Using Multi-Target Feature Evaluation to Discover Factors that Affect Business Process Behavior

Pavlos Delias[a,c,*], Athanasios Lagopoulos[b], Grigorios Tsoumakas[b], Daniela Grigori[c]

[a] *Eastern Macedonia and Thrace Institute of Technology, Kavala, Greece*
[b] *Aristotle University of Thessaloniki, Greece*
[c] *LAMSADE, Universit Paris-Dauphine, PSL Research University, France*

Abstract

Certain business environments, like health-care or customer service, host complex and highly variable business processes. In such situations, we expect fluctuating process behavior, which is difficult to attribute to specific causes, at least automatically. This work aims to provide process analysts with an additional tool to discover factors that affect the process flow. To this end, we propose a three-stage methodology to deal with the several challenges of this goal.

Adhering to the process mining paradigm that suggests for evidence-based process analysis and improvement, we introduce a horizontal partitioning approach to identify elements of process behavior during the first stage. Then, during the second stage, we discuss how log manipulations can yield characteristics that reflect various perspectives of the process. Finally, we propose a multi-target feature evaluation step to deliver insights about the associations between characteristics and process behavior.

The proposed methodology is designed to tackle challenges related to the general correlation problem of process mining, like dealing with general process behavior (not just local decisions) and relaxing the independence assumption among the elements of behavior. We demonstrate our approach step by step through a case study on a real-world, open dataset.

Keywords: Process Mining, General Correlation Problem, Multi-target Prediction

*pdelias@teiemt.gr

1. Introduction

Business process models, an essential tool for organizations to manage their processes [1], can be designed by experts or automatically discovered through event log files, i.e., records in an information system that provide detailed information about the activities that have been performed during a business process execution. Given the growing availability of event logs, an equally growing interest is drawn on automated process discovery. However, there are certain environments, like health-care or customer service, where processes are inherently complex [2]. Moreover, process variability may occur for a plethora of reasons. As indicative examples we can consider business rules that govern the process behavior (e.g., loyal customers can skip some steps); established habits (e.g., clients visit a particular office first, even if they should start from a different point); or even contingencies (a new employee did not know what task he or she should perform next).

In order to help in understanding such complex and highly variable processes, the goal of this paper is to propose a methodology that would consistently and effectively discover characteristics that affect process flow. This is part of the general problem of "relating any process or event characteristic to other characteristics associated with single events or the entire process" that in [3] is termed as the "general correlation problem" of process mining (not to be confused with the "case id correlation" problem [4], which refers to identifying a unique case id for each event). Assuming one achieves to correlate characteristics to process behavior, she can legitimately expect to deliver valuable insights [3]. This kind of insights can, for instance, be effectively used for off-line prediction (e.g., to predict tasks' load by examining a particular attribute of customers' profiles), or for on-line monitoring (e.g., to trigger an alert that a case will violate its Service Level Agreement (SLA) for duration, because it has performed a special ensemble of steps). The general correlation problem itself, can be viewed as a version of the issues related to the definition of *Context* in Business Process Management (BPM) since it involves what Rosemann et al. [5] call *context-aware* business processes, which can be defined as processes that can sense and react to changes in the context, leading to diversificated process executions. In addition, as Carvalho et al. [6] point out, the analysis of contextual information in business processes might indicate the need for their modification and exploit "learning from the past to support decision making". Overall, it is a matter of making evidence-based decisions for the process improvement and redesign endeavors. Of course, the "Context" thematic in BPM is a far broader area which can bring various contributions to process management (see for instance the summarizing Table 12 in [7]). This work focuses on the general correlation problem of process mining, which is still far from being a trivial issue. In the following, we enlist several reasons that make it a hard and challenging problem. We label them as "Challenge 1";"Challenge 2", etc. to facilitate the cross-references during the later sections.

First, the characteristics may refer to various process perspectives (Challenge 1) [8], like the control-flow perspective (e.g., what was the customer's last action?), the data-flow perspective (e.g., is this an emergency case?), and the organizational perspective (e.g., is a specific employee prone to taking shortcuts?). Second, characteristics may not be evident in the log file, thus they must be derived (Challenge 2)[9, 10, 3]. For example, when the analyst is interested in the number of loops performed during a case, or in the total duration spent on the five last activities, she can not find directly this information in the event log, which typically has the shape of a flat file, each row being the record of one event.

Other reasons concern how process behavior is defined. Hence, the third reason is actually a common pitfall, namely to consider too granular or too inclusive behavior (Challenge 3) [11, 12]. It's clear that a too granular view will generate irrelevant variability, as well as that a too inclusive behavior will lead to a fake homogenization. Moreover, a fourth challenge is posed by the fact that the emphasis is not limited to identifying the discriminating power of features, but there is also a great interest in connecting them with the process flows (Challenge 4). While the above reasons are related to the process behavior definition, two further challenges emerge from the scope of the behavior. The one is the typical process stakeholders' desire to interpret not just the local decision (e.g., the conditions of a decision point), but more general process behavior (Challenge 5). The other, a follow-up actually, poses a critical question (Challenge 6): Given the will to have insights on the *general process behavior*, what constructs or variables can reflect it, and what operations would be necessary to measure them?

Furthermore, the elements of behavior that we are trying to explain are not necessarily mutually exclusive, as well as they are rarely independent to each other (Challenge 7). As parts of the same process, these elements can interact in various ways, so trying to explain any of them in isolation involves a risk of missing certain aspects of reality, resulting in fragmented process knowledge [13]. Finally, a last challenge (Challenge 8), is that any methodology with an ambition to propose a generic solution, should be based mainly on the observation of the event log, and should not rely on the process analyst's skills and instincts to anticipate which variables are the most influentials and which ones should be involved in hypotheses formulations.

In this work, we propose a methodology to respond to all the above challenges. To this end, we developed an approach that consists of three stages. During the first stage, we present how a horizontal partitioning of the event log can tackle the challenges related to the general behavior, i.e., defining "Goldilocks" behavior which is neither too granular nor too inclusive; interpreting general process behavior and not just the local decisions; proposing constructs or variables that reflect the notion of process behavior, as well as the operations that are necessary to measure them. During the second stage, we discuss how we can acquire case characteristics from the event log, and how it is possible to address various perspectives. Finally, during the third stage, we demonstrate how to connect the characteristics to the process behavior by using algorithms that do not assume independence among the elements of behavior and can handle heterogeneous characteristics.

The rest of this article is organized as follows. In Section 2 we briefly review relevant works, and contrast them with the novelties of our approach, while the proposed methodology is presented in detail in Section 3. Next, in Section 4, we apply the methodology to a real world process log and we examine the results. Finally, a short discussion concludes the paper in Section 5.

