Mining multi-dimensional complex log data

Stephen Pauwels STEPHEN.PAUWELS@UANTWERPEN.BE
University of Antwerp, Middelheimlaan 1, 2020, Antwerp, Belgium

Toon Calders TOON.CALDERS@UANTWERPEN.BE
University of Antwerp, Middelheimlaan 1, 2020, Antwerp, Belgium

Keywords: process mining, preprocessing, multi-dimensional data, clustering

Abstract

In the field of process mining, many techniques have been developed to automatically derive process models from event logs. These algorithms, however, require that in the log files events are labeled with the activity that was executed, and that they already have been grouped into cases (instances of the business process). Unfortunately, this requirement is often highly non-trivial to realize for real-life log files. Therefore in this paper we propose a two step approach based on clustering to perform the preprocessing of the log file automatically. Experiments on a real-life log show the usabilty of the method.

1. Process Mining

Figure 1 gives an example of how a log file looks like. All entries in the log are called events, they consist of different attributes such as who executed the job, at what time, what is the type of the job, etc. We will use these attributes to calculate the similarity between two events in order to cluster them together. Events with timestamp 1,2 and 3 belong to the same workflow and are bundled together into a case {1,2,3}. Case {5,6,8} and {1,2,3} belong to the same business process. Events 4 and 7 form a case that belongs to a different process.

There exist already techniques that combine process mining with clustering (Song et al., 2008; Bose & van der Aalst, 2009; De Medeiros et al., 2007). The idea here is to cluster different cases of a single process in order to get better and more general results. The difference with our technique is that these techniques still require some extra knowledge about the logfile, the cases are defined and it is clear to which process each case belongs. Our approach starts with a log file where we only have timing information, we do not use any information about cases or processes present.

t	Activity	Supervisor	User
1	Send vacation request	Toon	Stephen
2	Mail sent to supervisor	Toon	Stephen
3	Confirmation sent to user	Toon	Stephen
4	Request status	Wim	Paul
5	Send vacation request	Chris	Stephen
6	Mail sent to supervisor	Chris	Stephen
7	Receive status	Wim	Paul
8	Confirmation sent to user	Chris	Stephen

Figure 1. Example of a log file where every event has a timestamp and 3 attributes

2. Process Finding

2.1. Forming Cases by Clustering Events

In order to be able to cluster the events we have to exploit the next observation: usually, within a single case, certain attributes are the same between the different events. We can use existing datamining techniques to automatically find the different cases and the attribute(s) that identify a particular process.

We start by building an Event Similarity Graph (ESG). This is a weighted graph where every node stands for an event present in the logfile. We will use the relative frequency of a value within an attribute as the basis for our similarity measure. We do this because events are more likely to belong together when they share a rare value than when sharing a common value. The weight of an edge between two events

Appearing in *Proceedings of Benelearn 2016*. Copyright 2016 by the author(s)/owner(s).

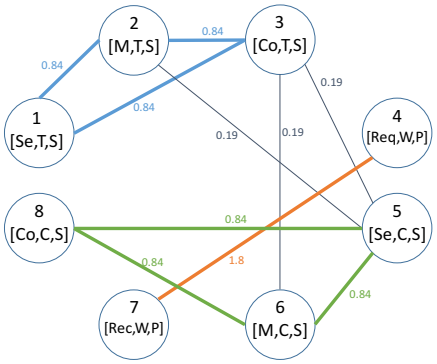


Figure 2. Example of an Event Similarity Graph with weights and best cliques highlighted in color.

(nodes) is determined by the Goodall3 measure found in Boriah et al. (2008).

To reduce the complexity of this costly operation, we only consider events that are within a limited time-window. In our example in Figure 2 the window is set to 3, therefor we only try to link node 1 with nodes 2,3 and 4. We add an edge between two nodes when the weight reaches a minimum threshold, this will reduce the size and complexity of our graph even further. In our example the edges with weight 0.19 will not be added to the graph when setting a threshold of 0.50. Note that these two reductions do not influence the results of our graph as most events from a process do occur within a particular time-window.

In order to cluster events into cases we search for maximal cliques (Cheng et al., 2010) that occur in our ESG. We determine the best clique by calculating the average similarity between two nodes in the clique. The clique with highest score in our example is the clique {4,7} (1.8). We first extract this clique and remove all nodes in the clique from the graph. Next we extract the clique with the second highest score, and so on. We repeat this step until all cliques are found or all nodes have been used. Possible cliques found in the graph that do not represent a case are eliminated using this way of extracting cliques (such as clique {3,5,6}).

2.2. Forming Processes by Clustering Cases

Now that we have grouped events into cases, we want to combine these cases on their turn into the different business processes to which they belong. When clustering the cases together we can no longer rely on the similarity of the data attributes per event. Indeed, within one case it is likely that the activities are executed by the same users, using the same resources, and within a bounded time. Between different cases of the same process, however, many of the data attributes will change. But the structure of which attributes remain the same will be present in all cases belonging to the same process. As can be seen between cases {1,2,3} and {5,6,8} from our example logfile. Therefor we will use a similarity measure that characterizes similarity between sequences of events. We will compare cases by calculating the length of the *longest common subsequence (lcs)*. In order to use this measure we first *flatten* our cases, as demonstrated in Figure 3. That is, every event in a case is represented as the sequence of its values, and the case is the concatenation of these per-event sequences. We will also add $-i$ to the flattened sequence when attribute i has the same value as in the previous event, these additions will capture the structure of the different sequences.

Case 1	Se C S M C -2 S -3 Co C -2 S -3
Case 2	Se T S M T -2 S -3 Co T -2 S -3
Case 3	Req W P Rec W -2 P -3

Figure 3. Flattened cases

3. Results

We evaluate the different steps of our algorithm separately using two real-life logfiles, one private (*log1*) and one public (*log2*) (Buijs, 2014). We looked at the recall and precision for pairs of events that belong together, these results can be found in table 1 as well as the specifications for the two datasets.

Table 1. Specifications and results for the datasets

	#Events	#Cases	Precision	Recall
log1	10000	4194	3.26%	51.89%
log2	8577	1434	91.27%	95.62%

To test the second step we start with all cases perfectly clustered together. We do this to avoid possible negative influences of the first step. We look at how much of the events are clustered together with events that belong to the same process. We see that we can achieve a rate of 98% of events clustered together correctly. These results give a clear indication that preprocessing can be done automatically, for certain types of log files with a certain structure, and so minimizing the need for users to interact with the system. As this are only our first results of ongoing work we are confident that certain parts of our algorithm can still be improved. Just as we would like to add more live user feedback to improve the results.

References

Boriah, S., Chandola, V., & Kumar, V. (2008). Similarity measures for categorical data: A comparative evaluation. *red*, *30*, 3.

Bose, R. J. C., & van der Aalst, W. M. (2009). Context aware trace clustering: Towards improving process mining results. *SDM* (pp. 401–412).

Buijs, J. (2014). Receipt phase of an environmental permit application process (wabo), coselog project.

Cheng, J., Ke, Y., Fu, A. W.-C., Yu, J. X., & Zhu, L. (2010). Finding maximal cliques in massive networks by h*-graph. *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data* (pp. 447–458).

De Medeiros, A. K. A., Guzzo, A., Greco, G., Van Der Aalst, W. M., Weijters, A., Van Dongen, B. F., & Saccà, D. (2007). Process mining based on clustering: A quest for precision. *Business Process Management Workshops* (pp. 17–29).

Song, M., Günther, C. W., & Van der Aalst, W. M. (2008). Trace clustering in process mining. *Business Process Management Workshops* (pp. 109–120).