Towards Unsupervised Detection of Process Models in Healthcare

Amirah Alharbi [a,b,1], Andy Bulpitt [a] and Owen A. Johnson[a]
[a] School of Computing University of Leeds, Leeds, UK
[b] Umm Al-Qura University, Mecca, KSA

Abstract. Process mining techniques can play a significant role in understanding healthcare processes by supporting the analysis of patient records in electronic health record systems. Healthcare processes are however complex and patterns of care may vary considerably within similar cohorts of patients. As a result process mining often creates “spaghetti” models which require significant domain expert input to refine. Machine learning approaches such as Hidden Markov Models (HMM) may assist this refinement process. HMMs have been explored in process mining research and advocated for patient pathways clustering purposes; however these models can also be utilized for detecting hidden processes to help event abstraction. In this paper, we explore the use of an unsupervised method for detecting hidden healthcare sub-processes using HMMs, in particular the Viterbi algorithm. We describe an approach to enriching the event log with HMM-derived states and remodeling the healthcare processes as state transitions using a process mining tool. Our method is applied to event data for ‘Altered Mental Status’ patients that was extracted from a US hospital database (MIMIC-III). The results are promising and show a successful reduction of model complexity and detection of several hidden processes unsupervised by a domain expert.

Keywords. Process mining, unsupervised learning, Hidden Markov Models, electronic health records, event abstraction, MIMIC-III

1. Introduction

Modeling the care processes within healthcare is a challenging task due to the inherent complexity of patient care. Processes may vary considerably among the same cohort of patients as organizations and clinicians vary in response to each individual patient’s different physiological, psychological and social needs. Process mining techniques can play a significant role in understanding these real patterns of care by applying machine learning algorithms to the event logs extracted from Electronic Health Record (EHR) systems [1]. EHRs record numerous events during a patient’s visit to a hospital including medical, administrative, laboratory, intensive care and billing events. An event log records each event as a tuple with identifiable attributes including event name, event time and patient ID. Many healthcare events occur overlapping or in conjunction with other events which reflect the “interrelatedness” of healthcare processes [2]. Previously published studies have highlighted the importance of event abstraction as a key preliminary step to detecting an understandable process model [3]. Without a strategy for event abstraction process mining, particularly in healthcare, produces “spaghetti-like” models which have little value. Although studies have successfully improved the

[1] Corresponding Author; Amirah Alharbi, School of Computing, University of Leeds, Woodhouse Lane, Leeds, LS2 9JT, UK. E-mail: ml13ama@leeds.ac.uk

understandability of process models, they have generally relied on involvement from a domain expert. Untangling spaghetti models with the help of domain experts can be expensive and time consuming. For example, in previous work [4], we developed a model for chemotherapy care which used a clinical reference group of domain experts which took eight iterations over nine months.

In this paper, we aim to use an unsupervised learning technique to detect hidden processes within the data without involving a domain expert in the model extraction stage. Hidden Markov Models (HMMs) have been explored by process mining researchers and advocated for patient pathways clustering purposes where different models are built for different group of patients [5], however HMMs can be also utilized for detecting hidden processes which can be used for event abstraction. Our paper presents related work, our approach, some example results, a discussion and conclusions for further work.

2. Related work

Process mining methods that address complex model issues mostly rely on the concept of event abstraction. Our review of the literature identified four major approaches that have been adopted:

2.1. Mapping based approach

In complex organizational environments such as healthcare the details of specific events are often recorded with a high degree of granularity. High granularity produces complex process models which can be reduced by mapping low level, highly specific events to high-level activities. In [3], the authors suggest a formal method for mapping events to activities using the domain knowledge provided by stakeholders. This method has successfully captured m:n mapping relations between low-level events and high-level activities. This is like our approach in [5] and is expensive in domain expert time. Some simple mapping can be achieved where low-level event names are grouped into categories or ontologies such as SNOMED-CT.

2.2. Clustering/decomposition based approach

An automated approach to improve process modeling by clustering similar events has been explored in [6,7]. The clustering method in [6] used a causality matrix to show the relationship between events. For instance, if two events are mostly followed by each other they are grouped into one cluster. The user is required to set a causality score for clustering events. On the other hand, a general divide and conquer approach is proposed in [7]. This approach explored different ways to divide a log into sub-logs by finding the best partitioning point between events that reduce sub-logs overlapping.

