ACD2: a tool to interactively explore Business Process Logs

Stephen Pauwels and Toon Calders

University of Antwerp, Antwerp, Belgium
{stephen.pauwels,toon.calders}@uantwerpen.be

Abstract. ACD2 is a tool for detecting anomalies and concept drifts in Business process logs. In contrast to many other existing algorithms, ACD2 does not require any manual parameter to be set. ACD2 is based on Extended Dynamic Bayesian Networks. These models are constructed automatically using machine learning, but can be revised by the user afterwards. This model can then be used for scoring cases. Our tool visually represents these scores, making it easy for a user to investigate the data.

1 Introduction

ACD2 is a fully interactive, easy-to-use tool based on our winning submission for the BPI 2018 Challenge [6]. Thanks to the positive feedback we have further elaborated the ideas presented in the report and incorporated them in this tool. The tool provides an intuitive way to test for anomalies or concept drift in log files. The tool aims at both Business Process experts and domain specific experts with no knowledge about the underlying algorithms.

An Extended Dynamic Bayesian Network (EDBN) [7] captures the different relations between attributes in a log file. Using the model, we are able to learn the normal behavior found in a log file. An EDBN represents a joint probability distribution and can therefore be used for scoring new events and cases (the case score is the accumulation of event scores), the score indicates how much a case is compliant with a reference log file. An important property of this score is its decomposability, we get the score contribution of individual attributes. In the remainder of this paper we refer to this score when talking about the score of an attribute, event or case.

The main workflow of the tool is to first use a log to learn the EDBN model structure. The user can then inspect the learned model and make changes to the structure. When the user is happy with the current model, she can use it to test a log for deviations or anomalies given a reference log containing normal behavior.

Performing both the training and testing phases can be a computationally heavy task. Therefor we have created a client-server based application, where all heavy computations happen on the server. Because the tool is fully web-based it can also be used on tablets and smartphones, increasing its usability for non-experts.

2 Functionalities

ACD2 consist out of the following main parts:

1. Loading datasets
2. Learning the model of the EDBN
3. Inspecting and updating the Model
4. Testing a dataset
5. Anomaly Detection

2.1 Loading the Data

A first step is to upload the datasets we want to use for both learning the model and testing. After uploading, the user can select which attributes she wants to include in the dataset and indicate which attribute corresponds to the case identifier, the time attribute and a (optional) label, indicating if the event is anomalous or not. After uploading, a background task starts to preprocess and store the data on the server for easy access later. The card on the dataset page indicates when preprocessing is done.

2.2 Learning the model

In this phase all relations between attributes that are present in a given dataset are learned. We have chosen to use an asynchronous process because learning the model is computationally demanding and can take some time. When learned, the model gets saved on the server and can be updated by the user or used for testing.

2.3 Inspecting and updating the Model

Once a model is fully learned it can be inspected by the user. She can make changes to her own insights in the data. Every time the user makes a change the difference in quality between the updated model and the original model is calculated and displayed. We use the Akaike Information Criterion [1] for determining the quality of a model. This metric takes the log-likelihood of the reference data given the model and the complexity of the model into account.

2.4 Testing a dataset

Now we can test a new log against the learned model. At this point the model only contains the structure of the EDBN, therefor we first need to further train the different parameters in the model. We thus select a training dataset and a test dataset. An important constraint on these datasets is that they all need to have the same attributes as the original dataset used for learning the structure in order to be compatible. After submitting, the model is further trained and

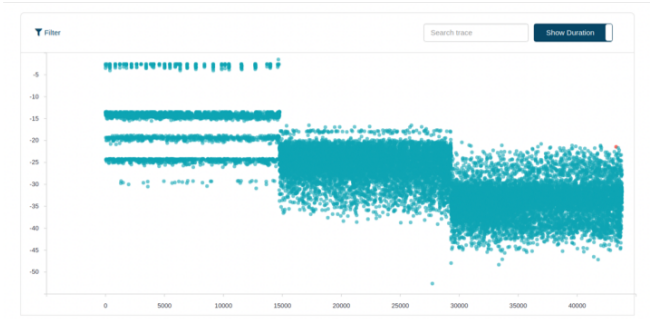

Fig. 1. Case graph with time (or ID) on the x-axis and the score on the y-axis.

the scores are calculated for the test dataset. When all results are computed a user can start analyzing the results.

The result page consists of three main parts. The middle graph plots all cases in the test log according to their timestamp (or ID) and score, an example graph is shown in Figure 1. This graph forms the basis of any analysis. It can either be used to analyze global behavior, finding clusters of cases (cases with similar properties or behavior), or to inspect individual cases. A low score for a case (or event) means that they do deviate more from the normal behavior, learned from the training dataset. When clicking on a single case, a new table appears below the case graph showing all events in that particular case. Events that deviate more than one standard deviation from the mean score of events are highlighted in red to indicate a possible inconsistency.

Above the case graph we show the attribute plot. For every attribute in the data it shows the mean value of the score for this attribute in the test set. And again, the lower the score, the more the scores for this attribute deviate from the training dataset. In order to compare a subset of the data to the entire dataset we can set a filter for creating a comparison dataset. Using this we can easily see in which attribute(s) a drift occurs.