2. Related work

A first attempt to address the general correlation problem in the context of process mining was Decision Mining [14], where authors use *decision trees* to analyze how data attributes influence the choices on decision points (XOR gateways). Decision trees are popular in process mining to discover causes for a particular dependent variable (e.g., process delay) [15], one of the pioneer work being [16]. Mining of decision rules is also addressed in [17, 18, 19]. There are two main differences of our work with that family of methods. First, as these methods seek to discover conditions for the branching points, they focus on local process behavior. They were not developed to support situations when the interest is on more general behavior, like a long sequence of steps. Second, it is clear that these methods, in order to discover branching conditions, require the process model as input. Therefore, these methods inherit the relevant process discovery bias, and the model's representation bias. Moreover, this requirement enforces the process analyst to discover a model early in her analysis, a fact that is not always desirable. An interesting solution to this problem is given in [20], although the authors' motivation in that work is in process discovery and not in the correlation problem. They propose to consider data during the discovery method, so the delivered model is data-aware. This way they achieved to eradicate the a-priori process model requirement, however, their approach still focuses on local process behavior and it exploits only the data perspective characteristics. A different approach, which also does not require a process model as input, is to take a declarative approach to model business processes. Declarative techniques [21, 22, 23, 24] introduce constraints in models as rules that have to be followed, i.e., they summarize complex behavior in a compact set of behavioral constraints on activities [25]. However, existing techniques (e.g., [19, 26, 27]) target the discovery of constraints based on a set of Declare templates (e.g., the "response(A,B)" template that requires that whenever activity A happens, activity B should happen after A), therefore they are limited to the control-flow perspective. In [28] authors try to address this limitation by discovering correlations, which are defined over event attributes and linked through relationship operators between them. In particular, they look into the generated set of constraints for three special event-based characteristics, namely property-based, reference-based, or moving time-window correlations between every two events.

To be able to correlate any characteristic, belonging to virtually any perspective, with any other characteristic, a general framework is proposed in [3]. In particular, the authors propose the use of decision or regression trees to test a number of characteristics against a dependent variable (a characteristic acting as a class attribute). The dependent variable as well as the set of the independent characteristics have to be explicitly defined by the analyst. In addition, the correlations tests must be run on a one-by-one basis, meaning that, it is not practical to check the interactions' effects.

The general correlation problem is tightly related to business process deviance mining, where the aim is to discover and explain deviances in business process executions. Deviance mining problems are usually treated as supervised problems, where there is a target variable that defines the deviancy (e.g., delays in performance), a classifier that assigns cases to classes, and outputs of classifiers in terms of patterns or rules that cater insights to business process analysts [29]. Nguyen et al. [30] provide a taxonomy of the techniques proposed for deviance mining, distinguishing between approaches that use individual activities, frequent sets of activities, or sequences of events as features.

An emerging need, concerning the classifiers that shall be used throughout the general correlation problem, is the simultaneous handling of multiple elements of behavior. Modeling multiple elements of behavior at the same time, falls into what is called *multi-target prediction* in the machine learning literature. Multi-target prediction is concerned with the simultaneous prediction of multiple target variables of diverse type, such as binary [31], nominal, ordinal, real-valued [32] or even mixed. Often, these multiple target variables are related either explicitly, for example, they could represent a ranking, be nodes of a graph, or have a spatial, temporal or spatio-temporal relationship, or implicitly, for example, via hidden mutual exclusion, or parent-child relationships. The main challenge in the area of multi-target prediction is the exploitation of such relationships for improved prediction accuracy.

The novelties of this work are that we do not require any process model as input and that we follow a conceptually unsupervised approach, since we do not require from the process analyst to define *any dependent variable*. In addition, our method can handle heterogeneous characteristics and involve them in patterns that can deliver insights, even when the behaviors that we want to explain are dependent to each other. In the following, we present how this challenging task can be performed.

3. Methodology

We propose a methodology that unfolds in three stages. The aim of the first stage is to address the challenges (mentioned in the Introduction) related to the general behavior:

- Challenge 3: To propose a compelling way to recognize elements of behavior that balance between being too granular or being too inclusive.
- Challenge 5: To suggest a technique that will allow interpretation of the flows with a broader scope than local decision points, i.e., richer insights than the conditions of a decision point.
- Challenge 6: What constructs or variables can reflect the *general* process behavior, and what operations are necessary to measure them, namely, how to introduce an effective operationalization of the general process behavior.

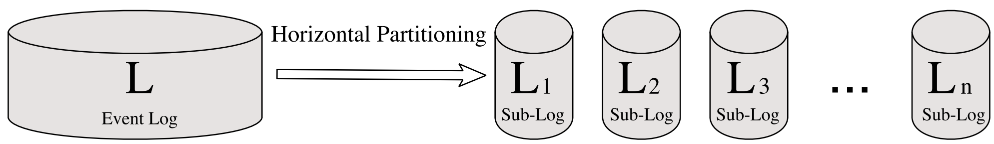


Figure 1: The general idea of horizontal partitioning. Adapted from [33] and [34].

We advocate that the points above can be tackled by horizontally partitioning the event log. A horizontal partitioning splits the event log into several sub-logs, while each sub-log contains all events that correspond to a particular subset of activities. This way, each case appears potentially in all sub-logs. The intuition of horizontal partitioning, illustrated in Figure 1, is to discover a process fragment per sub-log, which corresponds to the behavior that is defined by the activities included in the sub-log. Hence, the challenge is to group together coherent sets of activities since these sets should be able to i) reflect general behavior, as well as to ii) not provide an overly fragmented view.

The second stage is designed to address the challenges of dealing with characteristics that could be about various perspectives, i.e., control-flow, data, etc. (Challenge 1), and that should be derived through the log (Challenge 2). To this end, we present a guided procedure to build a case log.

The third stage conveys a feature evaluation approach that responds to the following challenges:

- Challenge 4: The outputs should not be limited to identifying the discriminating power of characteristics, but they should also suggest their effects on the process behavior.
- Challenge 7: The characteristics as well as the elements of behavior are not independent, so it is not enough trying to explain any of them in isolation.
- Challenge 8: The analysts is not required to state any a-priori hypotheses for the effects of characteristics, namely there is no need to define a dependent variable.

In the following subsections we present analytically the steps of the proposed methodology, which are concisely illustrated in algorithm 1.