2.3. Pattern based approach

Patterns can be extracted based on their frequency as in [8] where the authors aimed to find an episode which is an unordered subset of events that occurred frequently. An example of this approach is the episode miner in the ProM, process mining tool (www.promtools.org), which can extract events with direct follow and parallel relations. Such an approach does not support exclusive or and loop relations and can be extremely slow with large event logs. Another paper [9] proposed a supervised event abstraction method by involving a domain expert to identify event patterns.

2.4. Local process model approach

A local process model approach aims to discover relevant events with different relations such as sequence, parallel, choice and loops. The author in [10] developed a method to find the best-fit local process models through the generation of multiple possible local models from a limited number of events and then evaluation of the generated models using different quality criteria.

3. Method

In this work, we used the MIMIC-III (Medical Information Mart for Intensive Care) database [11] and extracted event logs for specific disease types expected to be reasonably homogenous. Here we present an example of experiments using event logs for 356 patients with 'Altered Mental Status' diagnosis. For a detail description of MIMIC-III and methods for event log extraction see [12].

Figure 1 shows the baseline spaghetti-like process model for our case study. This model was generated by the DISCO process mining tool (www.fluxicon.com/disco). We reduced complexity by mapping low-level event names to a set of 16 event types and excluded the 35% least common variants. The resulting model is still regarded as too complex and "messy" to provide useful insights into the care process.

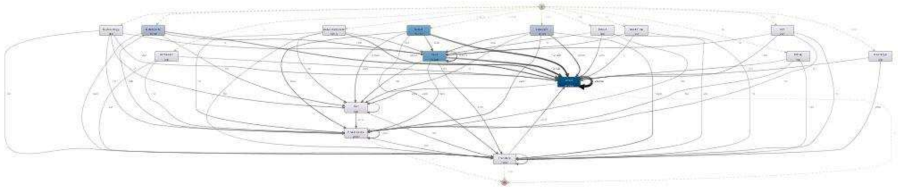

Figure 1: A "messy" spaghetti-like process model of the care of Altered Mental Status patients

3.1. Modeling healthcare processes with state abstraction

Our approach was to further reduce model complexity using event abstraction. Following [5] we trained HMM and used the Viterbi algorithm to find the underlying hidden states for sequences of events. Our method consists of two steps: Step 1) Finding the optimum number of HMM states; Step 2) Enriching the event log with HMM states and using these for further process mining.

One of the challenges of using HMM approaches is that parameters estimation is a time-consuming task and the EM 'Expectation Maximization' algorithm, which is used for model learning, is sensitive to initialization values. Therefore, we trained HMMs using the 'seqHMMR' package in the R statistical computing software tool [13] with a large number of random initializations in order to learn the most appropriate model. Table 1 shows the resulting HMMs with different initialization states. The best model is the model with 8 states because it has the best Final Log likelihood of fitting training data and test data, also it has the best value of Bayesian Information Criterion (BIC), a Bayesian measurement that takes model complexity into consideration [13].

Table 1. Different initialization for HMMs parameters estimation

States	Iteration convergence = (MX)3000	Final Log likelihood	BIC
6	900	-151989	305406
7	1572	-150257	302263
8	2679	**-149818**	**301728**
9	1454	-151251	304961

The second step was enriching the event log with HMM states using the Viterbi algorithm to extract the most probable sequence of states for each patient. Each event can be assigned to multiple states. The HMM state represents the “hidden” sub-process while the event is an observation within that sub-process. A group of hidden sub-processes constitute the end to end model.

4. Results and Discussion

We used the enriched event log as input to the DISCO process mining tool to produce process models based on these hidden states (Figure 2).