Drifts in the data can be detected by looking to the case scores. After a drift the distribution of these scores changes. Sometimes these changes can be seen on the graph itself. In order to determine the drift point(s) in more detail, we use

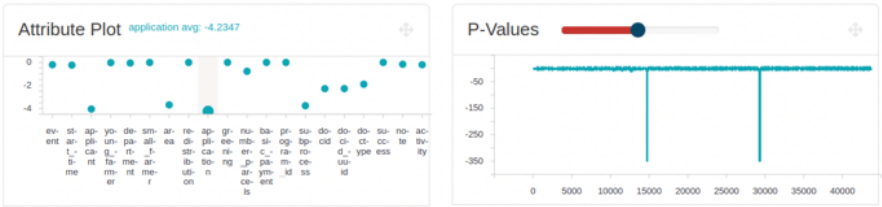

Fig. 2. The Attribute plot and P-values used for explaining and detecting drifts.

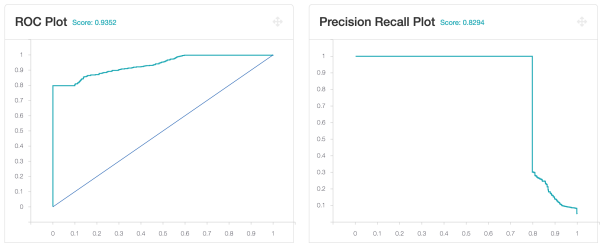

Fig. 3. The ROC and PR curve as shown in the tool.

the P-values plot, which is created using the Kolmogorov-Smirnov test (KS-test) [4]. The KS-test checks if two distributions are similar or different. We compare a number of case scores before a possible drift point to a number of scores after the drift point. The lower the p-value the more different the distributions are. Examples of these graphs can be found in Figure 2.

2.5 Anomaly detection

Besides concept drift detection, ACD2 can also be used for detecting anomalies in the test dataset. The lower a case in the case graph, the more likely it is to be an anomaly. We can also use the tool with labeled data in order to examine the quality of a certain anomaly detection method. When a user selected a label attribute, two extra graphs are added to the analysis page, namely the Precision-Recall curve (PR-curve) and the Receiver Operating Characteristic curve (ROC-curve) [2], which both give a indication of the quality of the scores given to the cases. Example PR and ROC-curves can be seen in Figure 3.

3 Related Work

The ProM tool[1] contains different conformance checking and concept drift detection algorithms, often based on computing alignments. Almost all available algorithms only take the control-flow (and sometimes resource) perspective into account [3]. Our method overcomes this shortcoming by allowing any number of extra attributes to be used. The ProM tool, however powerful and extensive, is less suitable for non-experts in the Business Process domain as these methods often require lots of parameters to be set and a fair understanding of the algorithms used to get the best results out of it. ACD2 tries to hide away all these details for the user, making the tool more easy and intuitive to use.

BINet [5] is another method for detecting anomalies in Business Process logs capable of dealing with multiple attributes. Since it is based on Neural Networks they do not offer the possibility for a user to intervene during the process of learning the model.

[1] http://www.promtools.org

4 Conclusion

We showed our fully functional Anomaly and Concept Drift detection tool ACD2 that is based on work done for the BPI 2018 Challenge. The main goal for this tool is to allow for visual analysis of Business Process log files. One of the ideas therefore is that people with only limited knowledge about Business Processes and the underlying algorithms should be able to interact with the tool and can use it to determine and explain Concept Drift or Anomalies in a given dataset. This is one of the main reasons for making it a web-based application with very easy wizard-like steps to perform.

Our method is based on the Extended Dynamic Bayesian Networks, making it a very expressive and intuitive way of capturing the normal behavior found in log files. Thanks to this our tool also allows for user intervention after the model has been learned. This in contrast to most other techniques that do not allow for intermediate user input.

We used the data from the BPI 2018 Challenge for the illustrative examples. This data consists of 2M+ events and 43,809 cases, indicating that our tool is able to handle larger datasets, thanks to our client-server architecture. Currently we are working hard to further improve and optimize loading of the result pages.

In the future we would like to extend the comparison capabilities of our tool. Where a user could, for example, draw a bounding box in the case graph to select a particular subset she wants to examine in more detail. We would also like to allow more different types of input data and further improve the preprocessing capabilities of the tool.

A link to the tool, screencast and fully elaborated use case are available at http://adrem.uantwerpen.be/conceptdrift.

References

1. Akaike, H.: A new look at the statistical model identification. IEEE transactions on automatic control **19**(6), 716–723 (1974)
2. Davis, J., Goadrich, M.: The relationship between precision-recall and roc curves. In: Proceedings of the 23rd international conference on Machine learning. pp. 233–240. ACM (2006)
3. Dunzer, S., Stierle, M., Matzner, M., Baier, S.: Conformance checking: a state-of-the-art literature review. In: Proceedings of the 11th International Conference on Subject-Oriented Business Process Management. p. 4. ACM (2019)
4. Massey Jr, F.J.: The kolmogorov-smirnov test for goodness of fit. Journal of the American statistical Association **46**(253), 68–78 (1951)
5. Nolle, T., Seeliger, A., Mühlhäuser, M.: Binet: Multivariate business process anomaly detection using deep learning. In: International Conference on Business Process Management. pp. 271–287. Springer (2018)
6. Pauwels, S., Calders, T.: Detecting and explaining drifts in yearly grant applications. arXiv preprint arXiv:1809.05650 (2018)
7. Pauwels, S., Calders, T.: An anomaly detection technique for business processes based on extended dynamic bayesian networks. In: Proceedings of the 2019 ACM Symposium on Applied Computing (2019)