Subgroup Discovery in Process Mining

Mohammadreza Fani Sani[1(✉)], Wil van der Aalst[1], Alfredo Bolt[1], and Javier García-Algarra[2]

[1] Eindhoven University of Technology, Eindhoven, The Netherlands
{M.Fani.Sani,w.m.p.v.d.aalst,a.bolt}@tue.nl
[2] Telefonica, Madrid, Spain
fco.javier.garciaalgarra@telefonica.com

Abstract. Process mining enables multiple types of process analysis based on event data. In many scenarios, there are interesting subsets of cases that have deviations or that are delayed. Identifying such subsets and comparing process mining results is a key step in any process mining project.

We aim to find the statistically most interesting patterns of a subset of cases. These subsets can be created by process mining algorithms features (e.g., conformance checking diagnostics) and serve as input for other process mining techniques. We apply subgroup discovery in the process mining domain to generate actionable insights like patterns in deviating cases. Our approach is supported by the ProM framework. For evaluation, an experiment has been conducted using event data from a large Spanish telecommunications company. The results indicate that using subgroup discovery, we could extract interesting insights that could only be found by spitting the event data in the right manner.

Keywords: Process mining · Subgroup discovery · Pattern mining · Performance management · Quality of metrics

1 Introduction

Our society, organizations and IT systems depend on processes. Products and services can only be delivered efficiently and effectively when processes are running as planned. Process mining aims to discover, monitor, and enhance processes by extracting knowledge from event data that can be extracted from almost all modern [1].

Process Mining is able to bridge the gap between Business Process Modeling (BPM) and data driven methods like data mining and machine learning [2]. Process mining is able to analyze the actual processes without relying on simplistic models. There are basically two main types of data-driven analysis [3]:

- **Predictive analysis:** involving techniques that extract knowledge and rules to predict or classify samples, such as classification, regression and time series algorithms.

W. Abramowicz (Ed.): BIS 2017, LNBIP 288, pp. 237–252, 2017.
DOI: 10.1007/978-3-319-59336-4_17

- **Descriptive analysis:** involving techniques that discover interesting knowledge about samples and their attributes to explain the data (e.g. association rules).

In other words, descriptive analysis techniques extract patterns from the data with respect to properties and their values. For example, a manager wants to know in which situations customers have complaints. Descriptive analysis will not be able to predict the complaints; however, it will provide insights about various factors that may cause the complaints [5].

The lion's share of process mining research has been devoted to descriptive forms of analysis. Next to process discovery techniques, there have been approaches to group traces. The approach presented in [4] clusters traces thereby characterizing each cluster. However, in this method class of samples cannot be used. The approach presented in [5] extracts interesting patterns based on a class attribute. In many applications, stakeholders prefer to analyze and know more about a subset of cases rather than all the cases. Examples of interesting subsets (or target group) include:

- Deviating cases from the reference model
- Cases with high or low performance
- Cases with high profits for the company
- Unfinished or canceled cases
- Cases from a particular period
- Cases that pertain to users complaints
- Events related to particular products or services

Given such subsets of cases, it is of the utmost importance to see what kind of attributes they share. For example, discovering that deviating cases are caused by particular resources or limited to specific groups of customers. According to our knowledge, there is no research has been done to extract such information from event data. The main contribution of this paper is that we apply subgroup discovery techniques in the context of process mining domain, to discover the statistically most interesting patterns in a subset of cases called the target groups. The attributes and also the target group can be created based on features extracted using process mining algorithms (e.g., conformance checking or performance analysis). Moreover, our approach also produces insightful collections event logs that can be used as input for a range of existing process mining techniques (e.g., process discovery). In short, this approach will help process analyst to find what are distinctive attributes in a subgroup of cases. Such information assists further investigations like root cause analysis.

To evaluate the possibility of using this method in the reality, we provide a case study where we applied our proposed method on the ticket handling process of Telefonica. Figure 1, shows the process model of this process. The ticket handling process in Telefonica consists of the following main steps. First, a ticket is created through the 'New' activity and then it should be activated by conducting the 'Active' activity. After this activation, a ticket should be handled appropriately and consequently closed through the 'Solved' and 'Closed'

Table 1. Small fragment of the dataset provided by Telefonica, related to the ticket handling process.

CaseID	EventID	Operation	Resource	Group	Severity	Type	Creator	Date-Time
A1001	1	New	Sara	G17	Major	Claim	G1	20150711-10:12
A1001	2	Active	Jon	G17	Major	Claim	G1	20150711-10:19
A1001	3	Solved	Alex	G10	Major	Claim	G1	20150711-16:01
A1001	4	Closed	Alex	G10	Major	Claim	G1	20150711-16:21
A1002	5	New	Sara	G17	Minor	Order	G1	20150713-08:32
A1002	6	Active	Tim	G17	Minor	Order	G1	20150713-08:51
A1002	7	Canceled	Leo	G19	Minor	Order	G1	20150713-14:04
A1003	8	New	Sara	G17	Slight	Claim	G2	20150711-11:20
A1003	9	Active	Tim	G17	Slight	Claim	G2	20150711-11:27
A1003	10	Active	Tim	G17	Slight	Claim	G2	20150711-11:28
A1003	11	Canceled	Alex	G10	Slight	Claim	G2	20150712-09:51

activities. It is possible to interrupt the handling of a ticket by the 'Delayed' activity. Also, a ticket could be restored to the customer via the 'Restored' activity. There is also another possibility, namely: the cancellation of tickets by the 'Canceled' activity. This can happen at any point in their lifetime. We consider 'Canceled' and 'Closed' as the possible final activities of a ticket.