Let us first define the basic notions relevant to our methodology. Activities in every process are going through states of their life-cycle. Every life-cycle transition performed in the context of a business process generates an *event* e. Transactional models for activities (e.g., as the one described in the XES standard [35]) can include various states such as "assign", "suspend", "resume", with the "start" and the "complete" states being the most common. Events have attributes $r_n \in R, n \geq 3$, since there are three mandatory attributes: the time-stamp, the case identifier, i.e., an attribute that uniquely correlates the event to a process instance (or *case*), and the activity label (when no attribute for the transition type exists, we can assume the "complete"). We shall use the operator $\#_{r_i}(e)$ to get the value of attribute r_i for event e. An *event log* $\mathcal{L}$ is a collection of events, which we assume to belong to a single process. Table 1 illustrates a sample event log of a healthcare process which contains the three mandatory attributes (the case identifier - the patient code; the activity that generates the event; the time that the activity's completion actually happened) and one additional attribute that states the Clinic where that particular activity took place. Such kind of data, i.e., timestamped events, likely characterized by additional attributes readily exist in process-aware information systems [36, p.3-8] like workflow management systems, ERP systems, enterprise application integration platforms. In addition, as noted in [37, p.3-10] there is now-days an abundance of event logs due to the logging potentials of e.g., IoT systems and customer journeys.

Case ID	**Activity**	**Time-stamp**	**Clinic**
1226	Administrative Rate - First Pole	11/2/16	Radiotherapy
1226	Follow-up counseling outpatient	5/16/17	Obstetrics & Gynaecology clinic
1227	Follow-up counseling outpatient	5/18/15	Obstetrics & Gynaecology clinic
1228	Follow-up counseling outpatient	5/18/15	Obstetrics & Gynaecology clinic
1228	Thorax	9/13/05	Radiology
1228	Immunopathological assessment	9/15/05	Pathology

Table 1: A sample event log with four attributes: The case identifier, the activity label, the time-stamp, and the clinic where the activity is performed.

Let $\mathcal{A}$ be the set of all the possible activities that can occur during the process. Then a horizontal partitioning [33] is an assignment of each $a_i \in \mathcal{A}$ to one or more of k subsets $\mathcal{A}_P \subset \mathcal{A}$.

A *case* $c \in \mathcal{C}$, uniquely identified by an identifier, is a process instance and may comprise several events. It is characterized by characteristics $h_m \in \mathcal{H}, m \geq 1$, while $\#_{h_m}(c)$ returns the value of characteristic h_m for case c. Since cases are uniquely identified, it is clear that $\#_{caseID}(c) \neq \#_{caseID}(c'), \forall c, c' \in \mathcal{C}$. If we order chronologically the events of every case, we get a sequence of events, which we call a trace τ. Finally, a *case log* $\mathcal{L}_{\mathcal{C}}$ can be treated as a relation whose relation scheme is specified by the set of case characteristics [38], and it is a matrix $|\mathcal{C}| \times |\mathcal{H}|$, like the one illustrated in Table 2.

Case ID	**Age**	**Number of visits**	**Received Treatment**
1226	65	11	No
1227	82	5	Yes
1228	67	5	Yes
1229	74	9	No

Table 2: A sample case log with three characteristics: The patient's age, the number of visits, a flag that indicates if she has received the treatment

3.1. Horizontal partitioning of the event log

The aim of horizontally partitioning the event log $\mathcal{L}$ is to end up with clusters of activities that correspond to clean-cut, recognizable fragments of process behaviors. Of course, the fundamental underlying assumption here is that process behavior is explained by activities' occurrences. One could argue that process behavior, in order to be explained, needs a process model and not just a set of activities, but this argument does not refute the plausibility of our assumption, since, given a set of activities (and the corresponding horizontal partition of the event log), it is trivial to discover a process model. Indeed, our method is agnostic to the process discovery technique that may be used for this purpose. Therefore, we operationalize process behavior as activities occurrences, and in particular, we will consider as an element of process behavior a finite set of activities.

On the grounds of the above operationalization, to deliver an effective horizontal partition of the event log, aiming at identifying distinct behaviors, we shall consider the following quality requirements. An effective partitioning should allow frequent patterns to be represented within single clusters, namely, it should deliver coherent clusters wherein activities are strongly connected to each other. The favorite situation is to have clusters with clean-cut borders, i.e., the connections among activities of different clusters should be as weak as possible. Well separated clusters (strong connections of activities within the same cluster and weak inter-cluster connections) would not let process behaviors to be spread on more than one cluster, as well as they would favor different behaviors per cluster. Moreover, as we have already mentioned in section 1, we do not want to consider too granular or too inclusive behaviors, therefore, there is an additional requirement to balance the size of the clusters.

As a way to derive coherent groups of activities with respect to process behavior discovery, we need a "connectivity" metric which will expose the network structure among the activities of the process. A connectivity metric should return high values for two activities when there is a frequent path connecting these activities in traces of an event log and low values when there is no such path (or it is faint). Therefore, in order to discover this kind of paths, we confine ourselves to direct dependencies of two activities.

In particular, let

$$w_{ij} = \frac{\text{number of traces where } i \text{ and } j \text{ are directly connected}}{\text{total number of traces}} \tag{1}$$

be the connectivity metric between activities i and j. Notice that at this stage we do not care about which activity is successor or predecessor, since what is important is to group together activities that are strongly connected.

Let us denote as $\mathbf{W} = (w_{ij})$ a form of an "adjacency" matrix for all activities $a_i \in \mathcal{A}$ that are registered in the event log. The matrix elements w_{ij} declare the dependencies (connectivity) among activities, hence a form of adjacency. Since while measuring the connectivity metric we did not consider the ordering of the activities, $\mathbf{W}$ is symmetric, and it has a complete set of real eigenvalues. Let us also denote an $|\mathcal{A}| \times 1$ indicator vector $\mathbf{v}_k = [\cdots v_k^i \cdots]^T$ whose elements v_k^i are given by

$$v_k^i = \begin{cases} 1, & \text{if activity } i \text{ is assigned to cluster } k \\ 0, & \text{Otherwise} \end{cases} \tag{2}$$

The indicator vector $\mathbf{v}_k$ denotes which activities comprise the k^{th} cluster. Each cluster is described by a distinct indicator vector, resulting in totally K different vectors.

