CDLG: A Tool for the Generation of Event Logs with Concept Drifts

Justus Grimm, Alexander Kraus and Han van der Aa

Data and Web Science Group, University of Mannheim, Germany

Abstract

Process mining targets the analysis of data recorded during the execution of business processes. When such data covers a period in which a process underwent changes, an event log is subject to concept drifts. Given that such drifts can greatly affect the quality of insights obtained from them, numerous approaches have been established to detect concept drifts. However, assessing and comparing the quality of these approaches is hard, due to the lack of appropriate data. Recognizing this, we present CDLG; a powerful and highly-flexible tool for the generation of event logs with known concept drifts.

Keywords

Process mining, Concept drifts, Event log generation

1. Introduction

Business processes are subject to frequent changes due to the dynamic environments in which they are executed [1]. Analyzing data recorded during the execution of such changing processes results in the presence of concept drifts, i.e., situations in which an event log contains data from different versions of a process. Recognizing the detrimental impact that such concept drifts can have on obtained process mining results, a broad range of detection approaches has been developed [2].

To assess and compare the quality of such approaches, appropriate evaluation data is required in the form of event logs that are subject to concept drifts and for which the details of these drifts are available as a *gold standard*. This is hardly the case for publicly available real-world event logs, for which there are only a few instances of (partially) known drifts, such as the *help desk* log [3] that was studied by various authors [4, 5]. When it comes to synthetic event logs, there are no reference collections available, whereas existing generation tools provide only minimal support for the generation of event logs that contain drifts [6] and do not provide information in terms of a gold standard.

To address this gap, this paper presents the Concept Drift Log Generator (CDLG), a tool that allows users to generate synthetic event logs with concept drifts and a corresponding gold standard. CDLG supports a wide range of options when it comes to log generation, allowing users to obtain logs of varying complexity (in terms of trace length, control-flow constructs, log size, noise, etc.) and insert one or more concept drifts of different types (i.e., sudden, gradual,

The BPM 2022 Demos & Resources Forum, Sept. 11-16, 2022, Münster, Germany

jugrimm@mail.uni-mannheim.de (J. Grimm); alexander.kraus@uni-mannheim.de (A. Kraus); han.van.der.aa@uni-mannheim.de (H. van der Aa)

recurrent, and incremental) into them. CLDG provides users with the means to obtain these logs through three execution modes, trading off customization versus speed. In particular, users can generate logs by following an interactive terminal script, by providing parameters through a text file, or by generating an entire log collection based on desired high-level settings.

In the remainder of this paper, Section 2 describes the functionality of the CDLG tool in more detail, whereas Section 3 discusses its maturity. The tool itself, a corresponding Python package, a tutorial document, and an instruction video can be accessed through our repository.[1]

2. The CDLG Tool

This section describes the CDLG tool in terms of its scope (Section 2.1) and the three execution modes that it provides to users (Section 2.2).

2.1. Scope

Our CDLG tool allows users to generate event logs that contain the primary kinds of control-flow-based concept drifts and provides users with considerable control over other aspects, such as model complexity, log size, and noise. As discussed in Section 2.2, users can customize all these aspects if desired or instead choose to go with default or automatically generated options.

Concept drift types. CDLG can generate event logs that contain drift scenarios with the four drift types visualized in Figure 1, i.e., sudden, incremental, gradual, and recurring drifts, as, e.g., defined by Bose et al. [1].

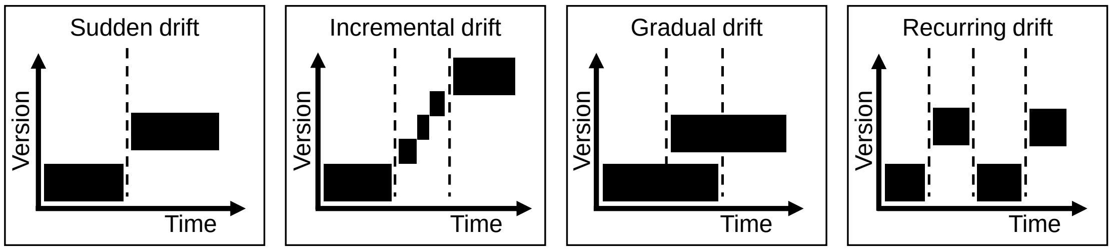


Figure 1: Concept drift types covered by CDLG. Adapted from [1].

Event log generation. CDLG generates event logs with concept drifts by playing out a *process tree* for each version of a process required for a given drift scenario. As seen in Figure 1, this means that the generation of a single sudden, gradual, or recurring drift requires two process trees, whereas an incremental drift requires three or more different trees. Depending on the employed execution mode (see Section 2.2), the necessary process trees can be explicitly provided as input by the user or automatically generated.