Every process may be executed for multiple cases (also called process instances). Each case is composed of a set of events that are stored in the event log. The standard format for storing an event log which is supported by the majority of process mining tools is XES [6]. In Table 1, a simple example of the event log for the Fig. 1 is shown that contains 3 cases. Cases A1001 and A1003 have 4 events and A1002 has 3 events. By using the CaseID field we know which events are related to particular cases. Note that, case A1002 is not completely "fitting" in the process model (there is one so-called "move in log" showing an event that happened in reality but could not happen according to the model). Furthermore, both events and cases may have attributes that can be used. For

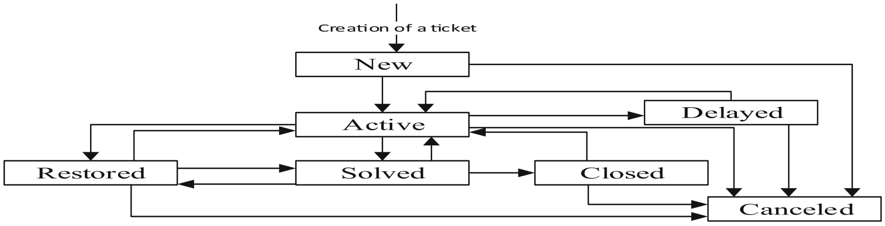


Fig. 1. A normative process model that describes the ticket handling process. This model was designed by Telefonica (of course such models can also be discovered based on the event log).

example, in Table 1, Resource and Group are event attributes. These attributes indicate that who is handled each event and do it in which organizational part of the company. Also, Severity that is a case property, shows the importance of different tickets (cases) in the event log.

The remainder of this paper is organized as follows. In Sect. 2, subgroup discovery is explained. In Sect. 3, we describe how we map and use subgroup discovery in the process mining domain. Section 4 describes the implementation of our approach. Next, Sect. 5 illustrates the usefulness of our approach through the application of our techniques to a real life dataset obtained from Telefonica. Finally, Sect. 6 concludes the paper.

2 Subgroup Discovery

Subgroup discovery was originally proposed by [7,8] and it is based on the idea of *local exceptionality detection* [9]. In contrast with most classification or prediction algorithms, subgroup discovery does not try to find rules that are used to decide or predict things for new instances of the problem. Also, unlike clustering methods, in this technique, we assume that we have a population of samples that have already a class label (e.g., deviating or not). As mentioned before, the aim of subgroup discovery algorithm is to discover patterns for particular class labels (target groups) [8]. In other words, we try to find the common characteristics in a subset of cases that are fewer happened in the other cases. For example, discovering cases that are delayed caused by particular resources or limited to specific type of tickets. Subgroup discovery is used in various domains including the filed of Bioinformatic, e-learning and medical domain [11]. Also in [12] this technique is extended to used multi class data.

We define a subgroup as ($ValueSet \rightarrow Target$) where *ValueSet* is an ordered list of independent attributes having specific values. In addition, *Target* is the desired class of samples that we are interested in analyzing them like deviated cases. For example, S_1, S_2 and S_3 are three examples of possible subgroups:

$$S_1 : Type = \text{“}Claim\text{”} \wedge Severity = \text{“}Minor\text{”} \rightarrow Target = Deviating$$
$$S_2 : Creator = \text{“}G2\text{”} \rightarrow Target = Deviating$$
$$S_3 : Severity = \text{“}Major\text{”} \rightarrow Target = \overline{Deviating}$$

Using subgroup discovery we want to discover interesting subgroups. According to [8], a subgroup is interesting if it satisfies the following conditions:

- it is of considerable size and
- it has the most unusual statistical distribution characterization (distribution of different classes in the subgroup compared to their distribution in whole samples)

In Fig. 2, this concept is illustrated. Consider that the class feature is depicted by a red dash or a blue plus. In this figure, three subgroups are shown. Subgroup (a) is not an interesting one because there are too few samples included in it. In other

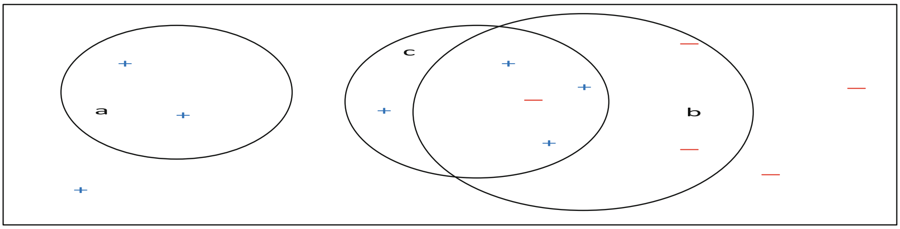


Fig. 2. Three different subgroups. Subgroup (a) is very specific, subgroup (b) has a class distribution similar to the whole and thus not "unusual" enough. The subgroup (c) is an interesting subgroup because it has sufficient samples with a class distribution sufficiently different from the rest. (Color figure online)

words, this subgroup is too specific. In contrast, subgroup (b) has more samples, but the distribution of samples in it is not unusual, because it is the same as the whole population. Finally, the subgroup (c) has a substantial number of samples and has an atypical distribution at the same time, therefore it is considered as an interesting subgroup. It should be noted, it is not required that all samples included in a subgroup have the same class (see for example subgroup (c)). Also, one sample could be placed in more than one subgroup simultaneously (or not placed in any subgroup), i.e., subgroups are not a partitioning of the whole set.

In this approach, we consider only the standard definition of interestingness (based on size and statistical difference), however other definitions could be applied that incorporate domain or business knowledge.

Many measures have been proposed in the literature to quantify the quality of a subgroup and its interestingness. Table 2 summarizes several of the proposed metrics mentioned in papers like [3]. Many of these measures have also been applied in the association rules mining field. To illustrate them in a better way, we use the contingency table presented in Table 3. This table is a useful way to examine relations between categorical variables [10]. A sample matches a particular *ValueSet* if its attributes have values in the ranges defined by *ValueSet*. Similarly, a sample matches a particular *Target* if its class attributes has a value defined by *Target*. In this table, the number of samples that match the *ValueSet* and the number of samples that match the defined *Target* are indicated by n_V and n_T respectively. Also, the number of samples that match both the selected *ValueSet* and *Target* is indicated by n_{VT}. In addition, n_S is the total number of samples. Note that $n_{\overline{T}}$ and $n_{\overline{V}}$ are define the number of samples do not match the *Target* and the selected *ValueSet* respectively. Therefore, $n_S = n_T + n_{\overline{T}} = n_V + n_{\overline{V}}$.