Let us also denote $\mathbf{D} = diag(\cdots d_i \cdots)$ as the diagonal matrix, whose elements d_i, $i = 1, 2, ..|\mathcal{A}|$ express the cumulative connectivity degree of the activity i with all the other activities. That is

$$d_i = \sum_j w_{ij} \tag{3}$$

As it is proved in [34], the optimal vectors $\hat{\mathbf{v_k}}$, namely, a horizontal partitioning of the event log that delivers well separated clusters which are capable to expose general process behavior, can be calculated by minimizing the connections among activities of different clusters, like the following:

$$\hat{\mathbf{v_k}}, \forall k : \min \sum_{k=1}^{K} \frac{\mathbf{v_k^T}(\mathbf{D} - \mathbf{W})\mathbf{v_k}}{\mathbf{v_k^T}\mathbf{D}\mathbf{v_k}} \tag{4}$$

The indicator vectors are binary vectors (the i^{th} row of a vector $\hat{\mathbf{v_k}}$ is 1 if i^{th} activity is assigned to the k^{th} cluster and 0 otherwise). Unfortunately, the optimization of (4) subject to the binary representation of the indicator vectors is an NP hard problem. However, if we relax the indicator vectors to take values in continuous domain, then we can solve the problem in polynomial time through the equation

$$\hat{\mathbf{V_K}} = \mathbf{D}^{-1/2}\mathbf{V} \tag{5}$$

where $\mathbf{V}$ is a $|\mathcal{A}| \times K$ matrix the columns of which are the *eigenvectors* of the K largest eigenvalues of matrix $\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$, and $\hat{\mathbf{V_K}}$ is the relaxed version of the indicator matrix $\mathbf{V} = [\mathbf{v}_1 \cdots \mathbf{v}_K]$, the columns of which refer to the K activities' subsets, while the rows to the activities $a_i \in \mathcal{A}$. Still however, we need to round the continuous values of the relaxed matrix into a binary format. More specifically, each row of $\hat{\mathbf{V_K}}$ must contain one element equal to 1 and the rest equal to zero. To this end, in [39], the k-means algorithm is proposed, however, since our overall goal is to identify clusters of activities that expose meaningful process behaviors, as it is suggested in [34], their grouping should not only allow coherent clusters, but it should deliver groups of balanced sizes as well. Nevertheless, due to the noise or to the infrequent behavior in the event log, the k-means algorithm will likely return one or two big groups, while the remaining groups will be small. Therefore, we need a more robust clustering technique, like the method proposed in [40], which handles different cluster scatter constraints. So, we consider the rows of $\hat{\mathbf{V_K}}$

as the population to be clustered in K classes. Essentially, the output of this stage is a variable that indicates the cluster membership of every activity class in a way that activities of the same cluster are as much as possible connected to each other, revealing coherent sets of activities that co-occur during process execution, in a way that they exhibit elements of the process behavior. We should note that since the clusters' size is balanced, and since we optimize for well separated clusters, we expect to observe behaviors that are neither too granular, nor too inclusive.

3.2. Transforming an Event Log into a Case Log

Information in an event log, as specified in the XES standard [35], may refer either to the level of the log itself, or to the case level, or to the atomic event level. In principle, attributes of any level could affect the process behavior. Therefore, the choice of granularity of the attributes is a key parameter for the general correlation problem. For instance, a popular approach is to focus on the atomic event level (e.g., [14]), while in [7] authors argue that since a process comprise several activities, it is difficult to determine an adequate focus of attention, so they introduce a broader focus of reference that they call *process essence*. The approach we present in this work supports the analysis at the *case* level. The intuition behind our choice follows actually a common marketing practice: to segment a heterogeneous population based on profile characteristics. In particular, we want to guide the correlation problem by cases' profiles as expressed by cases' characteristics and to propose a relevant *operationalization* of the general process behavior.

Therefore, the event log should be aggregated by case, and get transformed into a matrix, whose every row will be a distinct case, and every column a case-wise feature. These features may refer to *every perspective* (e.g., control-flow, data, time) and must be *derived* through the event log. It is important to notice that characteristics can be measured by any scale (nominal, ordinal, numeric, etc.). In [3], authors provide several log manipulations that can return such kind of features (first event in a case; average value of a variable for all events in one case; duration; etc.), yet the number of potential manipulations can be limited only by the creativity of scholars. We shall also note that even when the event log contains just the mandatory fields (case id, activity, time-stamp), it is still possible to derive several characteristics for cases, e.g., the number of activities performed, its duration, if it is performed on weekdays or on weekends, the last event (exit point) of the flow, etc.

We additionally propose to attach the clusters discovered during the previous stage as collateral (control-flow) features. More specifically, following the graph partitioning approach of section 3.1, we expect to get granular elements of process behavior as clusters of (strongly connected) activities. Let us call every set of clustered activities a *region*. If a case's trace comprises one region's activities, it would signify that this particular case exhibits that particular element of process behavior. Specifically, we assume that the occurrence of a particular set of activities within a certain case exposes a particular behavior for that case. For example, when we observe that a case comprises events relevant to anesthesia, we can assume that this case exposes surgical procedures. To operationalize this assumption, we propose three options:

- A binary scale: If a case has visited any of the region's activities, we put a 1 in the corresponding cell of the matrix, otherwise we put a zero.
- Percentage of cluster: In every "region" column we put the percentage of the region's activities that were visited by the corresponding case.
- Percentage of trace: In every "region" column we put the ratio of the number of the case's trace elements that belong to that region, over the total number of trace elements.

The output of this stage is a matrix that has at every row a distinct case and at every column a case characteristic. The "region" columns are included in this matrix to enable the next step of the method. Every cell contains the evaluation of its row case to its column characteristic (or region).

3.3. Discovering the influence of case characteristics

The rationale of this stage is to connect the process behavior (as expressed by a case performing activities that belong to regions) to the case characteristics. To this end, an approach based on multi-target feature evaluation is employed. In particular, we consider as features of a predictive model the case characteristics and as targets of the model the regions, which portray the process behavior. In machine learning, feature selection is commonly used to produce simpler, more interpretable and more precise predictive models while avoiding the curse of dimensionality and overfitting. The selection is usually performed by evaluating different subsets of the features and by estimating the quality or score of each attribute. In our case, feature evaluation can also be used to correlate the case characteristics to process behavior.

Thus, in the third stage of our process we treat the problem of discovering the influence of characteristics to process behavior as the feature evaluation problem of the machine learning field. Because we do not embrace the assumption of the independence of characteristics, we calculate the score and the rank of each case characteristic using the Relief family of algorithms [41]. These algorithms were chosen since besides being aware of the dependence between characteristics, they are also efficient, as well as they can offer a comprehensible interpretation of the results [41]. Specifically, we use the ReliefF method when the binary scale option for the regions is selected, and the RReliefF method when the regions are represented as percentages. We shall note that although typically, the quality estimates of attributes (characteristics) are interpreted as equation 6 suggests, i.e., the difference of two probabilities, when the problem space is dense, as [41] proved, the quality estimate of the characteristic can be interpreted as *"the ability of the attribute to explain the changes in the predicted value"*.

$$\begin{aligned} W[h] =& P\left(\text{different value of } h \mid \text{nearest case with different prediction}\right) \\ &- P\left(\text{different value of } h \mid \text{nearest case with same prediction}\right) \end{aligned} \tag{6}$$

The final scoring or ranking list is produced by evaluating the characteristics against each region-target separately and then averaging the score and rank of each feature across the different targets. The higher the average score or rank of a feature, the stronger the connection between the case characteristic and the region.

