Hi Gareth,
Submitted 2 days late. Please lookup our email conversation.
Regards,
Sujith

Sujith Galla
18214293
Sujith.galla3@mail.dcu.ie
CA640_MCM1
30/11/2018
Process Mining Event logs to map Customer Journeys.

An report submitted to Dublin City University, School of Computing for module CA640 Professional and Research Practice , 2017/2018.
I understand that the University regards breaches of academic integrity and plagiarism as grave and serious.

I have read and understood the DCU Academic Integrity and Plagiarism Policy. I accept the penalties that may be imposed should I engage in practice or practices that breach this policy.

I have identified and included the source of all facts, ideas, opinions, viewpoints of others in the assignment references. Direct quotations, paraphrasing, discussion of ideas from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged and the sources cited are identified in the assignment references.

I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

By signing this form or by submitting this material online I confirm that this assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study. By signing this form or by submitting material for assessment online I confirm that I have read and understood DCU Academic Integrity and Plagiarism Policy (available at: http://www.dcu.ie/registry/examinations/index.shtml)

Name(s):Sujith Galla
Date: 30/11/2018

Introduction

The goal of the master's project would be to use Process mining (PM) as a means to develop customer journey maps. There is a lack of papers that delve into the topics of Customer Journey mapping and PM together. Very few papers have been published that have been found using basic terms such as: Customer Journey Map, PM, Customer Journey Process Map etc. One excerpt from a book and Two papers published were found to have overlapped between these two topics [1], [2], [3]. The term Customer Journey is widely used across many different professions such as Sales, Marketing and management. There is insufficient concrete agreement on what constitutes a Customer Journey. A recent paper [4] tries to review the meaning of Customer Journey in detail. For the purpose of this project, we consider Customer Journey Maps (CJM) as a visual illustration of every event a customer has gone through as part of his journey. This includes a conformance check against a process map generated through PM. Any events that do not conform would be highlighted in the CJM. Instead of attempting to review literature on Customer Journey, the aim of this initial literature review is to understand and digress to the topic of PM. The aim of this literature review is to gain an understanding of the PM field which is at the intersection of Data Science and Business Process Modelling.

So, what is PM?

Many processes use IT systems in their processes. In fact, nowadays most processes of large enterprises or public sector are operated on IT systems. These processes have a manual Process diagram of how things the process should occur. This generally appears far from reality where IT systems allow for far more flexibility in actions that can be taken in a process and the same systems generate logs of all actions/tasks. PM is the use of these logs that IT systems generate to establish the actual process model vice versa the assumed process model by the process designer/owner.

In an event log, individual records are considered an event. If we take a complaints log, a record indicating new complaint, closed complaint, complaint transferred are all events. A specific complaint might be identified by a Case Id that is common to all events that took place under that ID. All event logs are underpinned by Timestamps to indicate when they have occurred. The ordering of a Case Id's events by the timestamp gives us a trace of events that occurred for that instance (Case Id). A more detailed introduction to PM can be attained from www.processmining.org which is dedicated to the introduction of PM and run under the authority of W.M.P. Van der Aalst, a foremost expert in the field of PM [5].

A range of challenges present themselves to PM, albeit not all event logs may feature all these challenges at the same time. Some prominent challenges are:

1. Noise: incorrect sequence of event for the process or additional/lack of events of the process.
2. Non-free Choice constructs: Events have strong dependencies to other events at the trace level.
3. Mining loops: An event might repeat itself in some instances.
4. Visualizing results: There are many different notations in the field of Business Process modelling to define a process (Work-flow nets, petri-nets, BPMN etc..). A PM algorithm needs to be able to convert its process model into one of these notations for visualization.

There are more challenges that come custom made to different event logs and these can be explored in section 3 of [6].