A higher value of the *coverage* metric means that the subgroup has more samples. *coverage* $= 1$ indicates that the corresponding subgroup includes all the samples. Therefore, an interesting subgroup should have a *coverage* that is high enough. A value of 1 (or 0) in the *support* metric indicates all the samples (or none of them) are match both the *ValueSet* and *Target* class. If a subgroup has a value of 1 in its *confidence* metric, it indicates that if a sample match

Table 2. List of various measures used in subgroup discovery domain.

Measure	Formula	Range
Coverage	$Cov(Subgroup) = \frac{n_{ValueSet}}{n_{Samples}} = \frac{n_V}{n_S}$	[0, 1]
Support	$Supp(Subgroup) = \frac{n_{ValueSet \wedge Target}}{n_{Samples}} = \frac{n_{VT}}{n_S}$	[0, 1]
Confidence	$Conf(Subgroup) = \frac{n_{ValueSet \wedge Target}}{n_{ValueSet}} = \frac{n_{VT}}{n_V}$	[0, 1]
Lift	$Lift(Subgroup) = \frac{Supp(Subgroup)}{Supp(Valueset) \times Supp(Target)} = \frac{n_{VT} \times n_S}{n_V \times n_T}$	(0, ∞)
Added value	$AddedValue(Subgroup) = \frac{n_{VT}}{n_V} - \frac{n_T}{n_S}$	(−1, 1)
Precision [7]	$Q_g(Subgroup) = \frac{TP}{FP+g} = \frac{n_{VT}}{n_{V\overline{T}}+g}$	(0, ∞)
Unusualness [13]	$WRAcc(Subgroup) = \frac{n_V}{n_S} \times (\frac{n_{VT}}{n_V} - \frac{n_T}{n_S})$	[−0.25, 0.25]
PS [14]	$PS(Target \rightarrow ValueSet) = \frac{n_{VT}}{n_S} - \frac{n_V \times n_S}{{n_S}^2}$	[−0.25, 0.25]

Table 3. Contingency table shows counts of all possible conjunctions of *ValueSet* and *Target* group.

	Target	$\overline{Target}$	
$ValueSet$	n_{VT}	$n_{V\overline{T}}$	n_V
$\overline{ValueSet}$	$n_{\overline{V}T}$	$n_{\overline{VT}}$	$n_{\overline{V}}$
	n_T	$n_{\overline{T}}$	n_S

the selected *ValueSet* it should match the *Target* too. The *lift* metric computes how dependent (or independent) are the *Valueset* and *Target*. If *Lift* equals 1 then they are independent. However, a value higher than 1 suggests a positive correlation and a value lower than 1 indicates a negative correlation. If the *added value* metric has a value of 0, it suggests that the distribution of the classes are similar in both subgroup and total samples and consequently, the *ValueSet* has no influence on the *Target* distribution. In addition, a higher positive (or lower negative) value for this measure, suggests higher positive (or negative) effect on the distribution of the target feature.

Precision measures the quality of a subgroup by computing ratio of different classes when samples match the selected *ValueSet*. In its formula, g is the generalization parameter which is usually in the range [0.5, 100]. The *unusualness* value of a subgroup is computed based on both the *coverage* and *added value* of it. It could be proven that $Unusualness(subgroup) = WRAcc(ValueSet \rightarrow Target)$ is equal to $PS(Target \rightarrow ValueSet)$ which is widely used in field of association rules mining. Both of them equal to $\frac{n_V \times n_{VT}}{n_S \times n_V} - \frac{n_V \times n_S}{{n_S}^2}$ (one of difference between association rule and subgroup discovery is in association rule we extract $Target \rightarrow ValueSet$ pattern, but here we are interested in $ValueSet \rightarrow Target$ patterns.) These measures account for *coverage* (size of a subgroup) and *added value* (unusual statistical distribution of subgroup) at the same time $(Conf(Target \rightarrow ValueSet) \times Supp(Target))$.

In this paper, we mainly use the *unusualness* measure and it's range is in [−0.25, 0.25]. *Unusualness* equals 0, suggests that a subgroup would not be interesting; however, a higher positive value indicates that the *ValueSet* has higher effect on the *Target* compare to the whole samples. Also, lower negative value for this measure, shows that the samples match the selected *ValueSet* have lower fewer in the *Target* class compare to other class. In many applications, discovering subgroups with negative *unusualness* would be also valuable. Thus, we use the absolute value of *unusualness* ($|WRAcc(subgroup)|$) or *RuleInterestVariant* [15].

3 Applying Subgroup Discovery in Process Mining

In this section, we formally define how to apply subgroup discovery in the field of process mining. The architecture of proposed method is illustrated in Fig. 3. The starting point of our method is an event log. An event log may contain many cases and each case has a set of associated events. Most of the process mining techniques consider events as the starting point for process analysis. In this research, we focus on cases rather than events.

Therefore, in the next step we extract properties for all cases. There are three types of properties in process mining: properties that are related to (a) cases, (b)events, and (c) processes mining properties. In general, a case property is the same for all the events of a specific case. However, for event attributes, the values could be different (or simply missing) for individual events within a case. For example, in Table 1, CaseID, Severity, Type, and Creator are case properties and EventID, Operation, Resource, and Group are event properties. Properties of events can also be mapped to cases properties indirectly. In Fig. 4, an example of such mapping is shown. All possible values of each event property are mapped to a case property. If in any event of a case this value occurred, then the corresponding property of case equals 1, otherwise, it will be 0 (here we use existence function, but other functions like frequency could be used as well). To explain more, a resource of event 6 is "Tim" and because this event belongs to case "A1002", the value of "R:Tim" for this case equals 1.