4. Application

4.1. Case description

In order to assess the proposed methodology, we applied it to a real life event log of a Dutch academic hospital [42], originally intended for use in the first Business Process Intelligence Contest (BPIC 2011). The original log contains data for 1143 cases who are patients of the Gynecologist department, but they may visit different departments of the hospital to perform any set of the more than six hundred available activities. For each event, among others, the log records the patient ID, a description of the activity that generated the event, its timestamp, a flag indicating whether it that was an urgent activity, the age of the patient at that time, the department where

Algorithm 1: Method to discover characteristics that affect process flow

Input : An event log $\mathcal{L}$, a set of relevant characteristics $\mathcal{H}$

Stage *1:* <u>*Horizontal partitioning*</u>

    Find unique activities of the process $a_i \in \mathcal{A}$;
    Calculate connectivity metric $w_{ij}, \forall i, j \in \mathcal{A}$;
    Create a non-directed graph with activities as nodes and metrics w_{ij} as the weighted edges;
    Partition the graph into K clusters by optimizing intra-cluster and inter-cluster connectivities, and by balancing clusters sizes ;
    return *cluster membership for every* $a_i \in \mathcal{A}$;

Stage *2:* <u>*Building a case log*</u>

    Create a matrix $\mathcal{L}_{\mathcal{C}}$ with $|\mathcal{C}|$ rows;
    foreach $h \in \mathcal{H}$ **do**
        Derive h through log $\mathcal{L}$ manipulations;
        Add a column in $\mathcal{L}_{\mathcal{C}}$ for h;
    end
    Add one columns in $\mathcal{L}_{\mathcal{C}}$ for each clusters of stage 1 (K columns) ;
    return *a cases' profile matrix* $\mathcal{P} = ||\mathcal{C}| \times (|\mathcal{H}| + K)|$;

Stage *3:* <u>*Case characteristic evaluation*</u>

    Set as $\mathcal{X}$ the $|H|$ first columns of $\mathcal{P}$;
    Set as $\mathcal{Y}$ the $|K|$ last columns of $\mathcal{P}$;
    Create zero matrices S_r and S_s with $|\mathcal{X}|$ rows;
    foreach $y \in \mathcal{Y}$ **do**
        $S_s = S_s + \text{ReliefScore}(\mathcal{X}, y)$;
        $S_r = S_r + \text{ReliefRank}(\mathcal{X}, y)$;
    end
    return $S_s/|\mathcal{Y}|,\ S_r/|\mathcal{Y}|$

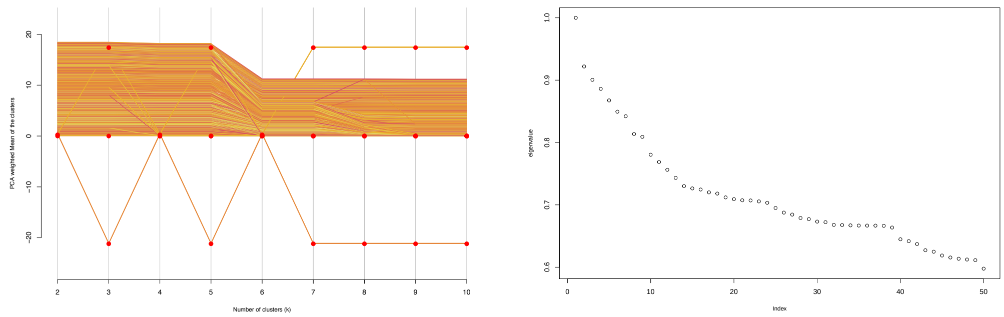
the activity was performed, and several diagnosis and treatment codes. We select this log because a) it is freely available and ii) because it contains many characteristics for each event. We pre-processed the dataset as we describe below.

To correlate events with cases, we found that the patient ID was not a convenient variable, because a patient may visit the hospital many times, yet in disjoint sessions (e.g., a series of visits during January and another series, several months later, at a different clinic). Therefore, we assumed that if the same patient does not visit the hospital for one week, for her future visits, she is considered as a different case (this concept is also followed by [43]). Then, we eliminated cases that contained just one event, to end up with a case log $\mathcal{L}_{\mathcal{C}}$ of 4640 cases that account for 147,888 events. Recall from section 3 that if we order chronologically all the events that belong to a single case we get the case's trace τ, i.e., $\#_{\tau}(c) = \langle e_1, \ldots, e_i, e_{i+1}, \ldots, e_t \rangle$, t being the number of events relevant to that case.

4.2. Horizontal partitioning

Following the method described in section 3.1, we began by identifying the unique activities of the process. Since we assume that all we can observe is the event log, it is possible that there exist some activities that are part of the process, yet they were not registered to the log. This is the known issue of *completeness* that is inherent in every process mining problem. Therefore, we shall proceed with the activities observed in the log. There is however, the following issue on identifying the unique activities in the original log $\mathcal{L}$: We expect the name of the activity and its code to be unique. However, none of them is. Consequently, to end up with a set of unique activities $\mathcal{A}$, we combined the two fields. This combination returned a set of 677 unique elements. The corresponding connectivity matrix $\mathbf{W}$ (see eq. 1) is a 677×677 symmetric matrix with 7,571 non-zero elements.

The critical decision of this stage is about the number of clusters. To make such a decision, we performed an exploratory analysis with visual aids of figure 2. More specifically, in figure 2a, we present a clustergram [44], which illustrates the number of clusters at the x-axis, and the cluster centers multiplied by the first component of the principal components of the original data at the y-axis. Then the cluster means are connected with parallelograms that indicate how many activities from a cluster are assigned to a cluster in the subsequent clustering test. In practice, when an increase to the number



(a) Clustergram

(b) Top-50 Eigenvalues of matrix $\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$

Figure 2: Selecting the number of clusters

of clusters brings a "split", that is an indication that the increment is meaningful, and when it does not, it is a recommendation to stop. In figure 2b, we plot the eigenvalues of the matrix $\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$, and we expect to see a sudden drop in the plot.