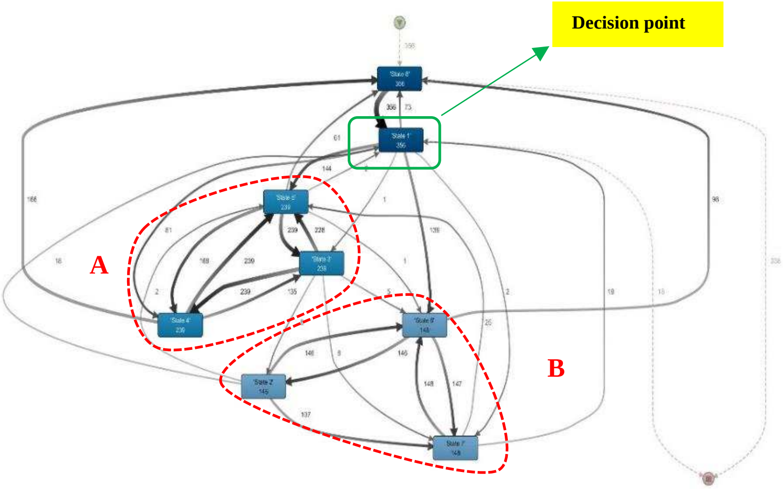

Figure 2: state-based abstraction of healthcare model

The model showed all states for 100% of cases rather than just the 65% most common variants shown in Figure 1. It helped to detect two groups of hidden care processes (clusters A and B in red in Figure 2). All patients have started their care process with State 8 then moved to State 1. In the model, State 1 appears as a decision point for the next care processes (indicated with a green box). 144 patients have followed hidden process A which consists of States 3, 4, and 5 while 139 patients have followed hidden process B which consists of States 2, 6 and 7). These two main care process clusters were not evident in the spaghetti diagram of Figure 1 and could not be discovered without the abstracted models.

5. Conclusion

The key advantage of using event abstraction is to reduce complexity and simplify healthcare models so that the limited and valuable time of domain experts can be better used. Our experiments show that using an unsupervised learning technique has the potential to help. The process mining community tends to neglect the possibility of using state transition modeling in representing process models because of its inability to capture parallelism and choices structures. Our approach however shows that state modeling can be used for event abstraction by adding HMM derived states as an attribute to events in the event log and generating a process model augmented by states. Detecting decision points in the abstracted model became easier and helped to provide a way of finding similar sub-processes which is a significant step towards detecting process models in highly complex processes. Our approach relies on the estimated parameters of HMMs so we suggest finding additional metrics for evaluating the number of states such as cluster entropy and transition rates. The results are promising but are based on small data. In further work we will refine the experiments using larger and more complex patient cohorts and comparing processes. For future work we will also aim to apply this method to local healthcare data to demonstrate practical value and to validate our results If successful, unsupervised detection approaches will make better use of valuable domain expert time in the future.

References

[1] Rojas E, Munoz-Gama J, Sepúlveda M, Capurro D. Process mining in healthcare: A literature review. Journal of biomedical informatics. 2016 Jun 30;61:224-36.

[2] Kannampallil TG, Schauer GF, Cohen T, Patel VL. Considering complexity in healthcare systems. Journal of biomedical informatics. 2011 Dec 31;44(6):943-7.

[3] Baier T, Mendling J, Weske M. Bridging abstraction layers in process mining. Information Systems. 2014 Dec 31;46:123-39.

[4] Baker K, Dunwoodie E, Jones RG, Newsham A, Johnson O, Price CP, Wolstenholme J, Leal J, McGinley P, Twelves C, Hall G. Process mining routinely collected electronic health records to define real-life clinical pathways during chemotherapy. International journal of medical informatics. 2017 Jul 31;103:32-41.

[5] Da Silva GA, Ferreira DR. Applying hidden Markov models to process mining. Sistemas e Tecnologias de Informação. AISTI/FEUP/UPF. 2009.

[6] Hompes BF, Verbeek HM, van der Aalst WM. finding suitable activity clusters for decomposed process discovery. In International Symposium on Data-Driven Process Discovery and Analysis. 2014 Nov 19 (pp. 32-57). Springer.

[7] Van Der Aalst WM. A general divide and conquer approach for process mining. In Computer Science and Information Systems (FedCSIS), 2013 Sep 8 (pp. 1-10). IEEE.

[8] Leemans M, van der Aalst WM. Discovery of frequent episodes in event logs. In International Symposium on Data-Driven Process Discovery and Analysis 2014 Nov 19 (pp. 1-31). Springer.

[9] Mannhardt F, de Leoni M, Reijers HA, van der Aalst WM, Toussaint PJ. From low-level events to Activities - a pattern-based approach. In International Conference on Business Process Management 2016 Sep 18 (pp. 125-141). Springer International Publishing.

[10] Tax N, Sidorova N, Haakma R, van der Aalst WM. Mining local process models. Journal of Innovation in Digital Ecosystems. 2016 Dec 31;3(2):183-96.

[11] Johnson AE, Pollard TJ, Shen L, Lehman LW, Feng M, Ghassemi M, Moody B, Szolovits P, Celi LA, Mark RG. MIMIC-III, a freely accessible critical care database. Scientific data. 2016; 3.

[12] Alharbi A, Bulpitt A, Johnson O. Improving Pattern Detection in Healthcare Process Mining Using an Interval-Based Event Selection Method. In International Conference on Business Process Management 2017 Sep 10 (pp. 88-105). Springer.

[13] Helske S, Helske J. Mixture hidden Markov models for sequence data: the seqHMM package in R. arXiv preprint arXiv:1704.00543. 2017 Apr 3.