The third type of properties, the so-called process mining properties, are obtained by performing some kind of computation over the events within a case. Examples include performance metrics (sojourn time, waiting time, etc.) or conformance checking metrics (fitness, precision, counts of move on logs and model, etc.). To extract some of the mentioned features we can optionally provide a process model (that could be given as a reference model or discovered by some process discovery algorithm). Some examples of process mining properties for the event log of Table 1 are given in Table 4. Note that process mining techniques

Table 4. Some process mining properties for the event log of Table 1. To compute alignment costs we use standard cost.

CaseID	Event count	VariantID	Case duration	Fitted model	Alignment cost	Completeness
A1001	4	X1	369 min	Yes	0	Complete
A1002	3	X2	332 min	Yes	0	Complete
A1003	4	X3	22.5 h	No	1	Complete

like conformance checking can be used as input for subgroup discovery. However, the very same techniques can be applied to the discovered subgroups in a later phase. This shows the close interaction between process mining techniques and subgroup discovery.

According to Fig. 3, the output of *Property Extractor* component will be a matrix where each row corresponds to a case and each of its columns refers to a property.

Definition 1 (Universes). *$U_S = \mathcal{P}(U_V)$ is the universe of value collections. $U_H = \mathcal{P}(U_S)$ is the universe of sets of value collections (set of sets). Note that $v \in U_V$ is a single value (e.g. $v = Claim$), $V \in U_S$ is a value collection (e.g., $V = \{Claim, Order, Query\}$).*

Definition 2 (Case Base). *A* case base *$CB = (C, P, \pi)$ defines a set of cases C, a set of properties P, and a function $\pi \in (P \rightarrow (C \rightarrow U_V))$. For any properties $p \in P$, $\pi(p)$ (denoted π_p) is a partial function mapping cases onto values. If $\pi_p(c) = v$, then case $c \in C$ has a property $p \in P$ and the value of this property is $v \in U_V$.*

Therefore, P includes case, events and process properties and each property $p \in P$ corresponds to column in the extracted matrix shown in Fig. 3. According

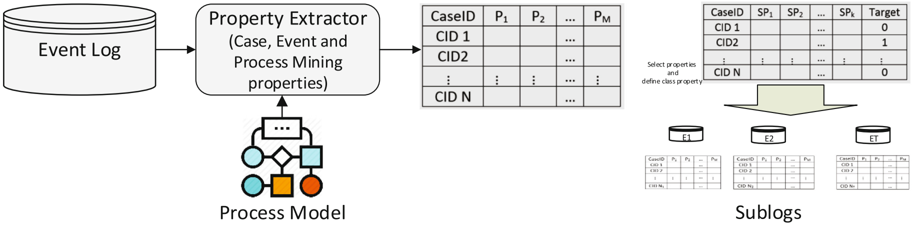
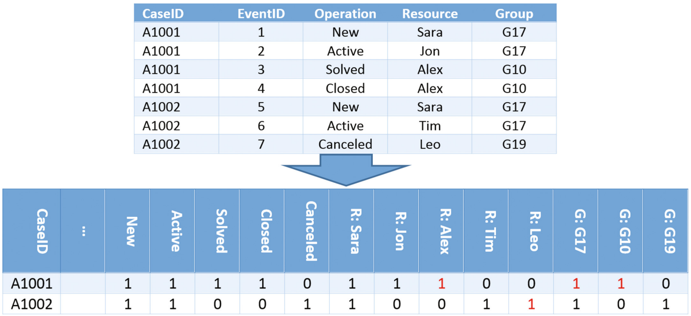


Fig. 3. The architecture of proposed method.



CaseID	EventID	Operation	Resource	Group
A1001	1	New	Sara	G17
A1001	2	Active	Jon	G17
A1001	3	Solved	Alex	G10
A1001	4	Closed	Alex	G10
A1002	5	New	Sara	G17
A1002	6	Active	Tim	G17
A1002	7	Canceled	Leo	G19

CaseID	...	New	Active	Solved	Closed	Canceled	R: Sara	R: Jon	R: Alex	R: Tim	R: Leo	G: G17	G: G10	G: G19
A1001		1	1	1	1	0	1	1	1	0	0	1	1	0
A1002		1	1	0	0	1	1	0	0	1	1	1	0	1

Fig. 4. Mapping the properties of events to case (trace) properties. Values of event attributes are transformed to a case property. For each case, it is computed whether the property is present. The values that are indicated with red color, explain that we use the existence of an attribute value. (Color figure online)

to the method's architecture (Fig. 3), for each case in the *case base* (that is created by the property extractor), a class attribute and intended properties are selected. The class attribute is a binary property that helps us to divide cases into two subsets (two classes). The first subset contains cases that we are particularly interested in analyze them (e.g., cases with delay or deviation). The rest of cases are placed in the other subset. By defining a class attribute, we specify which of the cases are interesting for analysis. In addition, not all of the properties in the *case base* may be noticeable and we should set aside them from properties that will be analyzed them in this subset. We name this subset *target group* and define it formally as follows:

Definition 3 (Target Group). *$TG(CB, Att, \pi_{class})$ is a target group of $CB = (C, P, \pi)$ where π_{class} is a membership function mapping cases to their relative classes. If a case $c \in C$ belongs to our intended subset, then $\pi_{class}(c) = 1$ otherwise $\pi_{class}(c) = 0$. Attributes $Att \subseteq P$ is a subset of the case base properties that we are interested to analyze their effect on the intended subset of cases.*

Therefore, we could say that *TG* specifies a subset of properties in the *case base* that we want to analyze them and class of each case. Using this definition, we take a case base as an input and returning a subset of it's properties and the class value of each case.

At last, by applying subgroup discovery on the *target group* we will discover many subgroups. Here we formally define a subgroup as the following definition.