Both figures suggest that seven is an informed choice for the number of clusters. Therefore, we applied the procedure described in section 3.1 to obtain the seven clusters. A concise description of the results is presented in table 3.

4.3. Building a case log

To build the case log $\mathcal{L}_\mathcal{C}$ we added a characteristic h (a column) for each diagnosis variable and for each treatment variable of the original event log. In particular, the event log $\mathcal{L}$ related every event to a set of diagnosis and to a set of treatment variables. All variables of this kind are binary variables (1 if that diagnosis/treatment code was noted, 0 otherwise). To derive the corresponding case characteristics we applied the existential quantifier. In other words, if any of the events relevant to a case had a "true" value for the reference variable, the case characteristic was taking the value "true", otherwise we assigned the value "false". For instance, for the variable "Diagnosis M13":

$$\#_{Diag-M13}(c) = \begin{cases} 1, & \exists e \in \#_\tau(c) : \#_{Diag-M13}(e) = 1 \\ 0, & \text{otherwise} \end{cases}$$

Region	Size	Description
1	215	Surgical procedures. Perioperative diagnostic and supportive care (anesthesia, histology)
2	173	Diagnostic and interventional radiology and related procedures
3	78	Investigation of kidney and urinary tract related disorders (baseline and immunological work-up)
4	64	Short clinic for chemotherapies and minor (surgical) interventions
5	60	Microbiological (infection-related) work-up
6	44	Screening for short hospitalization, or day-clinic procedures
7	43	Anemia investigation and general outpatient work-up

Table 3: Describing the activities composition of regions. The descriptions were made by a medical doctor by checking the activities in every region.

After this manipulation, we dropped the characteristics $Diag-X822, Diag-X821, Diag-X106, Diag-X823, Diag-X839$ because they were "false" for all the cases. We applied the same existential quantifier for the variable "Urgent". Next, we exploited the organizational perspective to derive four additional characteristics, one to indicate the department where the case was initiated, one to indicate where the case ended, one to count the number of different departments that were visited, and one to display and the most frequently visited department during each case. In particular:

$$\#_{Start.Dep}(c) = \#_{Dep}(e_1 \in \#_\tau(c))$$

and likewise

$$\#_{End.Dep}(c) = \#_{Dep}(e_t \in \#_\tau(c))$$

The number of unique departments visited was calculated as the cardinality of the set of the departments that were involved in the activities of the trace:

$$\#_{N.Dep.Visited}(c) = |\{\#_{Dep}(e), \forall e \in \#_\tau(c)\}|$$

and the most frequently visited department as:

$$\#_{MostFreqDep}(c) = \#_{Dep}(e_i \in \#_\tau(c)) : e_i = \underset{\#_{Dep}(e_i \in \#_\tau(c))}{\arg\max} \; Freq(Dep)$$

where $Freq(Dep) = \sum_{i=1}^{t} [\#_{Dep}(e_i) = Dep]$.

The age case characteristic was calculated as the arithmetic mean of the values of the corresponding event variable for all the events of the case. Finally, we added to $\mathcal{L}_\mathcal{C}$ one column for each cluster that we discovered by following the procedure we describe in section 3.1 and we exemplify in section 4.2, and we labeled them as "region" plus an integer value to mark the specific cluster. Following the binary scale option, to assign the value to these variables (region1, region2, etc.), we put 1 if any of the activities that is member of the corresponding cluster is also member of the trace, and 0 otherwise. When following the options "percentage of cluster" or "percentage of trace" (see section 3.2), we put the calculated percentages to those variables. Therefore, actually, we created three datasets that are identical in everything except their values of the "region" variables.

4.4. Case characteristics evaluation

During the Stage 3 of our application, we evaluated the case characteristics using the dataset with the regions as "percentage of cluster". We used the ReliefF attribute evaluator implemented in the Weka platform [45]. We computed the scores and ranks for each attribute independently for each region-target. For each region we performed a 10-fold cross validation and we set the number of neighbors $k = 10$ of the evaluator. The final results is a list of ranks and a list of scores for each characteristic per region.

Hence, we can calculate the average rank for each characteristic per region. Then, by taking the minimum value of the average rank for each characteristic across the different regions, we shall get an indicative ranking list of the importance of the characteristics with respect to their connection to the process behavior. Figure 3 shows the top-30 characteristics and their ranks in the different regions. The ranks are represented as circles, where the darker and the larger circle indicates the higher rank of the characteristic for the corresponding region.

4.5. Insights

Following the procedures of the proposed approach, the output (a matrix with the scores/ranks of characteristics per region) is subject to qualitative interpretations. Although it is clear that the interpretations will always be case specific, we can provide several guidelines to decipher the results (which will expectedly take the shape of Figure 3.

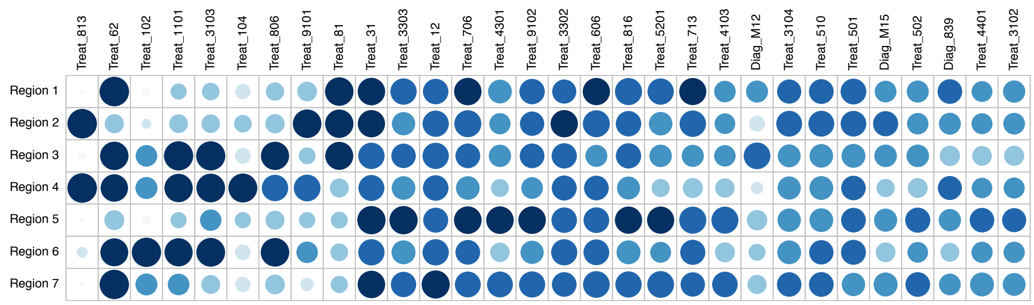


Figure 3: The average rankings of the top-30 features for every region. The darker and the larger the circle, the higher the rank of the feature.

The most intuitive explanation can be derived by looking at characteristics that have detectable "high" performance in any particular region. For instance, in our case study, the characteristics labeled "Treat_813" (i.e., when a patient receives the treatment with code 813) associates with regions 2 and 4, namely with a process behavior that drives the patient to perform activities related to radiology and chemotherapies (see Table 3). Similarly, we can associate "Treat_102" with screening activities for short hospitalization, and "Treat_104" to short clinics. We can even observe characteristics that are associated with sets of regions, like for example, "Treat_81" which directs patients to regions 1, 2 and 3.