Since sudden, incremental, and recurring drifts involve the non-overlapping execution of process versions, CDLG generates sub-logs for each part of the drift scenario, according to the corresponding process tree, and then concatenates these sub-logs. For instance, a sudden drift

[1]https://gitlab.uni-mannheim.de/processanalytics/cdlg_tool

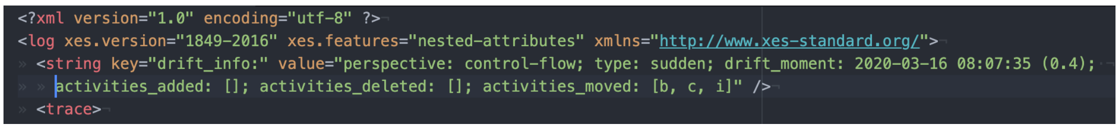
from version v_1 to v_2 that occurs after three quarters of a desired event log with 1000 traces will consist of 750 traces of v_1, followed by 250 traces of v_2. For incremental and recurring drifts, the tool generates as many sub-logs as necessary to capture the drift scenario (e.g., five for the incremental and four for the recurring drift in Figure 1).

For gradual drifts, which contain a period of time in which two process versions overlap, we generate sub-logs such that their joint version has a period with traces from both v_1 and v_2. Our tool allows for this transition to occur in a linear fashion, where traces of v_2 gradually become more common, as well as in an exponential fashion, in which this transition occurs more rapidly.

Multi-drift scenarios. CDLG also supports the generation of event logs with multiple, consecutive drifts. In this case, multiple of the aforementioned drift scenarios are concatenated, where the last process version of one scenario represents the first version used in the next.

Noise insertion. Finally, CDLG supports the insertion of noise into an event log, which can be used to assess the robustness of concept drift detection approaches. We allow for noisy traces to be inserted that are similar to the traces in the provided process trees, as well as for traces that are wholly randomly generated (over the same set of activities as the process trees).

Gold standard attribute. Having information on the characteristics of the concept drift(s) in an event log is crucial for the proper evaluation of detection approaches. Therefore, given an inserted drift, CDLG records information on its drift type, the moment at which it occurred, as well as the control-flow changes that occurred in the process. This is all stored as an event log attribute in the resulting XES file, as shown in Figure 2.



```
<?xml version="1.0" encoding="utf-8" ?>
<log xes.version="1849-2016" xes.features="nested-attributes" xmlns="http://www.xes-standard.org/">
  <string key="drift_info:" value="perspective: control-flow; type: sudden; drift_moment: 2020-03-16 08:07:35 (0.4);
    activities_added: []; activities_deleted: []; activities_moved: [b, c, i]" />
  <trace>
```

Figure 2: Event log attribute capturing the gold standard of a generated drift.

2.2. Execution Modes

CDLG allows users to generate event logs in various manners, generally trading off ease of use (and speed) versus the degree of customization that is possible.

Terminal mode. This mode, accessed through `start_generator_terminal.py`, provides the highest degree of customization to a user. As detailed in the tutorial document (see Section 1), this mode starts by allowing users to define the process trees necessary for a drift scenario (either provided as input or automatically generated according to desired characteristics), and define aspects such as the desired log size. Then, users can define the desired drift scenario between the established process versions in terms of the drift type and change moment(s). Finally, users can decide to insert noise and, if desired, add another drift to the log. Note that the terminal mode provides default settings for all options, which means that users have considerable freedom when establishing drift scenarios, but can skip customization of certain aspects if desired.

Parameter-file modes. For users that simply want to obtain event logs with concept drifts to evaluate a detection approach, CDLG provides two options. Users

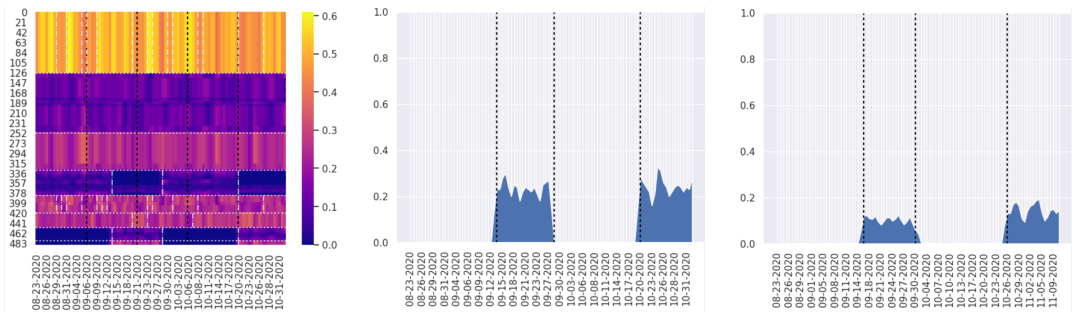

(a) Drift map. (b) Drift chart w/o noise. (c) Drift chart with noise.