Definition 4 (Subgroup). *S(TG,att,vs) is a* subgroup *of attribute $att \in Att$ when $\pi_{att} = vs$ on the target group* TG*. Each* subgroup *is a subset of cases in the* TG *that in these cases, the value of attribute att equals to vs.*

As an example, *att* can be *Type* and *vs* equals *Claim*. The resulted *subgroup* is the subset of cases in the *TG* and the value of "Type" property for these case is *claim*.

Considering several properties in the *target group*, we will have many subgroups. However, the discovered subgroups are different based on their size, interestingness, distribution, and effects of them on the *target group*. We use *unusualness* measure to compute the interestingness of discovered subgroups on the *target group*. We name this measure *Impact Effect* and denote it by $IE(subgroup)$. The higher value of *IE* suggests higher positive Influence of the subgroup. As mentioned before, we aim to discover subgroups with higher $|IE|$ values. Using this definition we can compute the interestingness (or *unusualness*) of each subgroup on the *target group*.

Until now, we just considered one attribute in the *ValueSet* of a *subgroup*. However, it is possible to have a subgroup with multiple attributes. The complexity of a subgroup could be defined by the number of attributes in its *ValueSet* [3]. For example, $S : Type = \text{"}Calm\text{"} \wedge\ Severity = \text{"}Major\text{"} \rightarrow Target = Deviating$ is a subgroup with multiple attributes and its complexity equals 2. Note that in combination of properties, each property should not appear more than one time in a subgroup.

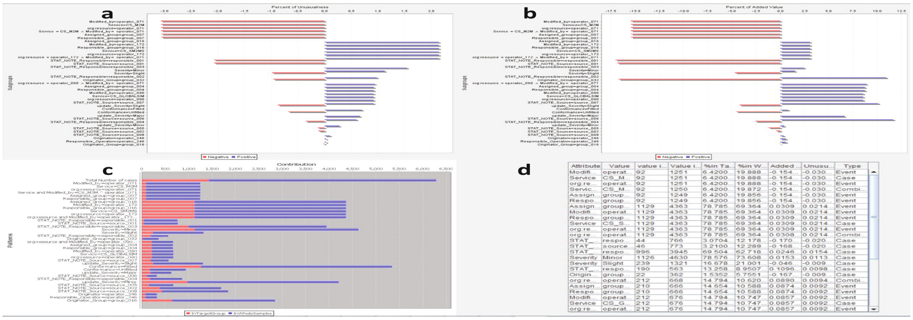
However, computing all possible subgroups would be very time-consuming. There are many methods proposed to overcome this issue [21]. Here, we use minimum coverage of subgroups. So, subgroups with *Cov(subgroup)* lower than the minimum threshold are not considered. Note that if the coverage value of $(ValueSet_1 \rightarrow Target)$ is Cov_1, the *coverage* value of $(ValueSet_1 \wedge ValueSet_2 \rightarrow Target)$, by definition, is less than or equal to Cov_1. Thus, if a subgroup does not contain sufficient samples to have the minimum coverage, no other subgroups included in this subgroup have a higher coverage and there is no need to consider them.

In the final step of our approach we apply process mining techniques to the subgroups created. For each subgroup, we could extract a sublog (i.e., a subset of the main event log). A wide range of process mining algorithms ranging from dotted chart [16] and process comparator [17] to the inductive miner [18] and various conformance checkers [19] could be applied on these sublogs for further analysis.

4 Implementation

To make it possible to apply subgroup discovery approach in the process mining context, a Subgroup Discovery plugin has been developed in ProM framework. ProM is an open source tool that allows to use and implement lots of different techniques in the field of process mining [20]. This tool can be freely downloaded from www.promtools.org.

The *Subgroup Discovery* plugin takes two event logs as input, one contains all the case samples (*Case Base*) and the other one is related to the subset of cases that we want to characterize (i.e., target group). Therefore, the second event log should be a subset of the first event log. Furthermore, regarding the output range of *unusualness* metric (and also *PS* metric) is in [−0.25, 0.25], we use range bar chart (it is also called *Tornado* chart) to visualize the impact of each subgroup on the target group. A screenshot of an example output result of subgroup discovery obtained using our tool is shown in Fig. 5.



Attribute	Value	value i...	Value i...	%in Ta...	%in W...	Added ...	Unusu...	Type
Modifi...	operat...	92	1251	6.4200...	19.888...	-0.154...	-0.030...	Event
Service	CS_M...	92	1251	6.4200...	19.888...	-0.154...	-0.030...	Case
org:re...	operat...	92	1251	6.4200...	19.888...	-0.154...	-0.030...	Event
Servic...	CS_M...	92	1250	6.4200...	19.872...	-0.154...	-0.030...	Combi...
Assign...	group...	92	1249	6.4200...	19.856...	-0.154...	-0.030...	Event
Respo...	group...	92	1249	6.4200...	19.856...	-0.154...	-0.030...	Event
Assign...	group...	1129	4363	78.785...	69.364...	0.0309...	0.0214...	Event
Modifi...	operat...	1129	4363	78.785...	69.364...	0.0309...	0.0214...	Event
Respo...	group...	1129	4363	78.785...	69.364...	0.0309...	0.0214...	Event
Service	CS_S...	1129	4363	78.785...	69.364...	0.0309...	0.0214...	Case
org:re...	operat...	1129	4363	78.785...	69.364...	0.0309...	0.0214...	Event
org:re...	operat...	1129	4363	78.785...	69.364...	0.0308...	0.0214...	Combi...
STAT_...	respo...	44	766	3.0704...	12.178...	-0.170...	-0.020...	Case
STAT_...	source...	46	773	3.2100...	12.289...	-0.168...	-0.020...	Case
STAT_...	respo...	996	3945	69.504...	62.718...	0.0246...	0.0154...	Case
Severity	Minor	1126	4630	78.576...	73.608...	0.0153...	0.0113...	Case
Severity	Slight	239	1321	16.678...	21.001...	-0.046...	-0.009...	Case
STAT_...	respo...	190	563	13.258...	8.9507...	0.1096...	0.0098...	Case
Origin...	group...	22	362	1.5352...	5.7551...	-0.167...	-0.009...	Case
org:re...	operat...	212	668	14.794...	10.620...	0.0890...	0.0094...	Combi...
Assign...	group...	210	666	14.654...	10.588...	0.0874...	0.0092...	Event
Respo...	group...	210	666	14.654...	10.588...	0.0874...	0.0092...	Event
Modifi...	operat...	212	676	14.794...	10.747...	0.0857...	0.0092...	Event
Service	CS_G...	212	676	14.794...	10.747...	0.0857...	0.0092...	Case
org:re...	operat...	212	676	14.794...	10.747...	0.0857...	0.0092...	Event