The point of view can be reversed so as to derive insights by looking at regions and identifying the characteristics that exhibit a high relevance. For example, we observe that activities related to the microbiological workup (region 5) occur when cases are characterized by having "Treat_31", "Treat_3303", "Treat_706", "Treat_4301", "Treat_9102", "Treat_816", and "Treat_5201".

An additional guideline to parse the results for insights is to inverse the logic, and look for characteristics that "avoid" regions. To get this potentiality across, we shall look at the characteristic "Treat_62" (the second column in Figure 3). This characteristic is strongly associated with all regions, except 2 and 5. Therefore, we could support a claim that this characteristic puts off the behaviors implied by region 2 and region 5.

Finally, two extra guidelines refer to looking for characteristics that prompt similar (or dissimilar!) behavior. For instance, in our case study, we regard the characteristics "Treat_1101", "Treat_3103", and "Treat_806" to stimulate similar behavior (regions 3, 4, and 6) or the characteristic “Treat_62”, to be quite unique in its associations’ pattern.

5. Conclusions

This paper contributes to the general correlation problem by providing process analysts with an additional potential to relate process instances characteristics to their flows. This is a hard and challenging task on the visionary path of evidence-based process improvement and redesign. A three-staged methodology is proposed to address a number of challenges.

Starting with an horizontal partitioning technique, we devised regions of strongly connected activities to define process behavior that are neither too inclusive nor too granular, and can reflect more general behaviors. Then, through a set of guided log manipulations, we presented how an appropriate case log can be built to host various perspectives of the characteristics. It is important to recall that no additional data is required during this stage (as well as in no other stage) from process stakeholders, since characteristics can be seamlessly derived through the event log. During the third stage, we leverage the attribute estimation problem of the machine learning field, and treat it in a multi-target prediction setting, to connect characteristics with process flows, and thus to discover their influence in process behavior.

Since this is essentially a process mining approach, it inherits the issues related to this paradigm. Of particular relevance for this work is the issue of *completeness* or the so-called “*snapshot*” problem. This refers to the situation where cases may have a lifetime longer than the time-window of the event log, hence it is possible that some activities are not logged (i.e., the event log only provides a snapshot of the process). Certainly, if this occurs, the assumption that we can assess the process behavior by considering the finite set of activities that is recorded in the log, does not hold. However, if the average duration of the cases is significantly smaller than the duration of the recorded period, the “snapshot” issue is not expected to get raised. Continuing with the inherent limitations of our approach, we shall briefly discuss the noise effects. Noise can refer both to incorrect logging, as well as to the fact that the event log contains rare and infrequent behavior not representative for the typical behavior of the process [46]. The latter can only be partially addressed by the robust clustering approach described in section 3.1, while for the former (incorrect logging), since this is an evidence-based approach, if evidence (data) are not of good quality, we are afraid that there are not much that the method can do.

Another limitation of the proposed approach is that it is confined to the analysis phase. As we exemplified in Section 4, the proposed approach is suitable for a so-called *post-mortem* analysis of a business process, since the ultimate outcome is a set of off-line recommendations. To unmask or to augment its value, it will be necessary to integrate it in the entire BPM life-cycle. To this end, a need for a relevant framework emerges, which will provide an all-embracing position of the elements of the process through a meta-model, similarly to the work of [47], to globally promote adaptation and responsiveness to the contextual elements. This is actually a part of future work, on the way to deliver a comprehensive tool not just for the general correlation problem, but for business process management in general.

Acknowledgement

The authors are grateful to Vasilios Devetzis, MD. Without him, it would not be possible to interpret the horizontal partitioning results and present Table 3.

References

[1] I. Davies, P. Green, M. Rosemann, M. Indulska, S. Gallo, How do practitioners use conceptual modeling in practice?, Data & Knowledge Engineering 58 (3) (2006) 358–380.

[2] C. W. Günther, Process mining in flexible environments, PhD dissertation, Technische Universiteit Eindhoven, Eindhoven (2009).
URL `http://alexandria.tue.nl/extra2/200911996.pdf`

[3] M. de Leoni, W. M. van der Aalst, M. Dees, A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs, Information Systems 56 (2016) 235–257.

[4] D. R. Ferreira, D. Gillblad, Discovering process models from unlabelled event logs, in: International Conference on Business Process Management, Springer, 2009, pp. 143–158.

[5] M. Rosemann, J. Recker, C. Flender, Contextualisation of business processes, International Journal of Business Process Integration and Management 3 (1) (2008) 47–60.

[6] J. d. E. Santo Carvalho, F. M. Santoro, K. Revoredo, A method to infer the need to update situations in business process adaptation, Computers in Industry 71 (2015) 128–143.

[7] M. Anastassiu, F. M. Santoro, J. Recker, M. Rosemann, The quest for organizational flexibility: driving changes in business processes through the identification of relevant context, Business Process Management Journal 22 (4) (2016) 763–790.

[8] W. M. van der Aalst, Mining Additional Perspectives, Springer Berlin Heidelberg, Berlin, Heidelberg, 2016, pp. 275–300.

[9] S. Suriadi, C. Ouyang, W. M. van der Aalst, A. H. ter Hofstede, Root cause analysis with enriched process logs, in: International Conference on Business Process Management, Springer, 2012, pp. 174–186.

[10] R. J. C. Bose, R. S. Mans, W. M. van der Aalst, Wanna improve process mining results?, in: Computational Intelligence and Data Mining (CIDM), 2013 IEEE Symposium on, IEEE, 2013, pp. 127–134.

[11] S. Smirnov, H. A. Reijers, M. Weske, From fine-grained to abstract process models: A semantic approach, Information Systems 37 (8) (2012) 784–797.

[12] C. W. Günther, A. Rozinat, W. M. van Der Aalst, Activity mining by global trace segmentation, in: International Conference on Business Process Management, Springer, 2009, pp. 128–139.

[13] M. Dumas, M. La Rosa, J. Mendling, H. A. Reijers, Process Discovery, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 155–184.

[14] A. Rozinat, W. M. van der Aalst, Decision Mining in ProM, in: D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, S. Dustdar, J. L. Fiadeiro, A. P. Sheth (Eds.), Business Process Management, Vol. 4102, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 420–425,

dOI: 10.1007/11841760_33.
URL http://link.springer.com/10.1007/11841760_33

[15] D. R. Ferreira, E. Vasilyev, Using logical decision trees to discover the cause of process delays from event logs, Computers in Industry 70 (Supplement C) (2015) 194 – 207. doi:https://doi.org/10.1016/j.compind.2015.02.009.

[16] D. Grigori, F. Casati, U. Dayal, M. Shan, Improving business process quality through exception understanding, prediction, and prevention, in: VLDB 2001, Proceedings of 27th International Conference on Very Large Data Bases, September 11-14, 2001, Roma, Italy, 2001, pp. 159–168.