Figure 3: VDD visualizations for recurring drift (from [7]).

can generate individual logs according to parameters set in a text file by calling the `generate_log_drift_type_from_doc.py` functions (one for each type of drift). Due to the randomization provided by CDLG, such as the automated generation of process trees of various complexity, each execution of these methods will result in a unique event log.

Users can also execute `generate_collection_of_logs.py` to generate an entire collection of a user-defined number of event logs with concept drifts. This method provides slightly less customization options as the methods for individual logs, but can provide users with a large number of event logs suitable for the automated assessment of the accuracy of a concept drift detection approach.

Python package. Finally, we provide the core functionality of CDLG as part of a Python package, which can be installed using *pip*.[2] Specifically, this package provides methods for the generation of event logs with each of the four kinds of concept drifts, the insertion of noise in such event logs, and the automated generation of process trees to be used as a basis for log generation. For more details on the methods and their parameters, we refer to the corresponding *README* file.

3. Tool Maturity

The core of CDLG was developed as a bachelor project, of which the resulting thesis [7]—available in the project's repository—provides details on the conceptual algorithms underlying CDLG's functionality.

Evaluation. As part of this thesis project, CDLG was evaluated by using it to generate various event logs with concept drifts and feeding these event logs as input to the Visual Drift Detection (VDD) tool [5], a state-of-the-art concept drift detection approach and visualizer. The evaluation showed the suitability of the CDLG logs to assess the potential of detection approaches to recognize concept drifts in different scenarios and under different circumstances, for instance by comparing event logs with and without noise. A visualization obtained using VDD for such a case is shown in Figure 3. The figures reveal that VDD was able to accurately detect a recurring

[2]https://gitlab.uni-mannheim.de/processanalytics/cdlg-package

drift in event logs without and with noise, to the significance of the results in the latter case was lower (Figure 3c).

Future work. We plan to continue the development of CDLG in the future.

Technical extensions of the current functionality that we aim to make are to enable CDLG to automatically establish intermediary steps for incremental drifts between two provided process trees and to provide users with more control over the noise-insertion procedure, e.g., by ensuring that noise inserted into processes with meaningful event labels makes semantic sense [8].

On a conceptual level, a key extension that we envision includes support for multi-perspective drifts, e.g., where the behavior of a process changes with respect to factors such as the execution time or arrival rate, distributions over data attributes, the resource perspective, and combinations of such aspects together with control-flow changes. Furthermore, we aim to provide support for the generation of multi-order drifts, which are drift scenarios in which multiple drifts occur simultaneously, such as a process that is subject to both seasonal and monthly recurring changes. Finally, we also intend to extend CDLG by providing methods for the automated computation of evaluation results of a detection approach for a set of CDLG generated event logs, providing measures such as precision, recall, and detection delay.

References

[1] R. J. C. Bose, W. M. Van Der Aalst, I. Žliobaitė, M. Pechenizkiy, Dealing with concept drifts in process mining, IEEE Trans Neural Netw Learn Syst 25 (2013) 154–171.

[2] G. Elkhawaga, M. Abuelkheir, S. I. Barakat, A. M. Riad, M. Reichert, Conda-pm—a systematic review and framework for concept drift analysis in process mining, Algorithms 13 (2020) 161.

[3] M. Polato, Dataset belonging to the help desk log of an italian company, 2017. doi:10.4121/uuid:0c60edf1-6f83-4e75-9367-4c63b3e9d5bb.

[4] A. Ostovar, A. Maaradji, M. La Rosa, A. H. ter Hofstede, B. F. van Dongen, Detecting drift from event streams of unpredictable business processes, in: ER, Springer, 2016, pp. 330–346.

[5] A. Yeshchenko, C. Di Ciccio, J. Mendling, A. Polyvyanyy, Comprehensive process drift detection with visual analytics, in: ER, Springer, 2019, pp. 119–135.

[6] A. Burattin, PLG2: Multiperspective process randomization with online and offline simulations, in: CEUR Workshop Proceedings, volume 1789, 2016.

[7] J. Grimm, CDLG: Generation of event logs with concept drifts, Bachelor's Thesis, University of Mannheim, 2022.

[8] H. van der Aa, A. Rebmann, H. Leopold, Natural language-based detection of semantic execution anomalies in event logs, Information Systems 102 (2021) 101824.