Fig. 5. An example of the output of the *Subgroup Discovery* plugin. (Color figure online)

Our plugin provides four types of results. First and foremost, we provide the impact effect analysis that is shown in Fig. 5(a). Each subgroup is shown in one row and its effect on the target group is indicated by a bar. The blue bars indicate a positive influence and red ones depict a negative impacts on the target group. The result presented in Fig. 5(b) illustrates Added Value that suggests how the percentage of classes are changed in a subgroup compared to whole samples. The chart presented in Fig. 5(c) shows how many samples in each subgroup are placed in the target class or another class (red bars correspond to target class). At last, in Fig. 5(d) the plugin shows a table with measured values for coverage, support, and confidence for each subgroup.

5 Evaluation

To evaluate the usefulness of applying subgroup discovery in the field of process mining we applied our approach and implementation to a dataset of Telefonica. As mentioned before, this data relates to the ticket handling process of three services provided by Telefonica and its corresponding process model is shown in Fig. 1. Also, a few statistics for this dataset are shown in Table 5. Guided by our assumption about complete cases, we just consider cases that contain “Canceled” or “Closed” activities. All other cases are removed from dataset.

The business questions that will be answered in the remainder of this section are:

1. Which attribute values often appeared in cases that have a long duration (cases with delays)?
2. Is there any difference in the property values of different services? If yes, what is the difference and which attribute values have more impact on such differences?

To answer each question, we should first define the intended cases that make our target group. Our target group for Question 1 is defined by cases that take more than 80 days to finish. Also, for answering Question 2, we consider the Jasper service (i.e., one of the three provided services) as our target group. Some statistics for these target groups are shown in Table 6.

The results of applying our new ProM plugin on these target groups are shown in Figs. 6 and 7 respectively. In the remainder of this section, we explain some of our findings for each question.

Question 1: Figure 6 indicates 37 subgroups for the class of slow cases. It shows that in the class of slow cases there is an under representation of $Service =$ “CS_M2M” and an under representation of modification by “$Operator\_071$” (in fact there is no case with this service in the slow case class). In contrast, in this class $Responsible\_group =$ “$Group\_016$”, $Modified\_by =$ “$Operator\_172$”, and $Service =$ “$CS - SM2MS$” are more represented and therefore, they have a higher positive effect. Therefore, if stakeholders want to collate with slowing cases they should pay more attention to these properties. For example, they

Table 5. Statistical information of Telefonica dataset

Service name	Case#	Events#	Activities#	Median case duration
All	7,426	146,597	7	12.6 day
SM2MS	5,269	110,536	7	7 day
GSIM	794	12,538	7	37.3 day
Jasper	1,363	23,523	7	22.7 day

Table 6. Statistical information of target groups. For each question, we have two classes.

	Case#	Events#	Activities#	Class%
Slow cases (Q1)	1,433	33,543	7	22.78%
Jasper cases (Q2)	1,251	22,022	7	19.89%
All (filtered)	6,290	125,728	7	

should think about the relation of "*Group*_016" or "*Operator*_172" with the slowness of cases. Also, in this class, the case with $Conformance$ = "*Fitted*" are more presented (for conformance checking we use "Replay a Log on Petri Net for Conformance Analysis" plugin.)

Question 2: In Fig. 7, again 37 subgroups for Jasper service class are illustrated (in our experiments accidentally the number of discovered subgroups be similar). This chart indicates that $Modified\_by$ = "*Operator*_071" and *Assigned_group* = "*Group*_007" have higher influence on this service. Also, the unfitted cases or cases with $Conformance$ = "$unfitted$" are more presented in this target class. Although, some of these subgroups may be obvious (like "*Service* = *CS_SM*2*SM*" has negative impact and *Service* = "*CS_M*2*M*" has positive impact, because we just consider "*CS_M*2*M*" service in this target group), the extracted rules indicate that this approach could extract interesting and correct patterns in subgroups that could not be uncovered by looking at the whole log.

We also present these subgroups to Telefonica experts who have business knowledge. They confirmed that all of the discovered subgroups are correct, but not all of them were considered as surprising. They also recommended us to define other target groups and reapply our approach using these new target groups.

Even though other techniques like correlation, association rule mining and decision tree have similarities with subgroup discovery algorithm, they could not find these discovered subgroups. For example, when we apply correlations we typically do not consider the sizes of subgroups. Also, when applying decision trees, we aim to discover rules for predicting future samples not describing current ones. In association rule mining variations that consider a class feature, the focus is on *coverage*, *support* and *confidence* of a class and an item set and

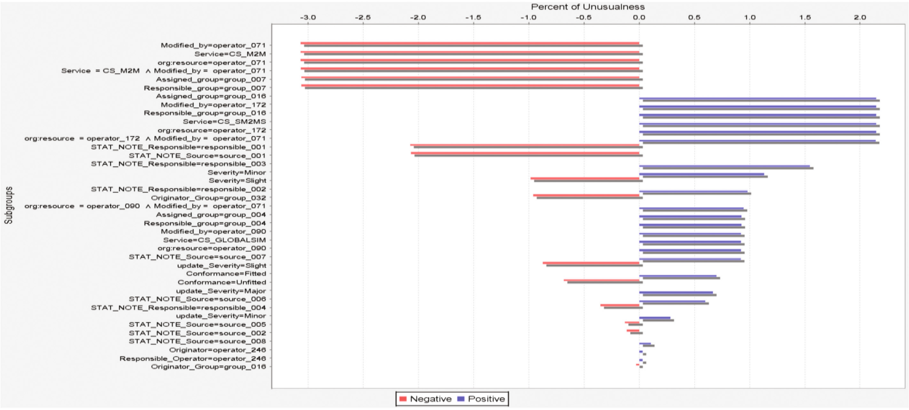


Fig. 6. Interesting patterns discovered by subgroup discovery technique for cases with long duration. The red bars indicate negative effect and coloration and blue bars suggest positive influence and correlation. (Color figure online)