[17] A. Rozinat, R. S. Mans, M. Song, W. M. van der Aalst, Discovering simulation models, Information systems 34 (3) (2009) 305–327.

[18] M. De Leoni, W. M. van der Aalst, Data-aware process mining: discovering decisions in processes using alignments, in: Proceedings of the 28th annual ACM symposium on applied computing, ACM, 2013, pp. 1454–1461.

[19] E. Bazhenova, S. Buelow, M. Weske, Discovering decision models from event logs, in: International Conference on Business Information Systems, Springer, 2016, pp. 237–251.

[20] F. Mannhardt, M. de Leoni, H. A. Reijers, W. M. van der Aalst, Data-driven process discovery: revealing conditional infrequent behavior from event logs, in: Advanced Information Systems Engineering: 29th International Conference, CAiSE 2017, Springer, 2017.

[21] F. M. Maggi, A. J. Mooij, W. M. van der Aalst, User-guided discovery of declarative process models, in: Computational Intelligence and Data Mining (CIDM), 2011 IEEE Symposium on, IEEE, 2011, pp. 192–199.

[22] F. M. Maggi, R. J. C. Bose, W. M. van der Aalst, Efficient discovery of understandable declarative process models from event logs, in: International Conference on Advanced Information Systems Engineering, Springer, 2012, pp. 270–285.

[23] M. Pesic, H. Schonenberg, W. M. van der Aalst, Declare: Full support for loosely-structured processes, in: Enterprise Distributed Object Computing Conference, 2007. EDOC 2007. 11th IEEE International, IEEE, 2007, pp. 287–287.

[24] S. Ferilli, Woman: logic-based workflow learning and management, IEEE Transactions on Systems, Man, and Cybernetics: Systems 44 (6) (2014) 744–756.

[25] C. Di Ciccio, F. M. Maggi, J. Mendling, Efficient discovery of target-branched declare constraints, Information Systems 56 (2016) 258–283.

[26] S. Schönig, A. Rogge-Solti, C. Cabanillas, S. Jablonski, J. Mendling, Efficient and customisable declarative process mining with sql, in: International Conference on Advanced Information Systems Engineering, Springer, 2016, pp. 290–305.

[27] M. L. Bernardi, M. Cimitile, C. Di Francescomarino, F. M. Maggi, Do activity lifecycles affect the validity of a business rule in a business process?, Information Systems 62 (2016) 42–59.

[28] R. J. C. Bose, F. M. Maggi, W. M. van der Aalst, Enhancing declare maps based on event correlations, in: Business Process Management, Springer, 2013, pp. 97–112.

[29] H. Nguyen, M. Dumas, M. La Rosa, F. M. Maggi, S. Suriadi, Mining Business Process Deviance: A Quest for Accuracy, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014, pp. 436–445. doi:10.1007/978-3-662-45563-0_25.

[30] H. Nguyen, M. Dumas, M. L. Rosa, F. M. Maggi, S. Suriadi, Business process deviance mining: Review and evaluationarXiv:1608.08252v1.

[31] G. Tsoumakas, I. Katakis, Multi-Label Classification : An Overview, International Journal of Data Warehousing and Mining 3 (September) (2007) 1–13. doi:10.1109/ICWAPR.2007.4421677.

[32] E. Spyromitros-Xioufis, G. Tsoumakas, W. Groves, I. Vlahavas, Multi-target regression via input space expansion: treating targets as inputs, Machine Learning 104 (1) (2016) 55–98. arXiv:1211.6581, doi:10.1007/s10994-016-5546-z.

[33] W. M. van der Aalst, Distributed process discovery and conformance checking, in: International Conference on Fundamental Approaches to Software Engineering, Springer, 2012, pp. 1–25.

[34] P. Delias, K. Lakiotaki, Discovering process horizontal boundaries to facilitate process comprehension, International Journal of Operations Research and Information Systems 9 (2) (2018) 1–31. doi:10.4018/IJORIS.2018040101.

[35] IEEE standard for extensible event stream (XES) for achieving interoperability in event logs and event streams, IEEE Std 1849-2016 (2016) 1–50doi:10.1109/IEEESTD.2016.7740858.

[36] M. Dumas, W. M. van der Aalst, A. H. ter Hofstede, Process-aware information systems: bridging people and software through process technology, John Wiley & Sons, 2005.

[37] W. M. van der Aalst, Data Science in Action, Springer Berlin Heidelberg, Berlin, Heidelberg, 2016, pp. 1–23.

[38] A. Pika, M. Leyer, M. T. Wynn, C. J. Fidge, A. H. Ter Hofstede, W. M. van der Aalst, Mining resource profiles from event logs, ACM Transactions on Management Information Systems (TMIS) 8 (1) (2017) 1.

[39] U. von Luxburg, A tutorial on spectral clustering, Statistics and Computing 17 (4) (2007) 395–416. doi:10.1007/s11222-007-9033-z.

[40] H. Fritz, L. A. García-Escudero, A. Mayo-Iscar, tclust: An R package for a trimming approach to cluster analysis, Journal of Statistical Software 47 (12) (2012) 1–26.

[41] M. Robnik-Šikonja, I. Kononenko, Comprehensible interpretation of reliefs estimates, in: Machine Learning: Proceedings of the Eighteenth International Conference on Machine Learning (ICML2001), Williamstown, MA, USA. San Francisco: Morgan Kaufmann, 2001, pp. 433–40.

[42] B. Van Dongen, Real-life event logs - hospital log (2011). doi:10.4121/uuid:d9769f3d-0ab0-4fb8-803b-0d1120ffcf54.

[43] R. J. C. Bose, W. M. van der Aalst, Analysis of patient treatment procedures., in: Business Process Management Workshops (1), Vol. 99, 2011, pp. 165–166.

[44] M. Schonlau, The clustergram: A graph for visualizing hierarchical and non-hierarchical cluster analyses, The Stata Journal 2 (4) (2002) 391–402.

[45] E. Frank, M. Hall, I. Witten, The weka workbench, Online Appendix for Data Mining: Practical Machine Learning Tools and Techniques, 4th edn. Morgan Kaufman, Burlington.

[46] W. M. van der Aalst, Getting the Data, Springer Berlin Heidelberg, Berlin, Heidelberg, 2016, pp. 125–162.

[47] T. da Cunha Mattos, F. M. Santoro, K. Revoredo, V. T. Nunes, A formal representation for context-aware business processes, Computers in Industry 65 (8) (2014) 1193–1214.