining.* Eindhoven: Technische Universiteit Eindhoven.

Schonenberg, H., Weber, B., Van Dongen, B., & Van der Aalst, W. (2008). Supporting flexible processes through recommendations based on history. *International Conference on Business Process Management* (pp. 51-66). Berlin: Springer International Publishing.

Solokova, M., & Lapalme, G. (2009, July). A systematic analysis of performance measures. *Information Processing & Management, 45*(9), 427-437.

Van Der Aalst, W. e. (2011). Process mining manifesto. *International Conference.* Berlin, Heidelberg: Springer.

Wetzstein, B., Leitner, P., & Rosenberg, F. (2009). Monitoring and analyzing influential factors of business process performance. *Enterprise Distributed Object Computing Conference, 2009* (pp. 141-150). IEEE INTERNATIONAL.