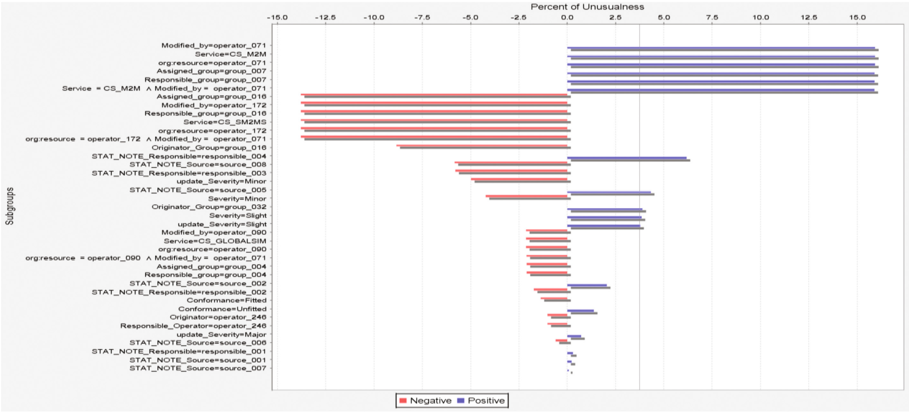


Fig. 7. Interesting patterns for Jasper service cases. Longer bars show higher influence for corresponding subgroup.

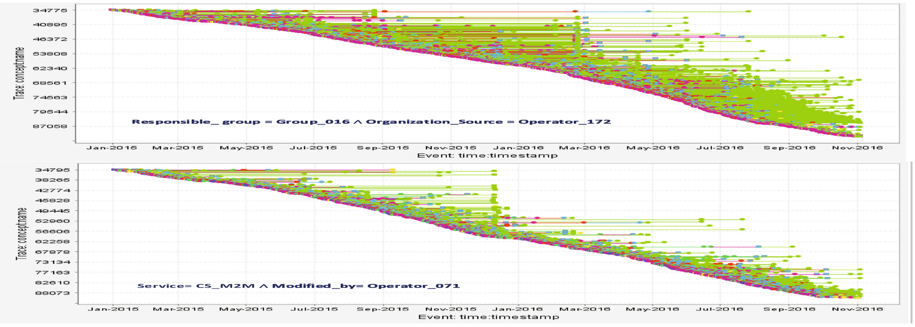


Fig. 8. The Dotted charts of two sublogs extracted from two discovered subgroups.

unusualness of a rule is of less relevant. In the other hand in subgroup discovery, at the same time *coverage* and *unusualness* of a pattern (not a rule) are intended to describe the model of samples. Also, subgroup discovery is focused on the target group rather than all the samples.

According to Fig. 3, each subgroup also is an event log that could be used for further process mining analysis. In Fig. 8, we compare the dotted charts of two sublogs related to subgroups of ($Responsible\_group$ = "$Group\_016$" $\wedge$ $Operator$ = "$Operator\_172$" $\rightarrow$ $Target = SlowCases$) and ($Service$ = "CS_M2M" $\wedge$ $Modified\_by$ = "$Operator\_071$" $\rightarrow$ $Target = SlowCases$). According to Fig. 8, it is indicated that the cases of the first subgroup take more time whereas cases in the second subgroup take less time. We also apply *"Mine Petri Net with Inductive Miner"* plugin on these sublogs. The discovered models using this plugin are shown in Fig. 9. According to this figure, there are difference in their process. For example, in the process model in Fig. 9(a) it is possible for a "Delayed" ticket to be "Active" again, but it is impossible in the in the process model in Fig. 9(b). These kinds of analysis could give valuable information to stakeholders for understanding the reasons for difference in behavior in subgroups of cases. It is also possible to apply any other process mining technique on the discovered subgroups.

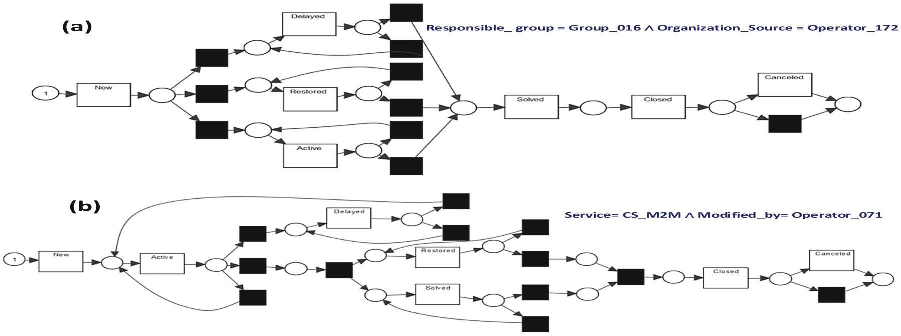

(a)
Responsible_ group = Group_016 ∧ Organization_Source = Operator_172
1
New
Delayed
Restored
Active
Solved
Closed
Canceled
(b)
Service= CS_M2M ∧ Modified_by= Operator_071
1
New
Active
Delayed
Restored
Solved
Closed
Canceled

Fig. 9. The process models of two sublogs discovered by *"Mine Petri Net with Inductive Miner"* plugin: (a) is the process model of subgroup ($Responsible\_group$ = "$Group\_016$" $\wedge$ $Operator$ = "$Operator\_172$" $\rightarrow$ $Target = SlowCases$) and (b) is the process model of ($Service$ = "CS_M2M" $\wedge$ $Modified\_by$ = "$Operator\_071$" $\rightarrow$ $Target = SlowCases$)

6 Conclusion

Process mining can be used to extract knowledge from event logs. However, event logs may contain information on cases with very different characteristics.