Literature Review

One crucial algorithm that broke away from initial variations of the Alpha algorithm and heuristics algorithms is the Genetic algorithm. It is introduced by A.K. Alves de Medeiros, A.J.M.M. Weijters, and W.M.P. van der Aalst in 2005 [7]. The authors state algorithms such as the Alpha algorithm, Heuristic miners and their variations have problems dealing with noise and complex process structural formations displayed in the event logs. The authors identify noise challenges to be incorrectly logged events or exceptional traces produced in the log. The following structural constructs are also highlighted as challenges not dealt with existing algorithm's; non-free-choice constructs: *An event is dependent on a predecessor event that is not it's immediate local*, Invisible tasks: *Events where routing decisions are made but not present in logs* and Duplicate tasks: *Tasks that appear at multiple intervals of a trace*. *Past algorithms have treated this duplication as a single task in the process.* The approach to solve these problems is to make use of genetic algorithms that 'mimic' the evolutionary method. The steps to this solution are

1. Create a set of randomly generated initial representation(s)(IR) of the dependencies between a set of tasks.
2. Take these dependency representations and form process models. Measure fitness of model versus event logs.
3. Promote Process Models that 'correctly parse the more frequent behaviour in the log'.
4. Then apply evolutionary process methods i.e combining parts of two process models or randomly alter part of the process model.
5. Measure the fitness of these models to the data.
6. Pick high fitness models and repeat step 3, 4 & 5 for a total of N times until either fitness = 1, N number of iterations is reached, or fitness is the same for N/2 number of times.

The paper provides details of the experiments conducted on a varied set of tasks in test data. Each log had 1000 instances of the process and was tested for a maximum of 100,000 iterations/generations. The initial population was 500 models with the possibility of having duplicate process models within them. The results show that Genetic Algorithm can produce Process Models that capture all the behaviour in the event log and allows for possible traces not found within the event log. The paper attributes these unsought traces to the fitness calculation method used where a model is not penalised in the fitness calculations if the model allows event traces that do not exist in the log. The paper's suggested solution to this would be: count the number of traces (different sequences) that each process model can generate outside of the possible traces from the Event log. If the initial population of process models is large, then the computing resources required as iterations increase could be Inversely exponential. It may only be feasible for process models with a small set of tasks. The results are based on noise free logs, however the paper does not present us with any idea of how the same experiment would have resulted if noise was present.

The Genetic Algorithm seems to be a good text book algorithm but might face a lot of challenges when it starts dealing with complex real-life event logs. Gunther and Van der Aalst in "*Fuzzy Mining – Adaptive Process Simplification Based on Multi-perspective Metrics*" [8] proposition that real IT systems produce highly complex set of event traces that do not necessarily adhere to simple processes. The paper stipulates process owners are purposefully choosing for their IT systems to be flexible for its operators. Traditional algorithms tend to produce what is called the 'Spaghetti Model' where the model is an accurate reflection of reality. It is however too complex and not useful as a

process flow graph. The paper characterises a few common assumptions previous PM algorithms made:

1. Event logs are reliable and trustworthy. That is there are no unrelated event logs and the logs are well formed and homogeneous.
2. There is always a meticulous process model that can represent the data in the logs.

The paper baselines some key high-level ideas for effective re-mapping of these 'Spaghetti models'. Abstracting away unnecessary information in the logs, grouping some events in the log to form a coherent task, increasing the significance of certain tasks in the visualization aspect and Customising or contextualising the process graph to suit the purpose of the process map are what could make the spaghetti models useful.

Two simple concepts of significance and co-relation are introduced to achieve the ideas baselined by the paper.

The first, a metrics framework is introduced where significance measurements are used to remove any large number of insignificant events (i.e. auto-save event), add value to events that create forks (decision points), add significance to local or global event relationships (structural construction). Co-relation is also calculated between events based on event attributes to better understand how well preceding/succeeding events are related to each other.

A purely graphical simplification algorithm based on nodes and edges is also produced. In this simplification process, the same high-level ideas are implemented where edges between nodes are removed based on significance and co-relation. Followed by application of abstraction and aggregation to remove or cluster the less significant nodes.

The paper provides a complex spaghetti diagram with many nodes and edges in Figure 1. It then implements the above principles and metric calculations to provide a much simpler diagram. We are not presented with any numeric figures, however Fig. 8 in the paper looks considerably more readable with far less nodes and edges. Sound evaluation on different datasets is lacking and a the algorithm seems to establish only a loose framework to build upon.

Many PM algorithms presented in academic papers try to mine accurate process models. One thorn for these algorithms is noise. Noise in PM is the outlier data points in data mining that skews data insights. The goal of this paper "*Filtering out Infrequent Behaviour from Business Process Event Logs*" by Raffaele Conforti, Marcello La Rosa and Arthur H.M. ter Hofstede [9] is to tackle this noise problem. The paper makes a case for filtering out noise before the implementation of any PM algorithm on an event log. It also claims to be the "first effective technique for filtering out noise from process event logs" that helps PM techniques achieve better performance results.