Analyzing these different group of cases together may conceal important phenomena. Delays and deviations may be linked to very particular subgroups that are not known beforehand.

To address this problem, we applied subgroup discovery technique to find the statistically interesting patterns in subsets of cases belonging to a predefined target class (e.g., cases that are delayed). In this regard, properties of the event log are extracted with their corresponding values. These properties could be related to the case, its events or computed by other process mining techniques. Afterwards, interesting subgroups of the target group can be extracted by applying well-known measures like *Added Value* and *WRAcc*. Interesting subgroups that contribute to the target group positively or negatively may be discovered. Importantly, any process mining algorithms can be applied to the discovered subgroups to extract surprising insights and behaviors.

To evaluate the proposed approach we developed a plugin in a *ProM* platform and applied it in a case study conducted together with Telefonica. Two target groups are defined for this purpose, one for slow cases and another for cases related to a specific service. This case study indicates that the proposed approach could is able to discover interesting patterns. However, not all of them were surprising for business experts.

In the current implementation we do not consider attributes with continues values. In *ProM* and other data mining tools there are techniques to make these attributes discrete. Not doing this up-front, but trying to integrate this in the approach itself may be very time consuming, especially for time and date attributes. Here, we also define target groups manually, however defining a suitable target group would be a challenging task for continues attributes.

References

1. van der Aalst, W.M.P.: Process Mining: Data Science in Action. Springer, Heidelberg (2016)
2. Van der Aalst, W.M.P.: Using process mining to bridge the gap between BI and BPM. IEEE Comput. **44**(12), 77–80 (2011)
3. Herrera, F., Carmona, C.J., González, P., Del Jesus, M.J.: An overview on subgroup discovery: foundations and applications. Knowl. Inf. Syst. **29**(3), 495–525 (2011)
4. Bose, R.P.J.C., van der Aalst, W.M.P.: Context aware trace clustering: towards improving process mining results. In: Proceedings of the 2009 SIAM International Conference on Data Mining, pp. 401–412. SIAM (2009)
5. Novak, P.K., Lavrač, N., Webb, G.I.: Supervised descriptive rule discovery: a unifying survey of contrast set, emerging pattern and subgroup mining. J. Mach. Learn. Res. **10**, 377–403 (2009)
6. Verbeek, H.M.W., Buijs, J.C.A.M., Dongen, B.F., Aalst, W.M.P.: XES, XESame, and ProM 6. In: Soffer, P., Proper, E. (eds.) CAiSE Forum 2010. LNBIP, vol. 72, pp. 60–75. Springer, Heidelberg (2011). doi:10.1007/978-3-642-17722-4_5
7. Klösgen, W.: Explora: a multipattern and multistrategy discovery assistant. In: Advances in Knowledge Discovery and Data Mining, pp. 249–271. American Association for Artificial Intelligence (1996)

8. Wrobel, S.: An algorithm for multi-relational discovery of subgroups. In: Komorowski, J., Zytkow, J. (eds.) PKDD 1997. LNCS, vol. 1263, pp. 78–87. Springer, Heidelberg (1997). doi:10.1007/3-540-63223-9_108
9. Atzmueller, M.: Subgroup discovery. Wiley Interdisc. Rev. Data Min. Knowl. Discov. **5**(1), 35–49 (2015)
10. Kateri, M.: Contingency Table Analysis. Springer, Heidelberg (2014)
11. Herrera, F., et al.: An overview on subgroup discovery: foundations and applications. Knowl. Inf. Syst. **29**(3), 495–525 (2011)
12. Duivesteijn, W., et al.: Subgroup discovery meets Bayesian networks-an exceptional model mining approach. 2010 IEEE 10th International Conference on Data Mining (ICDM). IEEE (2010)
13. Atzmueller, M., Baumeister, J., Puppe, F.: Introspective subgroup analysis for interactive knowledge refinement. In: FLAIRS Conference, pp. 402–407 (2006)
14. Piatetsky-Shapiro, G.: Discovery, analysis, and presentation of strong rules. Knowl. Disc. Databases, 229–238 (1991). https://www.bibsonomy.org/bibtex/26fa5f6987b667b728c7e94f7c68b52d7/enitsirhc
15. Huynh, X.-H.: Interestingness Measures for Association Rules in a KDD Process: Postprocessing of Rules with ARQAT Tool. Université de Nantes (2006)
16. Song, M., van der Aalst, W.M.P.: Supporting process mining by showing events at a glance. In: Proceedings of the 17th Annual Workshop on Information Technologies and Systems (WITS), pp. 139–145 (2007)
17. Bolt, A., Leoni, M., Aalst, W.M.P.: A visual approach to spot statistically-significant differences in event logs based on process metrics. In: Nurcan, S., Soffer, P., Bajec, M., Eder, J. (eds.) CAiSE 2016. LNCS, vol. 9694, pp. 151–166. Springer, Cham (2016). doi:10.1007/978-3-319-39696-5_10
18. Leemans, S.J.J., Fahland, D., van der Aalst, W.M.P.: Process and deviation exploration with inductive visual miner. In: BPM (Demos), p. 46 (2014)
19. Van der Aalst, W., Adriansyah, A., van Dongen, B.: Replaying history on process models for conformance checking and performance analysis. Wiley Interdisc. Rev. Data Min. Knowl. Disc. **2**(2), 182–192 (2012)
20. Verbeek, H.M.W., Buijs, J., Van Dongen, B.F., van der Aalst, W.M.P.: Prom 6: the process mining toolkit. Proc. BPM Demonstration Track **615**, 34–39 (2010)
21. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Proceedings of 20th International Conference Very Large Databases, VLDB, vol. 1215, pp. 487–499 (1994)