The key is to remove events at the granular (record) level so as to keep all event traces intact. The paper makes a distinction that Algorithms such as the Heuristic miner and Inductive miner do not remove infrequent behaviour. They simply cope with ambiguous behaviour (i.e. cannot figure out of two events are in series or parallel) or implement filtering processes at the graphing level rather than at the log level. Removing edges or nodes at the log level does not mean the influence of those edges/nodes on other nodes is also removed.

The paper makes a comment that the Fuzzy miner algorithm's problem is that it leaves room for interpretation of it's graphs due to its abstract representation of the process behaviour. This open-ended abstraction criteria can also become convenient for real life implementation, where the algorithm can be tweaked to the desires of business rules. In this process, it could also loose the "PM

presents realistic process model based on actual events" phenomenon that has helped to bring PM to the spot light of Business Process Modelling.

The paper presents two concrete measures to remove infrequent relations between events/nodes:

1. Infrequent behaviour between events (One to One) (section 3 in [3]).
2. Identify the frequency threshold that removes optimal number of infrequent behaviour (Many to One).

The paper produces a rigorous evaluation of the proposed algorithm to remove infrequent behaviour. The authors test the Filtering mechanism with Artificial and real-life datasets. Within the artificially created datasets, a varying degree of noise is induced (5%-40%). The tests are conducted using Inductive, Heuristic, Fodina and Integer Linear Programming (ILP) algorithms. The fitness and precision of the Logs mined to Process models fare much better when the logs are filtered using the infrequent behaviour removal techniques presented in the paper. The only exception is the ILP algorithm for Fitness, where ILP performs slightly better without the removal of the infrequent behaviour removal from the logs. The same form of testing is repeated for real-life logs and we can examine similar results in favour of removing infrequent behaviour. This algorithm is still relatively new so there is a lack of practical evaluation on real-life datasets or examination by other papers.

Conclusion

The problems of noise and accurate event linkages appear as two concentrates that any PM algorithm needs to achieve to be regarded as standard algorithms. All the algorithms reviewed here have their merits as well as some deficiencies. There are many tools online that provide access to work with various mining algorithms. A few prominent ones are Minit, Fluxicon and ProM. ProM plays the favourite for academics as it has a plug and play approach and is also available for free to use by anyone. The effect of decisions made at the Process modelling level needs to be examined more closely if we were to use these models to render Customer Journey Maps(CJM).

Bibliography

[1] G. Bernard and P. Andritsos, Information Systems in the Big Data Era (CJM-ab: Abstracting Customer Journey Maps using PM), 2018.

[2] G. Bernard and P. Andritsos, "A PM Based Model for Customer Journey Mapping," in *International Conference on Advanced Information Systems Engineering*, Asten,Germany, 2017.

[3] A. Terragni and M. Hassani, "Analyzing Customer Journey with PM: From Discovery to Recommendations," in *2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud)*, 2018.

[4] A. Følstad and K. Kvale, "Customer Journeys: A Systematic Literature Review," *Journal of Service Theory and Practice,* vol. 28, no. 2, pp. 196-227, 2018.

[5] W. V. d. Aalst, PM, Eindhoven: Springer, 2016.

[6] A. Tiwari and C. Turner, "A review of business PM: State-of-the-art and future trends," *Business Process Management Journal,* vol. 14, no. 1, pp. 5-22, 2008.

[7] A. A. de Medeiros, A. Weijters and W. Van der Aalst, "Genetic PM: A Basic Approach and Its Challenges," in *Business Process Management Workshops, BPM 2005 International Workshops.*, Nancy, France, 2005.

[8] C. W. G¨unther and W. M. v. d. Aalst, "Fuzzy Mining – Adaptive Process Simplification Based on Multi-perspective Metrics," in *Business Process Management, 5th International Conference, BPM 2007*, Brisbane, Australia, 2007.

[9] R. Conforti, M. L. Rosa and A. H. t. Hofstede, "Filtering out Infrequent Behavior from Process Event Logs," *IEEE Transactions on Knowledge and Data Engineering,* Vols. 1-1, p. PP(99), 2016.